

(Hello World)

THE MAGAZINE FOR COMPUTING
& DIGITAL MAKING EDUCATORS

INTERVIEW: KATE FARRELL

Using data to solve real-world problems

DECOLONISING COMPUTING

Widening global perspectives in computing

POLYGLOT PROGRAMMERS

Tips for teaching multiple programming languages

Issue 16

June 2021

helloworld.cc

THE
POWER

OF
DATA

INSPIRE STUDENTS TO USE
DATA TO MAKE SENSE OF
THE WORLD



PLUS

HELLO WORLD PODCAST • RUNNING SUCCESSFUL COMPUTING CLUBS • BIOINFORMATICS
GAME-BASED LEARNING • PANDAS FOR BEGINNERS • DATA SCIENCE FOR SOCIAL GOOD
ABSTRACTION • ROBOTICS AND FUTURE SKILLS • CLOUD COMPUTING • GRAPHING DATA



Raspberry Pi

Learn with the Raspberry Pi Foundation

Free for everyone anywhere in the world



Teaching resources

Discover training, resources, and guidance to help you teach computing with confidence.

teachcomputing.org



Project library

Browse our 200+ online project guides that include step-by-step instructions for learners.

projects.raspberrypi.org



Digital making at home

Join our weekly live streams and take part in Coolest Projects and Astro Pi Mission Zero from home.

raspberrypi.org/learn



Support for parents

Watch our support tutorials and access engaging resources for your child.

raspberrypi.org/learn

HELLO, WORLD!

Why hello, Hello World! I have been a long-time admirer and reader of the magazine, and so I am thrilled to have joined as the new editor.

Over the last year, we have all had to become data scientists to some degree: navigating graphs of infection rates and vaccine roll-outs; interpreting election results data; and ignoring our phones and watches buzzing, telling us we've not done enough steps today or slept very well. We are living in an increasingly data-driven world, and it is vital that students can understand, interpret, and use data. This issue, we're delving into the power data has to bring about change, mislead us, and allow us to make decisions about our lives.

We know that data science topics can occasionally, on the surface of things, seem dry or abstract to students, so we've got some wonderful articles and lesson plans to help you bring data to life in your classroom. We

hear from Simon Johnson about the importance of contextualising data science topics, and from Ben Garside about introducing visualisations to tell a story with data; we also bring you a profile of people working with data science in different industries, to pique your students' interest.

Since starting as editor, I've had the chance to look at some of the data (there it is again!) from our monthly reader surveys. We've had some great suggestions and comments, which you can read about on page 95, and we will continue to make improvements and feature content that you're interested in. Please get in touch with your thoughts at contact@helloworld.cc and visit helloworld.cc/writeforus if you have an article idea — we'd love to hear from you!

Gemma Coleman
Editor



FEATURED THIS ISSUE



YOTA DIMITRIADI

On **page 67**, Yota, an associate professor at the University of Reading in the UK, discusses the importance of widening global perspectives in computing education.



VAL QUAYE

Val teaches computer science and ICT in Kuala Lumpur, Malaysia. On **page 72**, she shares her experience of reflecting upon her career development as a computing teacher.



BRUCE AHERN

Bruce is a subject leader for computing in the UK. He kicks off what we hope will be a new regular feature on **page 87**, My Classroom, sharing how he brings his computing classroom displays to life.

(HW)

EDITORIAL

Editor

Gemma Coleman

Subeditors

Louise Richmond and Amy Rutter

Subscriptions

Joshua Crossman

Social Media

Claire MacDonald and Neena Patel

DESIGN

criticalmedia.co.uk

Head of Design

Lee Allen

Designers

Ty Logan, Sam Ribbits

Photography

Raspberry Pi Foundation, Adobe Stock

Graphics

Rob Jervis

Cover

©Muti, Folio Art

CONTRIBUTORS

Bruce Ahern, Shirley Atkinson, Stevie Bain, Daniel Barker, Mark Calleja, Katharine Childs, Kim Chouard, Gemma Coleman, Michael Conterio, Nicola Curnow, Jonathan Dickens, Yota Dimitriadis, Rebecca Donnelly, Catherine Elliott, Rebecca Franks, Ben Garside, Graeme George, Sway Grantham, Sujatha Gunja, Philip Harney, Daniel Hickmott, Martin Hill, Simon Humphreys, Laura James, Simon Johnson, Shashi Krishna, Thom Kunkeler, Alan O'Donohoe, Matthew Parry, Andy Perrin, Carrie Anne Philbin, Val Quay, Steve Rich, Neil Rickus, James Robinson, Julia Roebuck, Chris Roffey, Laura Sach, Omnia Saed, Sam Shallcross, Victoria Temple, David Walker, Huzafah Zainon

Supported by **ORACLE**

Hello World is a joint collaboration:



Raspberry Pi



This magazine is printed on paper sourced from sustainable forests and the printer operates an environmental management system which has been assessed as conforming to ISO 14001.

Hello World is published by the Raspberry Pi Foundation, 37 Hills Road, Cambridge, CB2 1NT. The publisher, editor, and contributors accept no responsibility in respect of any omissions or errors relating to skills, products, or services referred to in the magazine. Except where otherwise noted, content in this magazine is licensed under a Creative Commons Attribution-NonCommercial-ShareAlike 3.0 Unported (CC BY-NC-SA 3.0).



(HW)

CONTENTS

15-33

THE POWER OF DATA

A series of articles inspiring students to use data to make sense of the world

©cervo/stock.adobe.com

NEWS, FEATURES, AND OPINION

6 NEWS

New resources from Barefoot; CAS virtual showcase; Hello World podcast launch; two new reports; resources from the NCCE

13 INCLUSION COLUMN

Building key digital skills for learners with SEND

15 PANDAS FOR BEGINNERS

A step-by-step guide combining the pandas Python library with Google Colab for data visualisations

18 INTERVIEW

Data Education in Schools' Kate Farrell discusses getting learners to use data to solve real-world problems

20 INSIGHTS – DATA SCIENCE FOR SOCIAL GOOD

A coding camp emphasises meaningful uses of computing to increase diversity

21 INSIGHTS – DESIGN PRINCIPLES

Activity design principles for improving data literacy

22 HISTORICAL DATABASES

Using local history data to teach databases

24 DATA VISUALISATIONS

Visualising data to get students started with data science

26 CONTEXTUALISING DATA SCIENCE

Engaging students with data science by contextualising topics

28 DATA SCIENCE JOBS

Data science job profiles to inspire your students

30 BIOINFORMATICS

The interdisciplinary subject combining computing, biology, and maths

32 GRAPHING DATA

The importance of using more than just statistical measures to understand data

34 SUBSCRIBE TODAY!

36 CLOUD COMPUTING

Using cloud computing for practicals during the pandemic

37 CT POD

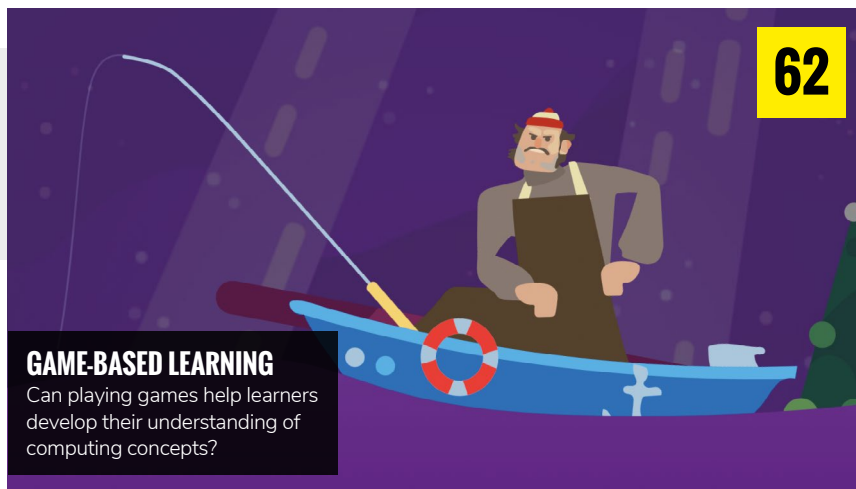
A school's podcast sharing ideas for computational thinking across the curriculum

38 ACTIVE PARTICIPATION

One teacher's experience of using Pear Deck to boost virtual participation

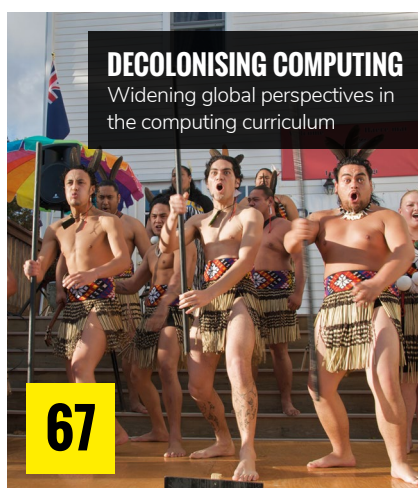
40 VEXCODE VR

Introducing physical computing remotely



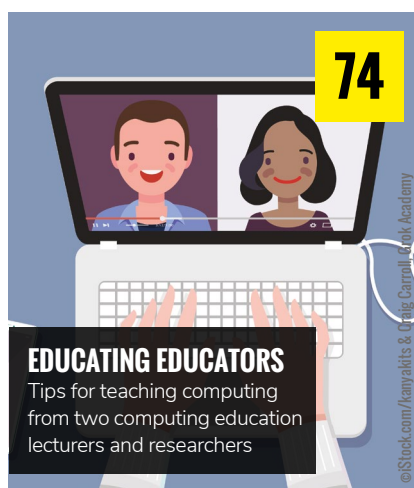
GAME-BASED LEARNING

Can playing games help learners develop their understanding of computing concepts?



DECOLONISING COMPUTING

Widening global perspectives in the computing curriculum



EDUCATING EDUCATORS

Tips for teaching computing from two computing education lecturers and researchers

42 COMPUTING OUTREACH

A university learns lessons for the online delivery of outreach events

44 JAVASCRIPT

Switching to JavaScript for teaching programming remotely

46 POLYGLOT PROGRAMMERS

Why and how you should teach students multiple languages

48 ABSTRACTION

How to teach abstraction using PRIMM and My Blocks in Scratch

50 WEB DEVELOPMENT

Using web page creation to complement programming and computational thinking

52 BLOCKLY TURTLES

A free resource to teach programming concepts

54 ASSEMBLY LANGUAGE

Using Raspberry Pi to teach assembly language to A level students

57 MOONHACK

This year's annual coding challenge focuses on Indigenous cultures in Australia and New Zealand

58 ROBOTICS COMPETITION

A school's experience with the FIRST Tech Challenge

60 ROBOTICS FOR THE FUTURE

Preparing students for an uncertain future through robotics

64 A GAME TO INSPIRE

A gamified curriculum teaching coding and digital literacy

66 PHYSICAL COMPUTING

Going back to basics with physical computing

70 GOOGLE'S CODE NEXT

A program expanding accessibility in computing

72 THE NEXT STEP

One teacher explores the next steps in her computing teaching career

LEARNING

RESOURCES & LESSON PLANS

76 GROUPING DATA

Labelling and matching activities for lower primary

78 DATA LOGGING

Activities for learners aged 8–9 answering questions using data

81 SPREADSHEETS

Engaging spreadsheet activities for upper primary

84 MACHINE LEARNING

A short introduction to machine learning concepts for lower secondary, using Teachable Machine and Scratch

CONVERSATION

87 MY CLASSROOM

One teacher's ideas for creating a computing classroom that will inspire students

88 INSIDER'S GUIDE

How to run a successful computing club

94 BEBRAS

A fun computational thinking challenge

95 SURVEY RESULTS

Some of the reader feedback we've received recently

REVIEWS

96 BOOK REVIEWS

100 computing lesson ideas; an adventure in coding for young learners; and a selection of books to help you explore data further

NEW BAREFOOT ONLINE SAFETY RESOURCES

Barefoot has produced free online safety resources for primary school students

Victoria Temple

Barefoot, in partnership with the National Crime Agency (NCA), has created a new set of resources to help primary school students stay safe online and inspire them with the opportunities cybersecurity offers.

The new #BeCyberSmart resource bank covers tough-to-teach topics such as permission and ownership, legal and illegal use of computers, and password protection. It does this in a way that's accessible and fun for both children and teachers, and also includes links to personal, social, health, and economic (PSHE) education and digital literacy.

It's increasingly important for children to understand online security from a young age, as Emily, cybersecurity expert at the NCA, explains: "Children use technology at a very young age, and it can be difficult to monitor how they spend their time online. The NCA has identified that children as young as eight have engaged in online activity which has breached the Computer

Misuse Act. Children can become so skilled; they use their computer to cause serious disruption and have been known to launch cyberattacks against schools. Education is a key foundation for an effective crime prevention programme. The new Barefoot resources will help pupils understand their responsibilities in a fun and informative way."

For lower-primary students, understanding their responsibilities online is an effective foundation to keep them safe and help prevent crime. That means raising awareness of copyright and digital permission, which the resources cover. "Through examining both physical and digital objects within a lesson, children develop their understanding of ownership and when they need to ask others to use and share content," explains Neil Rickus, education specialist for BCS, the Chartered Institute for IT, and one of the resource's content writers.

As children enter upper primary, they become increasingly skilled with technology and spend more time on screen, and the potential for them to use a computer to cause significant disruption increases. "The Key Stage 2 (upper primary) #BeCyberSmart resources introduce children to the Computer Misuse Act through a courtroom scenario, where children have to pass judgement on a range of offences linked to technology usage," says Neil. "Children are often unaware of the impact cybercrime can have, so the resources include information on the victims of crime, which has led



to really interesting conversations in the classroom when trialling the lessons."

The resources are not only about keeping safe online; they also aim to inspire children about the opportunities in the cybersecurity industry where demand for people with digital skills is already outstripping supply. Louise Patterson, cybersecurity expert at BT, says: "I use computational thinking daily in my job — which is fast becoming one of the most important roles in the tech sector. Every Barefoot lesson covers at least one computational thinking concept and approach. From algorithms to collaboration, the learning within the Barefoot lessons will help children develop strong foundations in their online practice. Who knows, it could even spark an interest in a cybersecurity career like mine!"

To download Barefoot's free #BeCyberSmart resources, head to barefootcomputing.org. ^(HW)

ABOUT BAREFOOT

Barefoot is the computing education programme from BT and Computing at School. It has reached more than 85,000 teachers and 3 million pupils in the UK. It produces free, cross-curricular resources for the classroom and home learning, plus continuing professional development workshops and online guides.

CAS VIRTUAL SHOWCASE IS BACK!

The CAS Virtual Showcase, a programme of events celebrating the latest innovations in computing education, returns this summer

Victoria Temple

In association with the National Centre for Computing Education (NCCE), the Virtual Showcase is hosted by Computing at School (CAS), a community of teachers, academics, and professionals with an interest in computing.

Before the pandemic, CAS hosted face-to-face events each summer to share innovative ideas and inspire the teaching community, but the showcase moved online last year. This year's showcase runs from Monday 21 June to Friday 2 July and brings together teachers and experts from the CAS community, guest speakers, and industry leaders to offer around 25 online events and webinars. The events are free and are open to everyone, including non-CAS members.

Beverly Clarke, National CAS Community Manager at BCS, the Chartered Institute for IT, said: "It was a hugely successful

programme last year, which saw over 750 attendees at a wide variety of events. We're building on that success and are really excited about the programme we've put together for 2021's event. It's been a momentous year for everyone working in education, and the CAS Virtual Showcase is an opportunity to share experiences, expertise, and the latest developments."

The showcase will host speakers from a variety of backgrounds and cover a range of themes. Kate Arthur, founder of Kids Code Jeunesse, a charity which introduces computational thinking, coding, artificial intelligence, and ethics to communities across Canada, will be one of the keynote speakers at the CAS Virtual Showcase. "While we are facing crises in climate, public health, social justice, and global education, as educators, we have the

opportunity to equip young people with the tools to address them. I'm looking forward to bringing what I've learnt in Canada to the audience at CAS in June," said Kate.

Highlights of the fortnight

This summer's showcase will see sessions for primary and secondary teachers focussing on non-digital remote education, cybersecurity, well-being and artificial intelligence. There are also collaborations with STEM Ambassador and NCCE Computing Hubs, and Barefoot Computing.

Highlights include (all times BST):

- Empowering Kids to Tackle the Ethical Questions of AI and to Meet the Sustainable Development Goals by 2030, Kate Arthur (Mon 28 June, 4pm)
- Accessing Aspirations Through Digital Literacy, Baasit Siddiqui (Mon 21 June, 4pm)
- Health and Well-Being in a Blended Learning Environment, Dr Hazel Harrison (Wed 23 June, 4pm)
- Supporting Learners with SEND through a Blended Learning Approach, Catherine Elliott (Fri 2 July, 12 noon)
- Instagram Live Session: Top Tech Tips for Early Years Foundation Stage, Bethan Ware (Wed 30 June, 5pm)

To find out more and book a place, please visit computingatschool.org.uk. You can follow the event on Twitter using the hashtag #CASVirtual21. [\(Hw\)](#)



■ Baasit Siddiqui and Kate Arthur are two of the keynote speakers at this year's Virtual Showcase

HELLO WORLD PODCAST

A new way for educators interested in computing and digital making to experience Hello World

Carrie Anne Philbin MBE

There's a growing momentum behind the idea of putting computing, computer science, and digital making at the heart of modern education. At the Raspberry Pi Foundation, we feel there is a need to do more to connect with and support educators, both inside and outside the classroom. That's why we started Hello World magazine, a platform to help educators all over the world to find inspiration, share experiences, and learn from one another. It's proven to be popular, with subscribers spanning 173 countries across the globe.

We hear from readers that waiting for a new issue of the magazine to be published can be tiresome, and you're hungry for more. Consequently, to complement the magazine, we're launching a free podcast available between issues, so that you can hear directly from the educators behind the articles, delve a little deeper, and have some fun along the way.

In season one, we explore the importance of creativity and passion in computing with PBS Digital Innovator and CUE (Computer-Using Educators) rock star Amanda Haughs, and dive into the role of ethics in computing with Isaac Computer Science content creator Diane Dowling. We discover how to look after our well-being while teaching with technology, with practical tips from computing educator Cat Lamin, and senior lecturer in computing education at the University of Hertfordshire Neil Rickus. We will also be asking, "Are these the droids you're looking

for to teach algorithms?" with computing teacher Huzaifah Zainon and advanced skills computing teacher Nicki Cooper.

Start listening now by subscribing to the Hello World Podcast ([helloworld.buzzsprout.com](https://helloworld.cc/buzzsprout.com)). Let us know if you have a question that you would like answered, or a topic you would like us to explore more. You can get even more involved by sharing your top tips and best teaching practice and be a guest on one of our future episodes. Get in touch with us at podcast@helloworld.cc with your suggestions. (HW)

SUBSCRIBE TO THE HELLO WORLD PODCAST

- Apple Podcasts (helloworld.cc/applepod)
- Spotify (helloworld.cc/spotifypod)
- Google Podcasts (helloworld.cc/googlepod)
- Amazon music (helloworld.cc/amazonpod)

“ HEAR DIRECTLY FROM THE EDUCATORS BEHIND THE ARTICLES, DELVE A LITTLE DEEPER, AND HAVE SOME FUN ON THE WAY



■ Guests Jane Waite and Matthew Wimpenny-Smith speaking with Hello World podcast hosts Carrie Anne Philbin and James Robinson about ABC curriculum design

ONLY 9 PERCENT OF UK TEACHERS FELT PREPARED FOR ONLINE TEACHING

A new report from the Open Data Institute outlines the impact the pandemic has had on teachers' lives in the UK

Gemma Coleman

In March 2021, the Open Data Institute (ODI) published a report exploring the impact of the pandemic on the teaching workforce in the UK. The report analysed survey data with just under 4,500 respondents from a major UK teacher union, as well as other open and publicly available current and historical data. It investigates teachers' preparedness for remote teaching; their change in workload; estimates of learning losses in their pupils; and the likely influence of these experiences on staying in the teaching profession.

Preparation for online teaching

Overall, teacher preparedness for remote learning, and feelings of support within this new learning environment, were low. The report found that only 9 percent of teachers who had been teaching remotely felt well prepared for online learning, with this dipping to 3 percent for special education teachers. This is perhaps not surprising given the speed of physical school closures and the fact that there was no national policy on how to implement virtual learning, with schools and teachers having to learn new skills and acquire new equipment and software incredibly quickly and under huge pressure.

Workload

The average working week increased to 52 hours for full-time teachers. This increase was largely attributed to the challenges of remote teaching. Teachers had to



spend large amounts of time scripting and recording lessons and thinking of ways to make their lessons creative and engaging in this new medium. Many teachers also faced the challenge of balancing remote lessons with in-person lessons for children of key workers, and the additional classroom management time this posed. Over half of teachers interviewed attributed these extra hours to their increased pastoral duties, such as delivering food parcels and learning materials to vulnerable pupils.

Learning losses

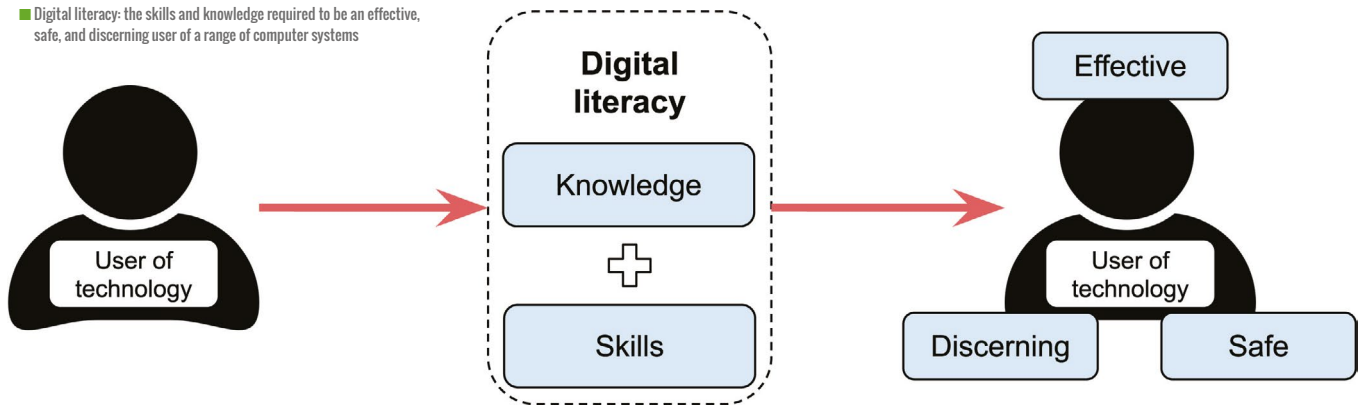
The survey reports that 59 percent of teachers felt at least half of their pupils had suffered academically during the pandemic and that this was worse for primary-aged children and

schools in deprived areas. Teachers felt much of this stemmed from a lack of engagement with online lessons, as well as a lack of parental support in some cases.

Despite the stresses and difficulties of the past year, there has been a decline in the proportion of teachers considering leaving their job or the profession. This may, of course, be a result of an uncertain job market, but the data suggests that teachers were, on average, enjoying their jobs more, due to their increased autonomy and a reduction in performance management.

The ODI hopes that these findings will be used to inform the pandemic response and support future education research and policy making. You can read the full report here: helloworld.cc/ODIreport. (Hw)

■ Digital literacy: the skills and knowledge required to be an effective, safe, and discerning user of a range of computer systems



NCCE DIGITAL LITERACY REPORT

In their first teaching and learning report, England's National Centre for Computing Education (NCCE) explores the meaning, role, and importance of digital literacy

James Robinson

In recent years, curricula and educators around the world have refocused on some of the more technical and practical aspects of computing, such as logic and programming. At the same time, there have been concerns about whether learners are acquiring the digital literacy skills they need to live, learn, and work in the modern world. The experience of the pandemic has highlighted the importance of these skills for learning, and shown us where the gaps are in learners' experiences and understanding.

Among the challenges for educators have been the varying meanings and vocabulary surrounding digital literacy, the limited tangible examples of what it looks like in practice, and a tendency to treat digital literacy as a separate aspect of computing. The development of the Teach Computing Curriculum (TCC) by the NCCE provides an opportunity to highlight examples of digital literacy throughout a learner's journey through computing.

Digital literacy findings

Through its analysis of the TCC and interviews with educators, the NCCE report shares a number of key findings:

- Digital literacy is not a new aspect of the curriculum; digital literacy skills and knowledge have long been integral to ICT/computing. However, there has been an increased focus on computer science skills and concepts, leading to the perception that digital literacy has been replaced. This may partly be because of how educators, publishers, and trainers have implemented the addition of computer science. This report shows that the curriculum was not designed to replace digital literacy; on the contrary, it is a priority.
- The TCC takes a holistic approach to computing as a broad discipline, mapping content to a ten-strand content taxonomy (for example, 'Algorithms', or 'Creating media'), across which we can identify elements of digital literacy. This demonstrates that rather than being a distinct area of the curriculum, digital literacy is woven throughout.
- Although digital literacy is typically associated with computing curricula, it shouldn't be limited to this subject. The report suggests that supporting learners

in developing their digital literacy is the responsibility of the school community as a whole, and learners should be able to demonstrate their digital literacy in all areas of their education.

- Providing learners with this experience is highly dependent on the skills and confidence of their teachers, who may need support in how to teach digital literacy effectively within their subjects, or in developing their own digital literacy and confidence.

The report identifies the digital literacy elements that students should develop at each stage of their learning, as well as highlighting examples from the TCC. Readers will also find suggestions for teachers who are looking to develop their own digital literacy, including professional development opportunities.

It is our hope that this report will be invaluable to educators when setting expectations for their learners and planning digital literacy opportunities with colleagues across the wider curriculum.

To read the full NCCE report, visit helloworld.cc/dlreport.



RECOVERY SUPPORT FOR COMPUTING EDUCATION

England's National Centre for Computing Education is offering free resources to teachers globally

Victoria Temple

Over the past year, teachers all over the world have become increasingly isolated as they've transitioned to remote and blended teaching. Opportunities to connect with other educators and gain peer-to-peer support have proved to be a vital resource. Computing at School (CAS) is a grassroots network of teachers across the UK, and during the pandemic its many communities have transitioned to regular online meetings and virtual sharing of resources and ideas. "We've found the number of teachers reaching out to the Computing at School community has increased here in Cumbria. They're making use of the knowledge and expertise that community leaders are providing," says Vicky Dodds, a CAS community leader.

"Although Covid-19 has tested us in terms of face-to-face delivery, it has not prevented us from reaching many schools and delivering a great level of support. Setting up CAS communities means we are future-proofing the support."

As well as networks such as CAS, the National Centre for Computing Education (NCCE) in England has curated an adaptable programme of resources, training, and expert advice to ensure teachers and their students are well positioned for the challenges faced in getting back to school, with a focus on rebuilding and recovery.

Prewritten tests

Effective recovery in delivering computing skills starts with an evidence-based approach and understanding the needs

of students, says Dave Gibbs, senior computing and technology specialist at STEM Learning, one of the consortium partners behind the NCCE. Home learning led to computing teachers reporting having a less secure view of what students have learnt, he says. "Effective, low-stakes assessment is integral to great teaching, and has been identified by head teachers as a top priority. Subject knowledge assessment can identify the key issues: what learning has been lost or misunderstood? Should topics be retaught to whole groups or a subset of a group? How secure are the foundations for the next stage of learning?"

"Covering the whole of the computing curriculum in England, our quality-assured tests are carefully written to uncover

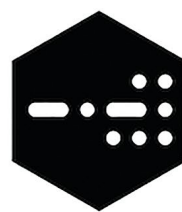
“RECOVERY WILL TAKE MORE THAN A COUPLE OF SCHOOL TERMS

➤ misconceptions and diagnose gaps in knowledge. Our two question banks can be used flexibly. For instance, a baseline assessment to identify priorities for interventions might be followed by a reassessment to check knowledge is secure. The tests can be customised to match your curriculum coverage, wherever you are in the world, and the Eedi platform produces detailed data to help you plan to teach,” says Dave. You can access them for free here: eedi.com/projects/teach-computing.

Curated recovery resources

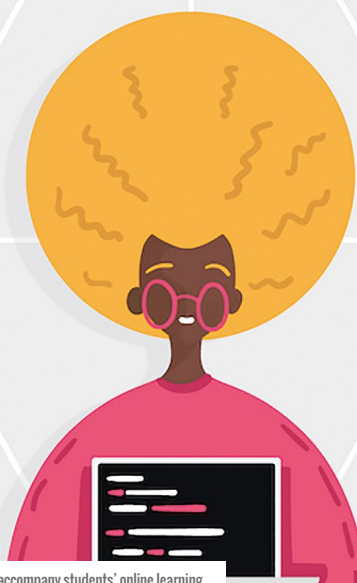
One of the key areas of focus for teachers are transition areas: from lower secondary to upper secondary, and from upper secondary to post-16 qualifications like A level Computer Science. Teachers need to be confident that their students' knowledge of the curriculum is sound in order to make a swift transition. The NCCE launched the Teach Computing Curriculum (TCC) a year ago, as a complete curriculum programme to teach children aged 5–16 (helloworld.cc/tcc). To support teachers with recovery, “We’ve brought together a wide range of ready-made lessons and resources from the TCC, as well as subject knowledge assessments hosted on the Eedi platform, and a set of targeted CPD interventions into a curated collection on the Teach Computing website,” says Carrie Anne Philbin, the Raspberry Pi Foundation’s director of educator support, who leads the team that created the TCC. “The aim is to save teachers time and reduce workload, as well as supporting them to identify gaps in students’ learning.”

For students over 16, Isaac Computer Science continues to provide learning materials that cover the full A level



isaac
computer
science

A Level Computer Science Workbook



■ Isaac Computer Science offers printed workbooks to accompany students' online learning

Computer Science curricula for AQA and OCR qualifications in England (isaacomputerscience.org). In addition, it offers low-cost printed workbooks to accompany students' online learning, and in the coming months it will offer booster workshops to help students catch up or revise, as well as online teacher CPD sessions (helloworld.cc/isaacevents).

Training

Having a comprehensive understanding of students' knowledge does, of course, require more than short tests. The NCCE is running a new face-to-face, one-day professional development course for teachers in England, *Assessment and progression in KS3 computing*,

which covers formative assessment strategies, including strategies relating to programming (helloworld.cc/ks3course).

Other online training courses are also available to educators all over the world (teachcomputing.org/courses). “We have new CPD in place to make sure that teachers not only have the subject knowledge, but also the understanding of curriculum design to enable them to plan for the future,” says Will Rogers, programme manager for the NCCE. “Education recovery from the impact of the pandemic is not something that will be solved in a couple of school terms, so we are putting things in place that support teachers and students over the months and years to come, both here in England and beyond.” (HWF)

KEY DIGITAL SKILLS FOR YOUNG PEOPLE WITH SEND

Catherine Elliott shares the importance of teaching key digital and digital literacy skills, and looks at how you can build them with your students

Computing curricula generally cover a wide range of skills, concepts, and knowledge, and much of the focus is often on creative projects, programming, and abstract computer science concepts. There is a strong argument, however, for ensuring that the basic skills are taught well in the first instance, with a particular benefit for young people with special educational needs and disabilities (SEND).

Basic digital skills are any skills that are required to access and use a computer effectively. The concept of 'digital natives' has been widely discredited, but many teachers believe that young people just 'get' computers and can use them with ease, as they are more confident users. However, although children may be experts at swiping and accessing content on certain devices, such as tablets, they often lack keyboard and mouse skills, and the understanding of the basics of an operating system.

Basic digital literacy is equally important. The skills and knowledge required to communicate effectively and use current and emerging technologies are essential if students are to remain safe and act appropriately online. Young people with additional learning needs and disabilities can be among the most vulnerable in terms of online risks and behaviours.

Why is developing fluency in basic digital skills and literacy important?

1. It reduces cognitive load

Cognitive load relates to the number of items a person can hold in their working memory. For many younger pupils and students with SEND, basic skills such as logging on and opening files are not practised enough to be moved to long-term memory, and therefore rely

on working memory to be completed. This results in the frustrating position of a child spending much of a lesson simply accessing a computer and finding work. Once these actions become fluent, they can concentrate on the content of the lesson more effectively.

2. It increases confidence in using technology

Once learners become fluent in using the computer in basic ways, their confidence will develop. This will help to increase motivation in lessons and contribute to a feeling of achievement.

3. It enables learners to use assistive technologies effectively and make simple modifications to content to support learning

Mainstream applications and devices now have a greater amount of assistive technology built in — for example, Immersive Reader in Office 365, Voice Typing in Google Docs, and Speak Screen on the iPad. If young people can be taught how to use these options independently as part of computing lessons, it will help them to become more effective in their learning. Similarly, highlighting how to increase the size of text or change the background colour in documents will allow learners to modify digital documents to be more accessible.



Catherine is the SEND lead for the Sheffield eLearning Service (sheffieldclc.net), and she works on ways to make the subject accessible to all learners. She is a member of the CAS Include working group, and leads the SEND Virtual and the Sheffield and South Yorkshire Secondary CAS Communities ([@catherinelliott](https://twitter.com/catherinelliott)).

4. Digital skills open up greater opportunities for employment

There are few jobs where digital skills are not required, and even the application process generally requires some element of using technology. For students not taking an IT or computer science qualification, schools need to consider how to teach key

ASSISTIVE TECHNOLOGY IN MAINSTREAM APPLICATIONS

- Support weaker readers with the Immersive Reader tool, built into Office 365 products and the Edge browser, and as an extension in Chrome (see helloworld.cc/immersivereader for more info)
- Enable Speak Selection in the Accessibility settings on the iPad to enable students to listen to any selected text
- Try the Voice Typing option in the Tools toolbar in Google Docs to allow learners to convert speech to text: there is also a dictation function via the keyboard on iPad and Android tablets, and in Microsoft Word online



©epiproductions/stock.adobe.com

- employability skills, such as sending emails and searching for information online. A functional skills qualification in IT would benefit some learners with SEND greatly in preparing them for their next steps in education or employment.

5. It helps students use technology safely and responsibly

Young people need to be taught about the risks of online technologies, how to act appropriately online, and where to go for help with safely accessing key services and the same opportunities for entertainment, shopping, and learning as their peers.

How can we teach key skills effectively?

- Develop fluency through routine. For example, pupils open and save work from the same folder each lesson, and have a routine for logging on when they enter the ICT suite or access a laptop. Share these routines with other teachers these learners work with.
- Provide support materials for learners for habitual actions — create a set of simple instructions for each step, with image support. For example, log on, open an application, save work, and take a photo on the iPad.
- Provide lots of opportunities for repetition to consolidate learning — time spent repeating an action multiple times or practising keyboard and mouse skills is worthwhile for helping key skills to become fluent early on in the year. Consider how to combine this with other meaningful tasks in the classroom, or to assist with other learning goals (for example, typing practice to support spelling).
- Teach learners how to use assistive technologies built

into the mainstream tools that are available at school or at home through a learning platform. There are a number of tools to allow students to dictate rather than type, have text read to them, or to add subtitles to videos. Where possible, share these with parents to use at home.

- Show students how to adapt digital content to make it more accessible. A great benefit of the past year has been that we have been able to provide lesson content for students to access in their own time, to revisit and consolidate their learning. If this is in an editable format (e.g. Word, PowerPoint, or Google Docs), learners can change the background colour, increase the font size, choose a more readable typeface, and increase line spacing to make it more readable. If you are providing videos to watch, enable the use of closed captions.
- Model safe and responsible use of technology in the classroom, and make your actions explicit through commentary and discussion.
- When teaching digital literacy and online safety, make sure you discuss risks and behaviour in a number of different situations and contexts, as some young people with SEND struggle to generalise their knowledge. Issues also need explaining explicitly for those students who can't infer risk from subtle cues.

Computing often doesn't receive enough space on the curriculum. However, an investment of time and effort in developing key digital and digital literacy skills early on in a child's computing journey will pay dividends later in increased confidence, fluency, and access to learning for all of your students. (HW)

PANDAS FOR ABSOLUTE BEGINNERS

Learn how to combine the pandas Python library with Google Colaboratory to make cool data visualisations

Pandas is a free and open-source Python library which provides the tools to manipulate and analyse data. Combine this with the Jupyter Notebook functionality of Google Colab and you've got a data science powerhouse tool already set up and waiting. As an educator, it is great to be able to show your students how to use code to analyse real data in the classroom. If you can find some open data about your local area (see the 'Where can I find data?' box for help with this), pandas will let you search and filter the data easily, or visualise it as a graph.

Set up your notebook

First, head to helloworld.cc/colab and create a new notebook, which is the name for the type of file you create with Colab. It's worth noting that you'll need to have a Google account to save a notebook. It is possible to use Colab without an account, via colab.to, with the caveat that you won't be able to save anything.

You will see a menu bar (which should be familiar to you if you have used Google Workspace products) and inside the document there will be a blank cell with a Play button next to it. A notebook is a series

of cells, and each one can either contain code, text, or some output; see Figure 1.

The first cell is by default a Python code cell, so type the program:

```
print("Hello world")
```

and click on the Play button; see Figure 2.

The shortcut to execute the code inside a cell is to first select the cell and then press **Ctrl/Cmd + Enter**.

Import some data

Now let's start using pandas. Delete your hello world test code, then type the following code to import the library:

```
import pandas as pd
```

WHAT IS GOOGLE COLAB?

You may be used to writing Python code in a browser, and perhaps you have already tried hosted services such as repl.it or trinket.io. Google Colab is special because it offers a cloud-based version of the open-source software Jupyter Notebooks, with no set-up required. A notebook lets you combine Python code cells with regular text, and also lets you view the output, all in one place.

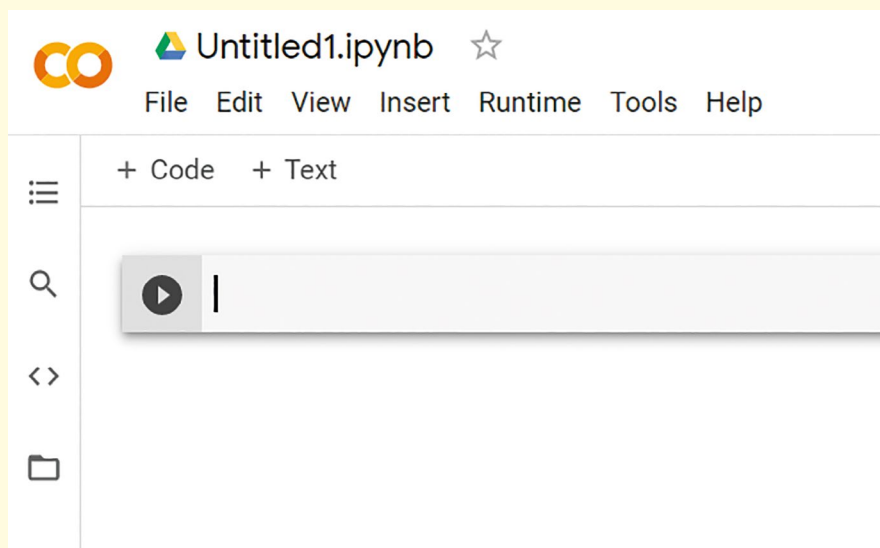


Figure 1: A blank notebook



```
print("Hello world")  
  
Hello world
```

Figure 2: A code cell with the output after it has executed

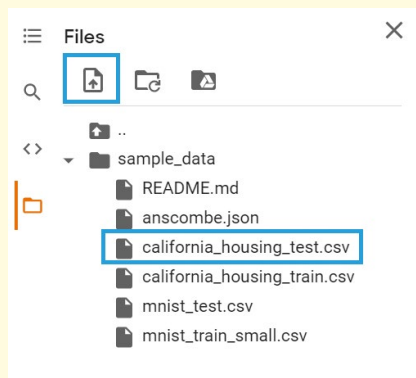


Figure 3: This is the demo data; you can also upload your own

It is customary to ask Python to import pandas as `pd`, so that you don't have to type the whole word 'pandas' each time you refer to it.

Next, let's use the CSV reading functionality of pandas to import some of the test data that is included with Colab. You can see the sample data files if you click on the folder icon in the left menu, and this is also where you can import your own data sets once you get up and running; see Figure 3.

WHERE CAN I FIND DATA?

A search for open data will often return interesting results. Local authorities in your area might publish data on all sorts of things, from felonies to flooding. If you want to try very large or quite specific data sets, the machine learning platform Kaggle has some interesting public-domain data. Government data: UK data.gov.uk: USA data.gov: Kaggle: kaggle.com/datasets

WHAT DATA CAN YOU USE? LOCAL AUTHORITIES IN YOUR AREA MIGHT PUBLISH DATA ON ALL SORTS OF THINGS, FROM FELONIES TO FLOODING

With the cursor in your first cell, press **Alt/Opt + Enter**. This will have the double effect of executing the code in the first cell to import pandas, then creating a new cell below, where you should type your data import code:

```
data = pd.read_csv('sample_data/california_housing_test.csv')
```

If you execute this new cell with **Ctrl/Cmd + Enter**, it looks as if nothing happens. This is because the data has been read successfully,

but you have not added any code to display it. Use **Alt/Opt + Enter** to add another new cell below, and type the name of the variable now holding the data (which in this case happens to also be named `data`):

`data`

Now execute that cell. As if by magic, a beautiful table called a data frame will be displayed; see Figure 4. (If your display is not so magical and shows an error message, make sure you have executed all of the cells in order; the shortcut is **Ctrl/Cmd + F9**.)

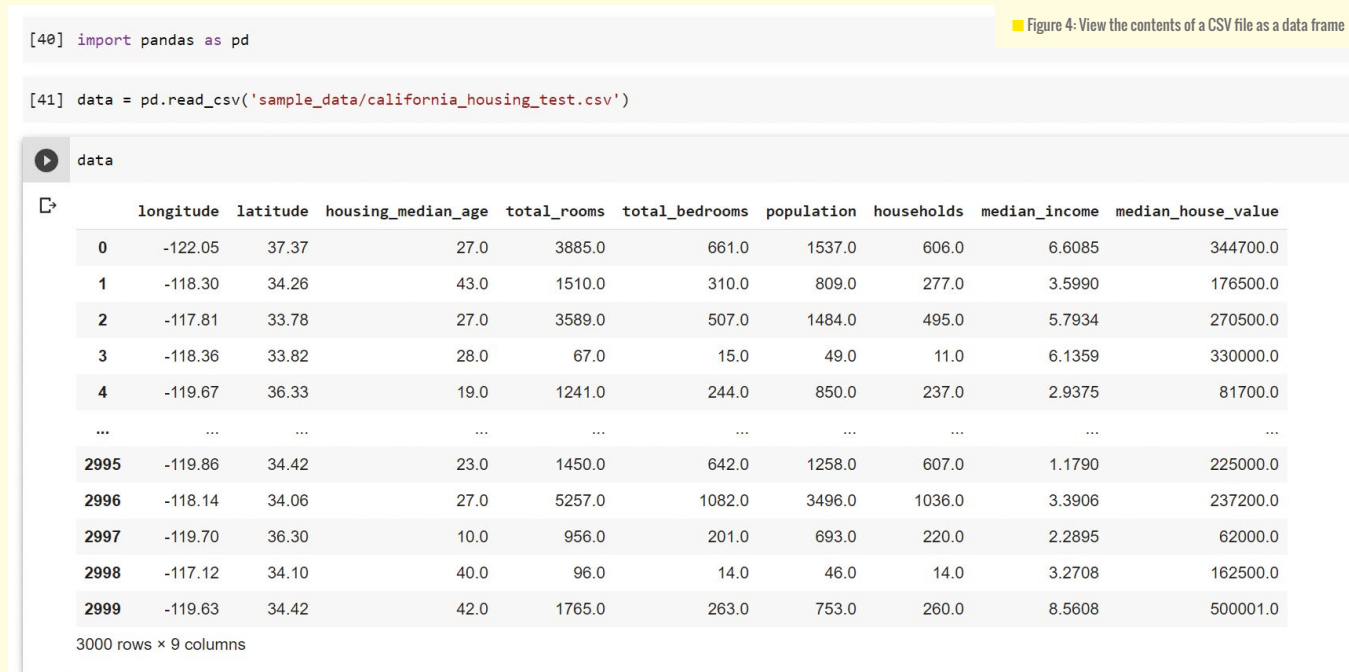


Figure 4: View the contents of a CSV file as a data frame

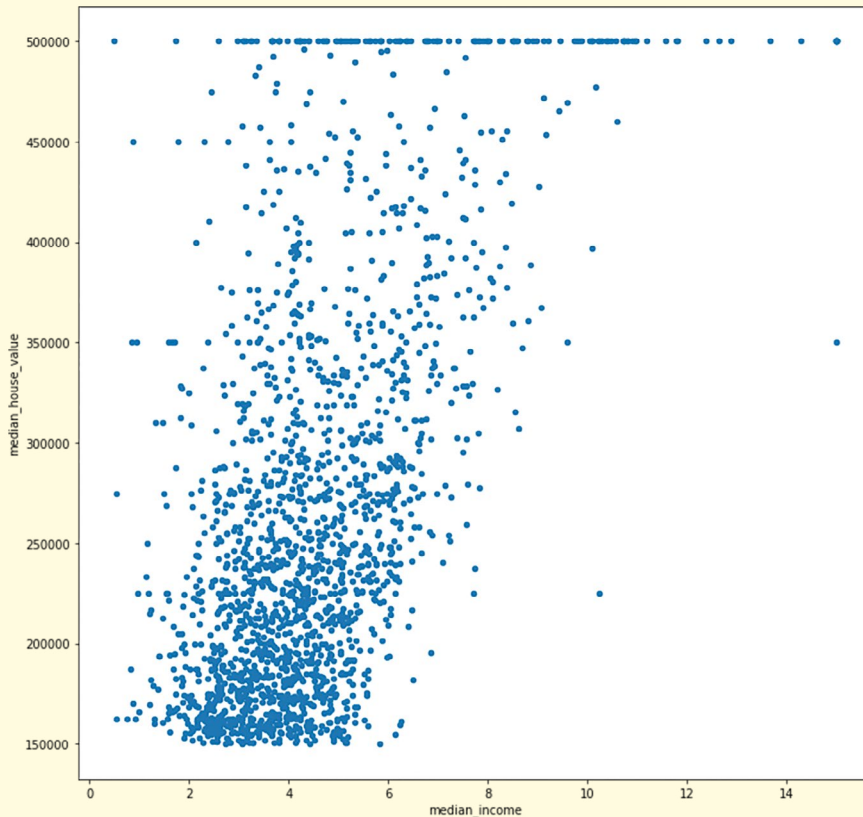


Figure 5: A scatter plot of the data

Filter the data

Now that you have some data imported, let's use pandas to filter the data. It may seem tricky to use, but if you are already familiar with SQL, you can translate between the two with a bit of practice, and the pandas documentation even has a direct comparison with some basic SQL queries to help you out (helloworld.cc/pandas).

You can refer to columns from the data by specifying the data frame and the column name; for example, to refer to the column `median_income` you would type:

```
data.median_income
```

Let's filter the data so that we see the areas with a median house value over \$150,000, and sort the data by the median income, which by default will be in ascending order. Type this into a new cell:

```
query1 = data[data.median_house_value > 150000].sort_values('median_income')
```

Breaking this down, this code means:

Show me the data where:
the `[data.median_house_value > 150000]`
and `.sort_values by ('median_income')`
(and save this as the variable `query1` so that you can refer to it later).

The equivalent as an SQL query would be:

```
SELECT * FROM data WHERE  
median_house_value > 150000  
ORDER BY median_income
```

If you want to see the output, you'll need to type `query1` on the following line to display the results. Run the cell and you should see that the data frame produced has narrowed to 1,893 rows, from the original 3,000 rows in the whole data set.

Plot a graph

Now let's plot a graph of this data to see how the median income of an area correlates with the median house value. Add a new cell and type:



LAURA SACH

Laura is a former head of department and has many years' experience teaching computer science in the classroom. She now leads the A level content team for Isaac Computer Science at the Raspberry Pi Foundation (@codeboom).

“IT'S GREAT TO SHOW STUDENTS HOW TO USE CODE TO ANALYSE REAL DATA IN THE CLASSROOM

```
query1.plot.  
scatter(x='median_income',  
y='median_house_value',  
figsize=(12,12))
```

This will create a very basic scatter plot of the data we just filtered; it will contain the data about all areas with a median house value of over \$150,000. The 'figsize' parameter in the code tells the graph to display larger, as the default size is a little too small; see Figure 5.

There are plenty of things that can be improved about this graph, and pandas is so powerful that we can't possibly list all of the things it can do. If you want to explore more, try searching for "pandas cheat sheet" or try one of the many free tutorials and courses available online.

I hope you have learnt a few fun data manipulation techniques, but at the very least you can now legitimately tell your boss, "I'm going to go and play with pandas!" (HW)



KATE FARRELL

HOW DO YOU USE DATA TO SOLVE A REAL-WORLD PROBLEM?

Gemma Coleman speaks to Kate Farrell from Data Education in Schools to discuss the importance of teaching data to help students navigate the world

When I was searching for contributors for this issue of Hello World, a pattern quickly began to emerge: “Data? You want to speak to Kate.” Kate Farrell is director of curriculum development and professional learning on the Data Education in Schools project, part of

the Data-Driven Innovation Skills Gateway in Scotland. With the project developing teaching materials, professional development, and even qualifications for schools who want to teach data education to learners aged 3–18, “It’s not the kind of role that fits easily on a business card,” she laughs.

The project started in 2019, with the team looking at the Scottish curriculum and mapping out where data could be embedded and how it could be used to support various subjects. “We know that teachers are under stress and won’t be able to deliver extra stuff, so we’re looking to understand how we get better at doing data literacy within the rest of the curriculum,” Kate explains. “How do we provide

and support opportunities to look at data in the rest of the curriculum in cool new ways?”

The team runs monthly seminars drawing upon this theme, to help teachers see its applicability across all subjects. “We like taking topics that you wouldn’t instantly think are about data science. Yes, the sciences, computer science, and maths are where you would expect it, but there are huge amounts of data and data use in geography, music, social studies, and even PE.”

One example is the DataFit series of lessons for upper primary and lower secondary students, with a mission to simultaneously increase data literacy and physical activity literacy. This includes an introduction to activity-monitoring devices, such as step counters on phones. The lesson has the twin aim of teaching students how monitoring steps or sleep activity can be a positive thing,

but also encourages them to reflect on how they feel about their phone collecting their personal data.

“A lot of students don’t realise their phone is keeping track of their step count, just by virtue of it sitting in their pockets,” Kate muses. “It’s been interesting to see just how little some learners know about the data that’s being kept and tracked about them.”

Data Education in Schools ran a similarly themed workshop for students aged 10–11, with a series of events in an imagined Data Town being examined, to investigate how data can impact our lives. The day started by giving each student a cardboard mobile phone on which they could install apps in the form of stickers if they gave the town certain pieces of information about themselves, such as their favourite colour or football team. “Some kids would just install anything, give up any data, because they wanted the stickers – just like many kids will just download any app,” Kate explains. The apps and associated products then developed as they gathered more data, which was then presented back to the students. The purpose was to get students to reflect on how they felt about the products and how they used their data.

Later in the workshop, the mayor of Data Town announced that the town had sold the data to an advertising company who wanted to know people’s favourite colour, and to a gym who wanted to know their fitness data to help them decide the location of a new branch. “This meant a series of ‘aha’ moments for students, as they realised what sharing their data meant. Some of the kids who had opted not to collect the stickers were suddenly very smug!”

The project keeps a balance in the story it tells about data, with teaching materials encompassing both the risks of data collection and the huge benefits it can bring. “That is our main aim: how can we help learners use data to make their lives and the lives of their communities better — data for social good.” In the Data Town workshop, students also chose to share data with hospitals and researchers, and later found that this had helped them to develop new medicines. “We didn’t just want to send across the message that sharing data is bad. Yes, you can share your data, but be aware who you’re

sharing it with, who you’re trusting with it.”

The materials that Data Education in Schools has produced use a framework called PPDAC: Problem, Plan, Data, Analysis, and Conclusion. This is an established approach to statistical literacy, and using this data problem-solving cycle in a real-world context

“ YES, YOU CAN SHARE YOUR DATA, BUT BE AWARE OF WHO YOU SHARE IT WITH, WHO YOU TRUST WITH IT

is a powerful way to engage learners with data topics.

“The aim is to empower students with the tools to be campaigning, to be making real-world changes to their lives and their communities using data.”

Kate gives a simple example of how a class could look at how much plastic their canteen is using, collecting the data on plastic products and then using that data to make the case to reduce their plastic consumption.

The project has also worked with Scottish exam board SQA to develop a National Progression Award in Data Science; they believe it is the world’s first data science school qualification. The award is aimed at upper secondary students, colleges, and workplaces as an introductory qualification in data science. It carries the same ethos as their materials for younger learners: to help students understand how data is used in society, both negatively and positively, and develop skills to help them make better decisions.

“I want people to realise that although data science sounds scary, it’s so important to learners’ lives these days. We’ve seen it with the pandemic. Being able to interpret and analyse data is hugely important. We need learners to be able to look at the news, and their social media stream, and question what they’re looking at, or ask: where is the evidence? This is so important, whether or not they go on to become a data scientist ... although we’d love it if they did!” (Hw)

For more information about the project and free teaching materials, visit: dataschools.education.

#INSIGHTS

DATA SCIENCE FOR
SOCIAL GOOD

STORY BY Thom Kunkeler

It is well documented that the field of computing has traditionally lacked diversity, be it in gender, ethnicity, or race, and new research has pointed to middle-school camps as a possible intervention for inclusion. In this study, summer code camps for young people were designed in Grinnell in Iowa, USA, with an emphasis on meaningful uses of computing and computing for social good. These coding camps, the researchers find, are successful in making a difference in students' sense of who can 'do' computing, and in their self-efficacy and interest.

ACTIVITIES TO TRY

Algorithms: Design an algorithm to determine the winner of the Homeless Football World Cup

Talking about data science for social good: Try paired activities in which students explore a social issue they care about, what patterns generate the problem, and how data can be used for solutions

Case study: Use pandas library in Python to explore and analyse data related to asylum seekers

Creative ways to represent data: Use data sets related to social issues such as contaminated water in Gaza, quitting smoking, or global forest loss for data visualisation

Research has suggested that there is a pipeline problem: students at a young age decide not to pursue computing because they do not have a sense of belonging to the field and its career trajectory. Stereotypes, and students' beliefs that computing can be irrelevant or asocial, further impact diversity in the field. For this study, the research team designed a code camp centred around data science for social good, in order to understand how data science and social good can be combined to positively impact students' self-efficacy and their perceptions of computer scientists. Although the participants were mostly from rural areas and lower socio-economic backgrounds, the research team went further in making sure that the camp was inclusive by offering free tuition and other no-cost options.

The students started out with micro:bit workshops, before moving on to the BlockPy web-based Python editor, and finishing with Python in Jupyter Notebook. For all the activities, the researchers used data sets that were related to social issues. These included wildlife trade and homelessness,

and they used case studies that integrated new concepts with social good data.

Both before and after the camp, the students were asked to fill out a questionnaire using statements such as "I look like a computer scientist" and "I like the challenge of computing."

The responses in the survey after the camp were on average more positive than in the first survey, indicating that the students felt more sure of their abilities as computer scientists and independent thinkers. Although girls, on average, agreed less with the statements in the precamp survey, interestingly enough, their gains were higher overall than the male campers, as discovered in the survey after the camp. For educators, this study highlights that data science and social good can be taught in a way that positively impacts students' self-efficacy and perceptions of computer scientists. You can try out some of the activities from the code camp with your students — see the boxout for inspiration.

Full details on the article and activities can be found here: helloworld.cc/codecamp. (HWO)

FURTHER READING

- ✓ Bryant, C. et al. (2019). A Middle-School Camp Emphasizing Data Science and Computing for Social Good. SIGCSE '19: Proceedings of the 50th ACM Technical Symposium on Computer Science Education, [online] pp.358-364.
- ✓ Gürrer, D. and Camp, T. (2002). An ACM-W literature review on women in computing. ACM SIGCSE Bulletin, 34(2), pp.121-127.

PRINCIPLES FOR ENCOURAGING DATA LITERACY IN CHILDREN

STORY BY Jonathan Dickins

Despite the fact that young people today are growing up in a data-driven society, there is evidence that data literacy among the general population is low. Data literacy refers to the ability to read, write, and communicate data in context. It is increasingly important for young people's employment prospects, as well as their own agency in society and their use of technology.

Classroom approaches for data literacy

A new study by Annika Wolff and colleagues at the Open University discusses approaches to teaching data literacy in a school setting. In particular, they suggest that the common approach of having children collect and analyse their own simple data sets is insufficient to give them the skills needed. Instead, students need experience with larger, externally sourced data to foster genuine data literacy.

Activity design principles

The authors propose a set of principles to support the design of activities for teaching data literacy, bringing together a number of existing principles from the research literature:

- 1. Enquiry principle:** Lead students in a guided enquiry to familiarise them with the data, before moving to a student-centred open enquiry when students are more confident asking questions themselves about the data.
- 2. Expansion principle:** Start from a small, representative snapshot of the data and then expand outwards.
- 3. Context principle:** Use data from a student's context, or data that is relatable to their lives.
- 4. Foundational competencies principle:** Foundational competencies, such as asking focused questions of data, are more important than practical or technical skills.
- 5. STEAM principle:** Learn complex data literacy skills by framing them around creative activities alongside practical ones.
- 6. Personal data collection principle:** Students should collect some data themselves, or be given activities to help them understand what data collection is like and any potential limitations.

FURTHER READING

✓ Wolff, A., Wermelinger, M., & Petre, M. (2019). Exploring design principles for data literacy activities to support children's inquiries from complex data. *International Journal of Human-Computer Studies*, 129, 41-54.

An incremental approach to developing resources

The authors proposed these principles after studies in which, alongside practising teachers, they created initial lesson plans relating to smart city data that were adapted and refined each time they were taken to a new classroom. For instance, one lesson plan involved using data from smart meters in 70 homes, with the aim of guiding students to identify common energy consumption patterns, and to see how they differ between households or days. These were

“ DATA LITERACY IS INCREASINGLY VITAL FOR YOUNG PEOPLE'S OWN AGENCY IN SOCIETY

LESSON OUTLINE USING THE PRINCIPLES FROM THE STUDY

Aim: Use aerial photography data to demonstrate that houses differ in their potential for producing solar energy.

- 1. Enquiry:** A guided enquiry stage looks at photos of roofs and their potential yield
- 2. Expansion:** Learners use an online map to search the city and ask and answer their own questions
- 3. Context:** The images were from the school's local area, and students were encouraged to find their own house

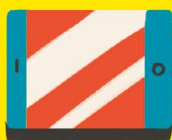
Intended outcomes

1. Understand how roof size, pitch, and direction affect solar yield
2. Understand the use of different types of data
3. Find errors in data and understand its limitations

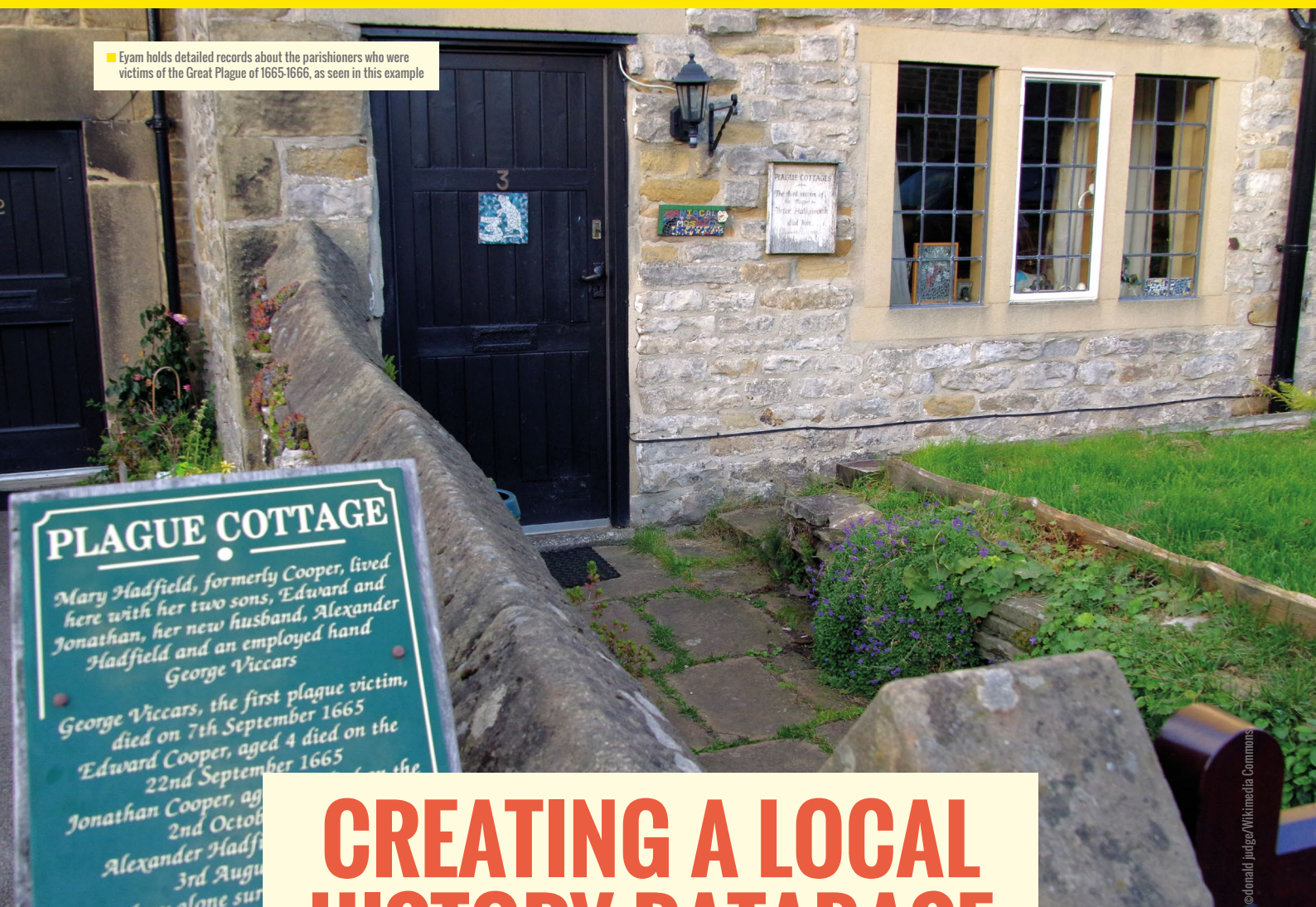
tested with teachers, with incremental improvements identified between tests.

The authors found that more support is required for younger students in framing their enquiry questions from externally sourced data, perhaps because they found it more difficult to link their questions to the data and to generate explanations. They also found that students were able to be more critical of the data when they assumed the role of data collector, and suggested that allowing students to collect some data themselves helped them to understand its limitations.

See the boxout for an example of a data literacy lesson outline designed according to some of these principles. (HW)



■ Eyam holds detailed records about the parishioners who were victims of the Great Plague of 1665-1666, as seen in this example



CREATING A LOCAL HISTORY DATABASE

Katharine Childs describes how her pupils aged 7–9 used local history data to create a flat-file database about the villagers of Eyam in the UK

Creating a database of local history information can neatly link computing and history together in the primary curriculum. In Derbyshire in the UK, the village of Eyam is well known for the decision of inhabitants to self-isolate during the Great Plague of 1665–1666. Many lives were saved elsewhere by halting the spread of the plague, but almost a third of the residents in Eyam died. This collective act of bravery was included in my school's topics for Years 3 and 4 (ages 7–9) and so I planned a series of lessons to create a database of plague victims.

Databases unplugged

The lessons started with some unplugged activities to help pupils learn more about why databases are useful. The class could quickly sort themselves into height order, and found it reasonably easy to sort themselves alphabetically by first name. Sorting by less visible data, such as date of birth, was a lot slower. We talked about how computers can help to speed up tasks such as these.

Local records provide a detailed list of the 273 plague victims in Eyam. We created a paper database of this data by making index cards of each person (see the

image on the next page). I split the full list into sections and allocated each section to a different group. This helped with learning some key database vocabulary. Each card was a record and each specific piece of data was held in a field. As the class worked, they organically voiced questions to investigate further, such as “Did more men or more women die?” and “How many children the same age as me died?”

Creating the database

We then created an electronic database using 2Investigate, which is part of the Purple Mash suite of software. Pupils

USING LOCAL HISTORY DATA SOURCES

Finding local history data sources is often easier as a collaborative task. Asking school governors and parents, or an enquiry to the local archives, may turn up information. It's also important to remember that local history is different for everyone, and that historical data records often contain bias towards a single gender or ethnicity. Some examples of other data sets that could be used in this task include:

- A passenger list from Windrush in 1948
- A list of suffragettes arrested from 1906 to 1914
- A list of parishioners derived from the 1911 UK census

“**LOOKING AT
PLAGUE DATA
IN COMPUTING
HELPED THE CLASS
GRASP THE SCALE
OF ITS IMPACT**”

worked together in pairs to input the data from the cards, with one child reading aloud the data and the other child inputting it into the database. We paid close attention to one key difference between paper records and digital records: on the index cards, the fields could be populated with any type of data, but in our 2Investigate database, each field had to be assigned a type: letters, numbers, or a drop-down menu of specified options. Because all the work was saved to the school network, I was able to merge the different databases together and create one whole-class database. Finally, we used the database to find out the answers to our initial questions.

Eyam Plague victims

First name:

Surname:

Male / Female:

Date of death:

■ Index cards helped pupils to connect to the idea of a record in a database

In English, the class had been writing diaries from the point of view of a family living with the plague; in history, they had been looking at the causes of the plague and the beliefs that people had about its origins. Computing lessons gave them an understanding of the scale of the impact on the village and helped them to piece together all the individual stories to form an astonishing narrative. They also came to better understand the importance of computing skills in the real world.

A natural next step from these lessons would be to go on to create visualisations of the data, which you can read all about in the next article, *Data visualisations for inquisitive minds*, by Ben Garside. (HW)



KATHARINE CHILDS

Previously a network manager and a primary computing classroom teacher, Katharine now works in the Impact & Research Team at the Raspberry Pi Foundation (@IAmKatharineC).



DATA VISUALISATIONS FOR INQUISITIVE MINDS

How to get your students started with data science by introducing exciting and interesting visualisations

Visualising data is the art of being able to not just present data in a visual format, but to truly tell a story with it. A good visualisation will bring the data to life, with the aim of providing the audience with something that they might not have been able to spot by just looking at numbers on a page. While writing a unit on data science for the Teach Computing Curriculum, I became infatuated with the creative ways in which people have visualised data, drawing out trends, correlations, and patterns. In this article, I want to highlight some of the best visualisations that are adaptable to most age groups, that you can use to inspire your classes and get them thinking about data.

Telling a story with data

Take a look at the data that you can see in Figure 1 below. Can you extract any meaning from it? What story is it telling you? You can inspect this data in more detail here: ncce.io/minard-data.

The data is from Napoleon's 1812 march on Russia. The numbers alone don't tell much of a story, but in 1869 Charles Joseph Minard, a French civil engineer known for his information graphics, produced what is widely regarded as the best statistical graph of all time; see Figure 2.

The visualisation shows Napoleon's army, departing at full strength from the Polish border to Russia, and then their subsequent retreat. The thickness of the lines represent the size of the army, with

the beige line representing the march on Moscow and the black line representing the retreat back to Poland. The scale of the losses becomes very clear if you simply compare the thickness of the two lines. Napoleon set off with 422,000 troops and returned with just 10,000.

Other forms of data have also been represented in the visualisation. Geographical features such as locations and rivers crossed have been plotted, as well as the varying temperatures at different points on the march and retreat. Upon close inspection, this helps you to see where and why tragedy occurred at various points on the journey. During the retreat on 28 September 1813, the data labels on the black line read 50,000, before dropping to 28,000, showing that 22,000 men died crossing the Berezina river near Minsk!

What I love about this visualisation is that the more you look at it, the more you start to see. The visualisation really made me feel as though I understood a little more about the story of Napoleon's march on Russia — a story that I certainly wouldn't have been able to get from simply studying the data as numbers on a page.

Effecting change with data

As well as telling a story with data, visualisations can be starting points to bring about change or to help gain support for a cause. The next historical example dates back to 1854, when there was an outbreak of cholera in the Soho area of London. At the time, it was a commonly held belief that cholera was being caused by bad air in the area. A physician named John Snow held an alternative belief that cholera was being transmitted by a contaminated water supply. To help prove this theory,

Long	Lat	City	Temp	Long	Lat	Survivors	Direction	Division
24	55	Kowno	0	24	54.9	340000	A	1
25.3	54.7	Wilna	0	24.5	55	340000	A	1
26.4	54.4	Smorgoni	-9	25.5	54.5	340000	A	1
26.8	54.3	Molodexno	-21	26	54.7	320000	A	1
27.7	55.2	Gloubokoe	-11	27	54.8	300000	A	1
27.6	53.9	Minsk	-20	28	54.9	280000	A	1
28.5	54.3	Studienuka	-24	28.5	55	240000	A	1
			-30	29	55.1	210000	A	1

Figure 1: What does this data tell you?

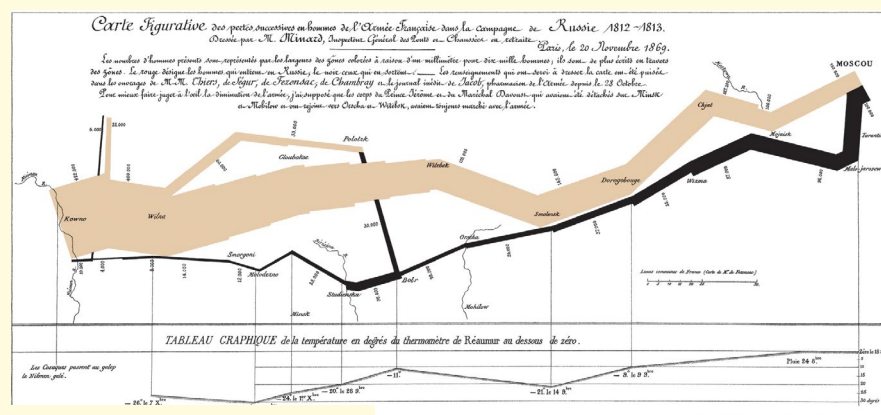


Figure 2: Statistical graph of Napoleon's march on Russia



he mapped the deaths from cholera in the Soho area. The map clearly showed that the deaths were centred around Broad Street, and the residents there were getting their water from the pump on this street.

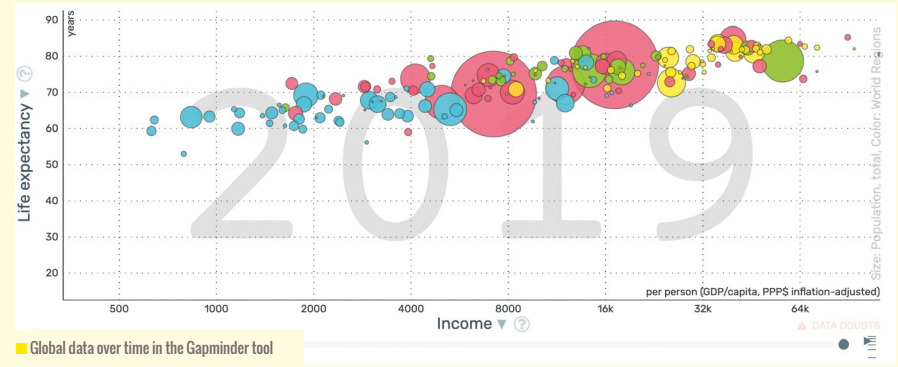
By visualising the data in this way (now known as a dot map, a map type that uses a dot symbol to show the presence of a feature or phenomenon), Snow was able to convince the local council to disable the water pump. It is widely recognised that this visualisation helped to save many lives.

Getting hands-on with interactive visualisations

Moving from the past to the present day, the final point that I want to highlight is the use of a free visualisation tool by Gapminder (helloworld.cc/gapminder). This software allows you to view global data and see how it has changed over time. The default data in the bubble graph compares life expectancy with income. By using the slider and clicking on the play button, you can see how this has changed from the year 1799 to the present day.

The data on the x and y axes is customisable, allowing you to compare a huge range of other factors, such as CO₂ emissions and educational standards. I like this tool, compared to the historical examples I've discussed, as it is much more hands-on for students and presents opportunities to discuss concepts such as trends, correlation, and outliers.

After allowing students to see the data available to them, a good activity would be to get the class to think about what factors make



a country a great place to live. They can then use the Gapminder tool to investigate which country best fits their ideals.

Practical suggestions for the classroom

As suggested, the Gapminder tool is a good place to start to get students exploring data sets. You can also use the Turner's Graph of the Week website (helloworld.cc/turner) to reinforce the idea of using visualisations to spot trends and correlations, as well as to make predictions and gain insights. The site provides free worksheets that include visualised data and sets of questions that prompt analysis of the data, perfect for in-class activities and homework.

I'd recommend allowing time for your students to explore visualised data sets on the Information is Beautiful website, which visualises data talking points such as riskiest activities during the pandemic, and how much music streaming services pay their artists (helloworld.cc/information). Another favourite of mine is Dear Data (helloworld.cc/deardata), a project between two information designers to collect data about their lives each week

and visualise it on a postcard sent to each other. A nice follow-up activity to viewing the Dear Data website would be to get students to spend a week collecting data about their own lives and visualising it. For example, they might monitor their mobile phone or app usage, or what food they've eaten. Once they have completed the activity, get them to write a short report on what they have learnt from their visualised data and how they might use the information to make changes in their lives.

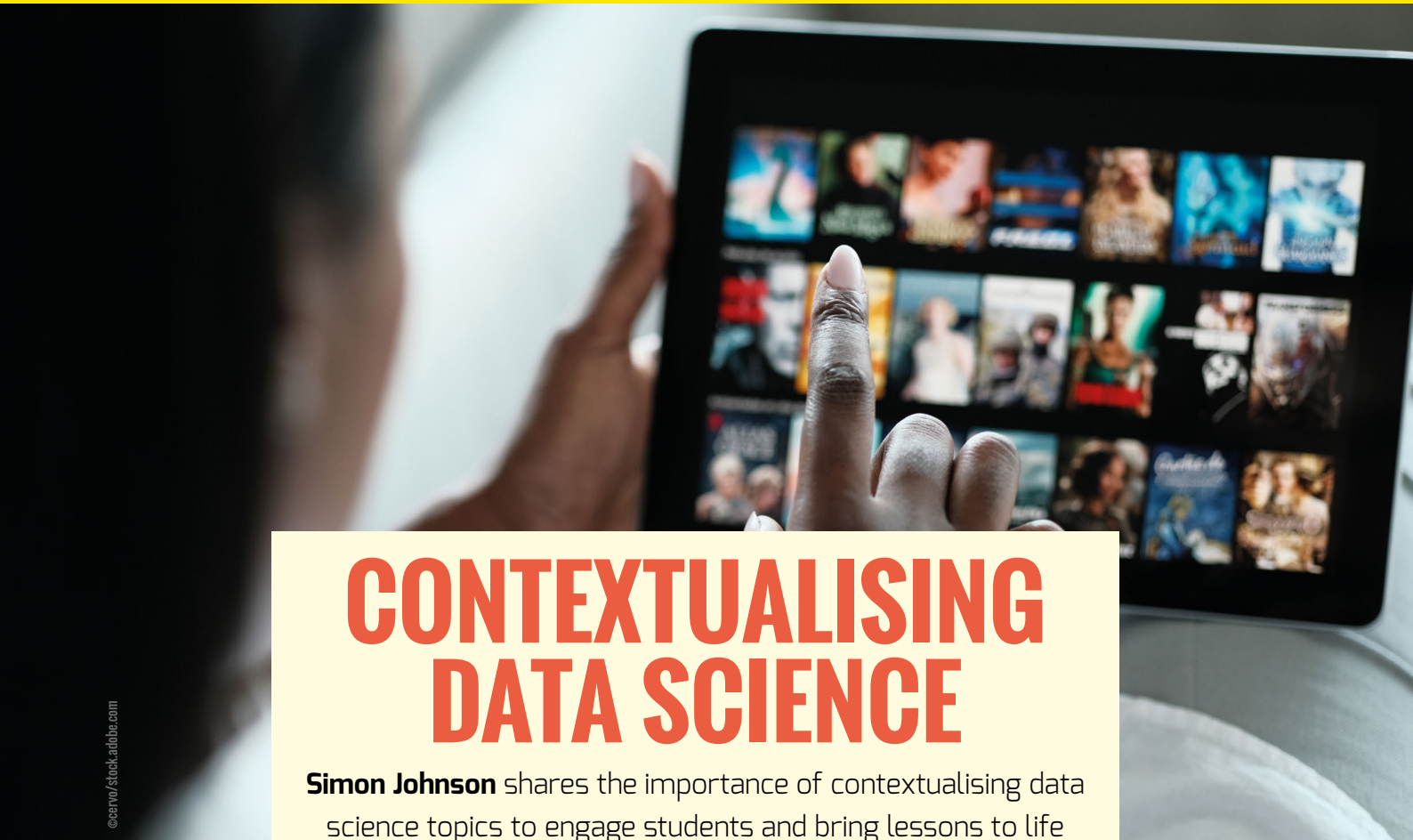
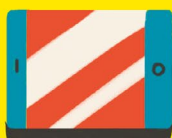
Finally, data science is not just the remit of computing. Explore the cross-curricular opportunities that you could take advantage of when teaching visualisations. The examples in this article have clear links with history, geography, and maths.

Using some of these examples and tools, I hope your students will have understood that visualisations can be much more creative than the traditional bar and pie charts we're all used to seeing and the power of data to tell a story, share information, and bring about change. (HW)



BEN GARSIDE

Ben works for the Raspberry Pi Foundation and is also a Computing at School community leader. Recently, Ben has been developing resources for the Teach Computing Curriculum, as well as writing online courses and content for Isaac Computer Science (@BenberryPi).



CONTEXTUALISING DATA SCIENCE

Simon Johnson shares the importance of contextualising data science topics to engage students and bring lessons to life

Data science is the study of data. It involves developing scientific methods, processes, and algorithms to extract knowledge from unstructured data in order to provide insights and support decision-making. In this article, I share the importance of bringing context to data-themed lessons — something that is important in all topics and subjects, but can be particularly helpful for what can occasionally be a dry topic.

Exploring data science in the classroom

When introducing pattern recognition, I often use the John Snow cholera map outlined in the previous article. While I find this activity is generally very well received by students, the activity does have its issues. The first problem I come across is one that I find with most unplugged lessons. Now don't get me wrong; I'm a huge evangelist for the unplugged approach, but some students find it too abstract, and struggle to find a connection between the activity and the concepts being taught. The other problem is one suffered by

most other subjects: that it has little context for the students. To paraphrase a student during one of my lessons: "Sir, why are you teaching us history in a computing lesson?"

Context is key to learning

The most common questions you will be asked by students when undertaking a new topic or subject are:

- Why are we doing this?
- Will this help me with my exam?
- When will I ever use this?
- Will this help me get a job?

In other words, how is it relevant?

Contextualisation is not about changing the learning outcomes or objectives. It's about modifying the learning materials so that they are relevant to the students. When teaching a new topic, such as data science, ask yourself the following questions:

- Does this relate to students' aspirations or common interests?
- Can it help students to understand how computing will be relevant to their future career choices?

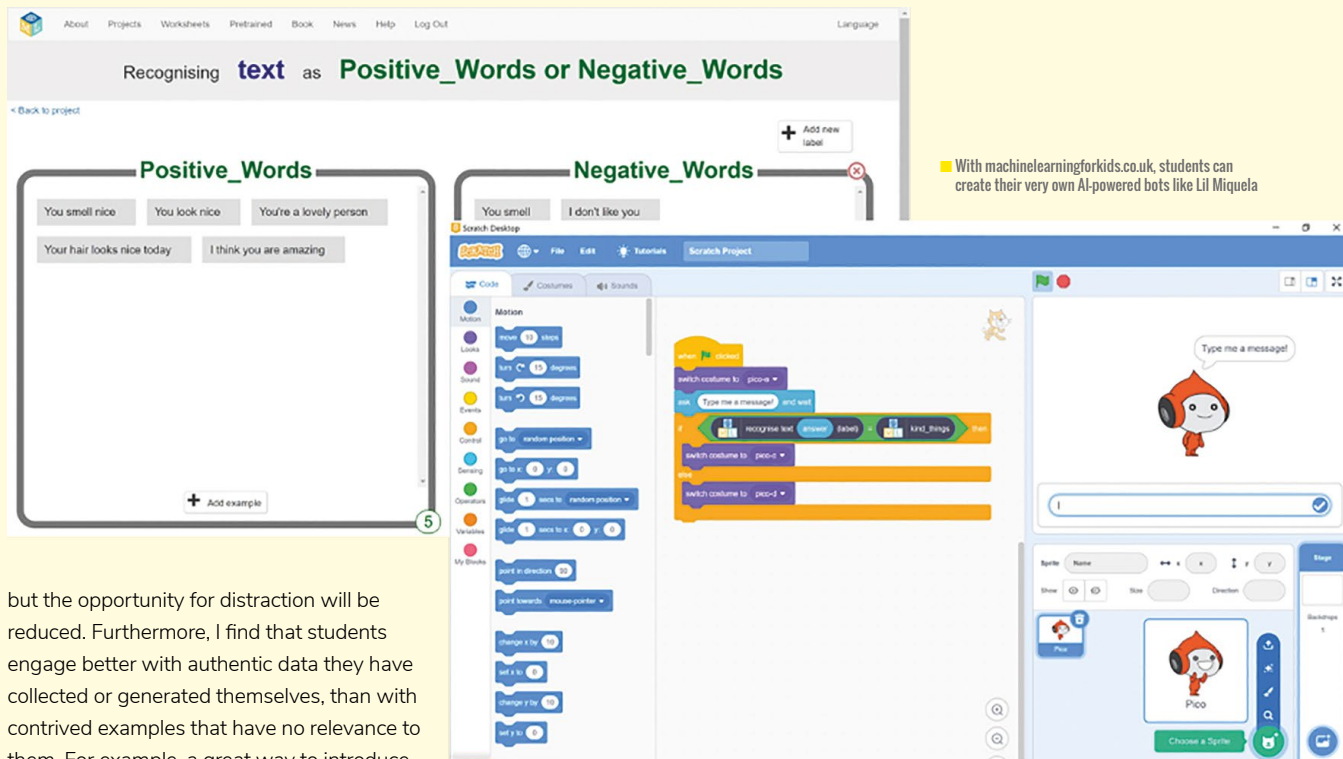
- Does it involve skills that can be used in the workplace?
- Can it be linked to popular culture or a recent newsworthy event?

Let's now explore some of these questions.

Does it relate to students' interests?

When a topic connects with what students are interested in, engagement increases. However, it is prudent to understand the distinction between a student-centred approach, where students can explore their own interests, and a contextualised approach, in which the students are connected to a topic through a common interest.

Using a more student-centric approach can often lead to students focusing on the interest, rather than on the activity. Case in point: when creating a magazine cover as part of an image editing lesson, my students would often be preoccupied with finding pictures of the subject rather than focusing on the editing skills. However, I find that by connecting the learning to a common interest, engagement will still improve,



With machinelearningforkids.co.uk, students can create their very own AI-powered bots like Lil Miquela

but the opportunity for distraction will be reduced. Furthermore, I find that students engage better with authentic data they have collected or generated themselves, than with contrived examples that have no relevance to them. For example, a great way to introduce students to data analytics is to have them create infographics based on data they can collect about their own social media habits or mobile phone usage.

Does it involve skills that can be used in the workplace?

"When I grow up, I want to be a data scientist," said no one ever! But, at its heart, data science is all about solving problems. The ability to think analytically and to recognise patterns are not just key computational thinking skills; they are also valuable skills for any workplace.

In a 2009 interview, Google's chief economist, Hal Varian, stated, "The ability to take data — to be able to understand it, to process it, to extract value from it, to visualise it, to communicate it — that's going to be a hugely important skill in the next decades." Fast-forward to 2021 and many businesses would agree with Varian's prediction — a sentiment echoed by LinkedIn's 2020 Emerging Jobs Report, which found data science roles proliferating across nearly every industry.

And while some applications of data science, such as self-driving vehicles, might be fairly obvious, there are some less obvious examples that are very relatable and which you can use to draw students into

the subject. For example, Liverpool Football Club's recent record-breaking run owed a lot of its success to data science. The club had been collecting and analysing exhaustive amounts of data in order to dictate the style of play and determine who should play — all managed by a dedicated team of researchers, statisticians, and data scientists.

Can it be linked to popular culture or a newsworthy event?

Data science plays an integral part in our everyday lives; it's just not always obvious where it's being applied! It's no secret that companies such as Amazon, Facebook, Netflix, and Spotify all use AI, machine learning, and data science to interact with the end user to influence their content choices. What might be a surprise to students, though, is that the very same technology is used to create those infamous Tom Cruise deepfake videos (AI-generated fake videos) on TikTok, and it's also behind the faces of popular AI influencers such as Lil Miquela!

Now, thanks to free AI teaching initiatives like machinelearningforkids.co.uk, students can create their own AI-powered projects, using Scratch or Python, while also exploring the power of big data. By explaining that these topics and tasks are building the skills

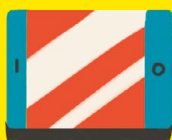
that are behind technology that they engage with and enjoy, you are much less likely to hear, "Why are we learning this?"

By making computing relevant and providing a real-life context, you can create meaningful learning experiences for your students. So, what will you do to contextualise learning? (HW)



SIMON JOHNSON

Simon is an experienced teacher of ICT and computing. He is a former network manager, and currently a member of the Computing at School working group and management board. He is a Level 2 CAS Master Teacher, Microsoft Innovative Expert Educator, Raspberry Pi Certified Educator, and Microsoft Teacher Trainer.



DATA SCIENCE JOBS TO INSPIRE STUDENTS

Three individuals explain how they use data science in their jobs and share tips for students interested in following a similar path

As Simon Johnson explained in the previous article, providing students with context about why they are learning something is a valuable

engagement tool. Here, we profile three individuals who use data science skills and knowledge in their jobs. You can use these profiles to help inspire your

students when introducing data topics, helping them to understand the variety of career paths they could take with a grounding in data science.

THE GAMES PRODUCT ANALYST

Name: Daniela Fontes

Job role: Product analyst at Star Stable Entertainment, Stockholm, Sweden

What did you study and where: MSc in computer engineering with a major in AI and multimedia from UTL, Instituto Superior Técnico, Lisbon, Portugal

Describe your job and how you use data science:

In my role, I work closely with game and product teams, in order to help them understand and monitor current performance, as well as optimising based on quantitative information. It starts with having a good understanding of the game and the business. Other common tasks involve:

- Making sure that the team is collecting the necessary data (analytics specification/ data taxonomy)
- Querying existing data sources, performing analyses, forecasting (here it's of the utmost importance that you have a good data science toolbox and know which tool to use where!)
- Designing experiments

- Presenting findings, reports, and results of experiments
- Improving my team's data literacy

What do you enjoy about your role: Being a product analyst allows me to combine my passion for business and strategy with analytical skills and communication. I get to work on continuously improving games and products, with the goal of creating the best user experience. Last but not least, I get to work with amazing people from many different backgrounds.

Do you have any advice for students who are interested in a similar career path: Game teams often work like start-ups. At the end of the day, your mission is to add value. As an analyst, you should always be thinking about the return on investment (ROI) of your initiatives. However, you should also block time for being creative by exploring and learning. Every analyst I have met has a different story and path, but at the end of the day, all analysts have a combination of hard analytical skills (programming, statistics, and so on) and soft skills (communication, writing, etc.).





THE LECTURER AND RESEARCHER

Name: Ciira Maina

Job role: Senior lecturer at Dedan Kimathi University of Technology, Nyeri, Kenya

What did you study and where: Electrical engineering at the University of Nairobi, Kenya (BSc) and Drexel University, USA (PhD)

Describe your job and how you use data science: As part of my job at the university, I lead the Center for Data Science and Artificial Intelligence, where we apply data science to various problems we are trying to solve. Some examples include working on an environmental monitoring project, where we designed a system to collect acoustic data from nature to try and determine ecosystems that are threatened. We do this by using machine learning to classify bird sounds in the recordings and then

use this information to see if there are changes in birds species' presence. This is a direct measure of ecosystem degradation. Another project involves monitoring rivers in Kenya to try to understand the state of the catchment areas. For example, if the catchment has been degraded and trees have been cut, there is increased surface run-off, which we are able to detect with our sensors.

What do you enjoy about your role: We are able to apply our skills to areas such as environmental conservation and health, which could directly increase quality of life.

Do you have any advice for students who are interested in a similar career path: Discover your passion, and find problems that you can work towards solving.



THE MACHINE LEARNING PHD STUDENT

Name: Minttu Alakuijala

Job role: PhD student in machine learning at Google, Inria, and École normale Supérieure, France

What did you study and where: Computer science, University College London

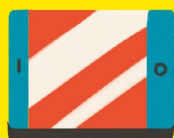
Describe your job and how you use data science: I do research in machine learning to make robots a little smarter and more helpful. There are two main ways to do this: we can teach robots with demonstrations (supervised learning), or we can let them learn on their own through trial and error (reinforcement learning). My research work combines both ways, giving just enough demonstrations to get the robot started, and letting it improve itself autonomously. For now, our robot arm can solve simple tasks, like pushing cubes around, picking them up, and placing them in a bowl. However, the advantage of using machine learning instead of classical robotics is that learning methods are very general and could be applied to any task that we can collect data about!

What do you enjoy about your role: As a PhD student I have a lot of freedom to work on interesting research questions. My research is exciting because it has so many applications in the real world. Who wouldn't like a robot assistant to help with chores at home? Even though Roomba vacuuming robots already exist, there is so much more to do in this fast-growing field! In particular, using data science and machine learning to control robots is really useful in that it allows people without a background in control engineering to teach robots.

Do you have any advice for students who are interested in a similar career path: If you think you might enjoy delving deep into a topic and studying it in a lot of detail, research is a great career choice. It allows you to be creative and think outside the box, constantly learn new things, and follow your curiosity. There are many research positions in data science and machine learning in companies, too, so you have options in both academia and industry. If you are interested in pursuing this path, I would recommend getting

research experience as early as possible, as this will be an advantage when applying to master's, and especially PhD, programmes. Some ways to get involved during your undergraduate studies, or even earlier, could be taking a summer job as a research assistant or contacting a professor to propose a project of your own.





BRINGING BIOINFORMATICS TO SCHOOLS

Biological research has become rapidly more data-driven and digital, and university researchers in Scotland have designed workshops for schools that combine biology and computing

Rapid advances in technology over recent decades have changed the nature of life sciences research. Biologists are increasingly required to develop and use computational tools to work with and interpret biological data. Genomic sequencing, in particular, has led to the generation of such large data sets that computers are essential for their storage, management, and analysis. Almost all areas of the life sciences benefit from knowledge of DNA and protein sequences — including evolutionary biology, cancer biology, ecology, animal breeding, and conservation, to name but a few. Bioinformatics is an interdisciplinary field that combines aspects of biology, computer science, mathematics, and statistics. The field experienced rapid expansion in the mid-1990s, driven

largely by the Human Genome Project, and is now an essential element of modern biology. This has generated demand for interdisciplinary researchers with strong computational skill sets.

Despite this need, biology education has failed to keep pace with advances in research. In schools, computing is often taught in isolation from other sciences, and its importance in biological research has not yet been fully realised. This leaves biology pupils unprepared for the computational component of higher education and subsequently, research in the life sciences. The 4273π Bioinformatics at School project aims to address this issue by designing and delivering curriculum-linked, hands-on bioinformatics workshops combining biology and computing for secondary school pupils.

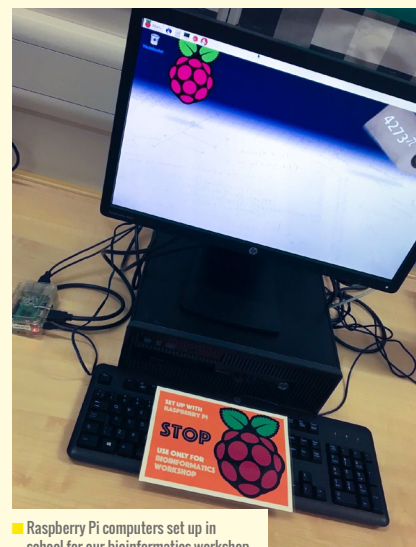
At the present time, our project has reached 166 schools and colleges across Scotland through in-class workshops and teacher CPD. Although our primary focus is Scotland, our open educational resources are used internationally (elsewhere in the UK, Mexico, Argentina, India, USA, Finland, Brazil, China, the Netherlands, and Saudi Arabia). All of our resources are freely available at 4273pi.org.

Our workshops

We use biological case studies based on DNA sequences as an entry point to engage young people with computer science. We have two workshops, one for senior biology pupils (aged 16 to 18) (helloworld.cc/workshop1) and one for pupils at an earlier stage (aged 13 to 16) (helloworld.cc/workshop2). The senior

GETTING STARTED WITH BIOINFORMATICS

- Bioinformatics is an ideal field for collaboration between science and computing teachers. Work together through our workshops to engage pupils with the different skills and knowledge required, or develop your own case studies. Lesson plans can be found on our website: 4273pi.org/teacher-resources.
- Make use of the free, publicly available online biological databases (see our introductory videos at helloworld.cc/4273videos). These databases are accessible on any device with internet access, e.g. laptops, tablets, and smartphones.
- Use case studies that support the curriculum but that also represent everyday scenarios, e.g. health.
- Encourage active learning and working in pairs.
- Use research-grade, freely available bioinformatics software for Raspberry Pi, e.g. the command line version of the NCBI sequence database search tool, BLAST. Thanks to work by the Debian Med project and others, many of these are available as Debian packages that can be installed on Raspberry Pi OS using apt-get.



■ Raspberry Pi computers set up in school for our bioinformatics workshop



■ Resource package for teachers attending our current series of free online CPD events

workshop introduces the Linux command line. For many pupils and teachers, this is their first experience of command line computing. In order to provide the Linux operating system and the tools required to perform bioinformatics at the command line, we bring Raspberry Pi computers to schools. These are preinstalled with all required software and files, and do not require internet or school network access.

Secondary school science curricula vary across the world. In Scotland,

workshop with computing classes. We ran a joint session with a computing class and a biology class, in which pupils worked through the exercises in pairs comprising one computing pupil and one biology pupil. We are continuing to develop our workshops to maximise their applicability across subjects. Bringing bioinformatics to computing classes provides a real-life example of the power of computing in research — for example how bioinformatics algorithms and biological databases can be used to investigate biodiversity and the evolutionary relationships of species. These case studies provide a practical insight into the significant role computation plays in modern biology, and demonstrate how pupils' skills can be used in this field.

Engaging more girls with computing

In Scotland, around 66 percent of biology pupils identify as female, in contrast to only around 16 percent of pupils in computing classes (helloworld.cc/SQA). This disparity exemplifies the startling gender inequality associated with computer science in the UK and many other countries. Therefore, targeting biology classes with our computational workshops allows

the curriculum is important. Bioinformatics is used across many areas of life sciences and so it is possible to create case studies that cover specific areas of the curriculum. Many pupils in both age groups reported that they enjoyed working in pairs or small groups to complete activities. We believe this creates a collaborative and relaxed environment when learning a new subject, as well as being good preparation for further studies and work. Feedback also shows that using case studies provides real-life context to abstract scientific concepts, making the activities feel more relevant to both pupils and teachers.

Our project aims to highlight the key role that computing plays in modern biological research. Computing is far from an insular subject, and our bioinformatics workshops offer pupils and teachers the opportunity to gain practical experience in this interdisciplinary field. (HW)

BRINGING BIOINFORMATICS TO COMPUTING CLASSES DEMONSTRATES THE POWER OF COMPUTING IN RESEARCH

bioinformatics is included in the senior level biology syllabus; however, our workshops also incorporate other key topics covered in most secondary school biology courses, such as DNA sequencing and evolution. Our senior workshop uses a gene involved in vitamin C biosynthesis as a case study to explore mutations and nutrition, and our junior workshop uses DNA barcoding to investigate food fraud. Both use freely available online research resources: the National Center for Biotechnology Information (NCBI) database and its BLAST tool. These can be accessed on any device with an internet connection.

Bioinformatics is interdisciplinary by nature, so we have also trialled our senior

us to reach an audience not necessarily engaged with computing. We also conduct whole-class visits during the school day to avoid the potential audience bias and exclusion that may occur through self-selection for extracurricular activities. Importantly, feedback from pupils showed that there were no significant differences in enjoyment and usefulness of our workshops between the genders (helloworld.cc/workshop3).

Pupil and teacher feedback

Evaluation of feedback from teachers and pupils has highlighted several key elements that underpin the success of our workshops. For teachers, a strong link to



STEVIE A BAIN

Stevie is a postdoctoral research associate in bioinformatics education at the University of Edinburgh, UK (@steviebain_).



DANIEL BARKER

Daniel is a reader in bioinformatics at the University of Edinburgh, UK (@4273pi).



MORE THAN THE MEAN: THE IMPORTANCE OF GRAPHING

Using statistical measures such as the mean is important for understanding data — but they can hide just as much as they reveal

The average planet in our solar system has zero, zero, or about a billion people on it, depending on whether you take the median, mode, or mean. Obviously none of these figures quite captures the real situation, with all of the population living on (or just above, in the case of the International Space Station) just one of the eight planets — and nor does calculating the range or the standard deviation. What else could these statistical measures be hiding, that could be uncovered by drawing a graph?

Much of the time, using averages and measures of spread such as the interquartile range and standard deviation can be useful for getting a quick feel for a set of data, or to easily compare two different sets of data. Whether it's results between different classes in your school, medical outcomes, or tests being run on different prototypes of a new product, if two sets of data have different means and standard deviations, you can use these

to describe some of the key differences between those sets.

However, some of your students may think that the definite and exact values they get from their calculations are more important than something as open to interpretation as the shape of a graph. Luckily, the real world provides examples of collections of data that behave very differently, which becomes obvious when you graph your data.

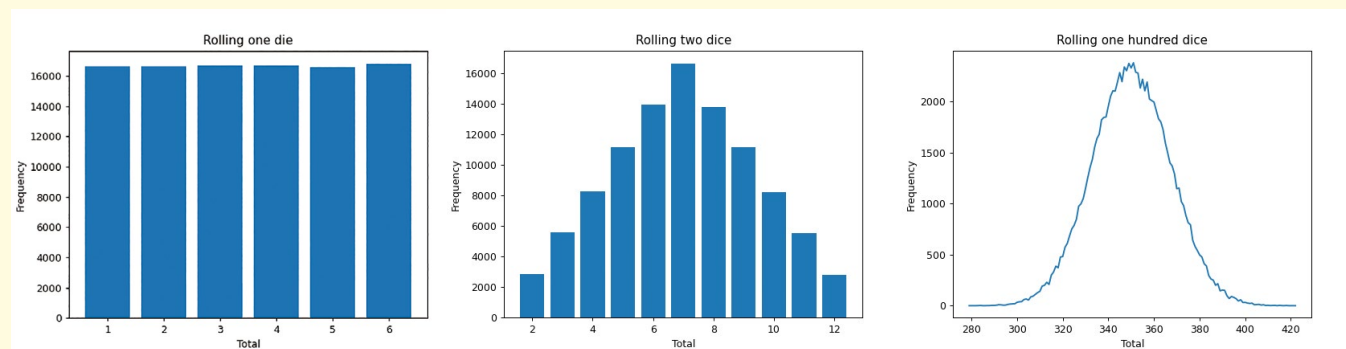
Finding the shape of data

An activity you might use in your classes when looking at data and statistics is to look at the outcomes of rolling a fair standard six-sided die. In this case, there's an equal chance of rolling any of the numbers, leading to a flat distribution if you graph it. The mean and median values of a large number of dice rolls will be 3.5, while there isn't a single modal value.

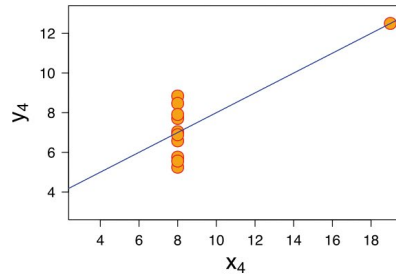
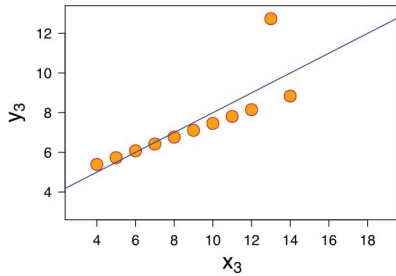
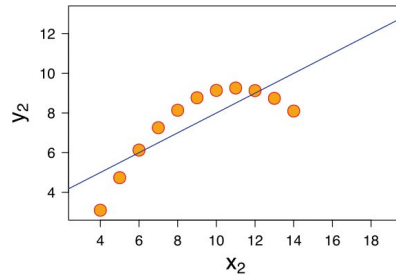
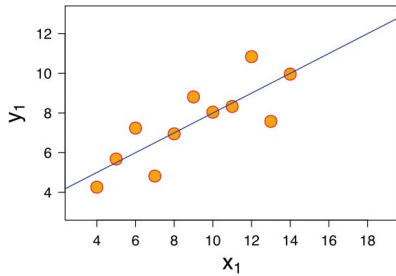
Start looking at the rolls of a pair of dice, however, and the graph changes shape

significantly. Although the mean doubles to 7, the graph is no longer flat; there is a peak at 7 and the chances of rolling each value decreases as you move away from this, towards 2 or 12. As you increase the number of dice you are rolling at once, the graph moves towards a commonly observed shape, a curve known as the normal/Gaussian distribution, or bell curve. You can find this same shape right across the sciences, whether you are looking at how far molecules have diffused after a given time, the sizes of animals, or random errors in an experiment. It's important for everyone to appreciate the long tails of these graphs — even though most of the data will fit inside the main body of the curve, there's plenty of space further away where you might find a rarer specimen.

Unusual shapes can also help you to spot interesting areas to study — or at least to tidy up your data. Sets of data can often include outliers, points that don't seem to fit with the rest of the distribution,



Common graphs for rolling one, two, or 100 dice



■ Anscombe's quartet: all four sets of data share the same set of statistical measures, but plot very differently on graphs

“ UNUSUAL GRAPH SHAPES CAN HELP YOU TO SPOT INTERESTING AREAS OF STUDY – OR AT LEAST TIDY UP YOUR DATA

and drawing a graph of your data can help you to spot these. These could be due to experimental error, or they could demonstrate an unusual or rare effect. Either way, these outliers could be worth investigating, even if the end result is that you remove them so that your remaining data more accurately represents what you are investigating.

Same stats, different graphs

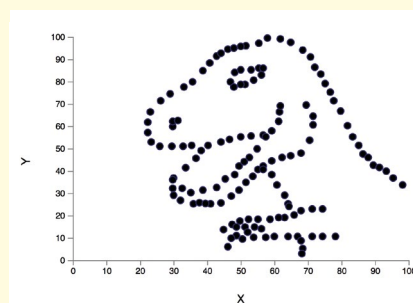
As well as hiding the details of a data set, just using statistical measures can also hide the differences between different data sets, a fact which has been well known to teachers of statistics for a while. In 1973 the statistician Francis Anscombe created four sets of data that clearly appear different when plotted on scatter diagrams, but share the same set of statistical measures, including the means and standard deviations of both x and y .

The first two graphs above show common shapes on scatter plots. The straight line shows a linear relationship between two variables: as you increase

one, the other increases by a proportional amount. The second shows a slightly more complicated but still common relationship: a quadratic relationship. Although these graphs are clearly distinct, they share the same mean values and standard deviations.

The other two graphs show how much an outlier can affect your interpretation of the data. If you just use the statistical measures, these sets of data would look the same as each other, as well as the previous two.

In fact, with some careful manipulation you can create a whole set of graphs that



■ The Datasaurus created by Albert Cairo



MICHAEL CONTERIO

A former physicist, Michael now works as an online course production manager for the Raspberry Pi Foundation.

share the same statistics, but look very different to each other — even as far as giving a picture of a dinosaur with the same statistics as a set of rings (see image and helloworld.cc/datasaurus).

You could use these data sets to really emphasise this to your students. You might present the statistical values and ask students to draw what they think the graphs would look like, before revealing the range of possible graphs. Or you might give each student a data set and the corresponding graph, asking them to calculate the x - or y -mean, before bringing the class back together to share the results that they've all calculated the same values. Either way, this could help start a discussion about why graphs are useful.

Summarising data

I hope I've convinced you about the importance of encouraging your students to use graphs to look beyond the mean and think more about the shape of data. I suggest that you make the time to plot and look at graphs, and ask your students to describe in words what the graphs they've plotted tell us. And maybe then you can link it all back to the statistical measures we've been avoiding so far — and if they can link their numerical calculation with the graphs, they might gain a better grasp of just what the mean means. (HW)

(HW)

SUBSCRIBE TODAY

**FREE
IN PRINT**
for UK-based
educators



Why subscribe?

- Teaching resources and ideas used by over 90 percent of our readers
- Exclusive news, research findings, and in-depth features
- Delivered to your door three times a year



TO SUBSCRIBE VISIT:
helloworld.cc/subscribe

Not a UK-based educator?

- Subscribe to receive the free PDF on the day it is released
- Read features and news at helloworld.cc



FREE PDF
 for anyone,
 anywhere

MOVING TO CLOUD COMPUTING DURING THE PANDEMIC

With learners being remote, and no chance of hands-on physical computing happening, **Steve Rich** turned to the cloud to generate practical evidence for his 16+ students

Where I teach, practical, hands-on assessment has been crucial for producing the evidence that our learners need for their BTEC Computing qualification, especially for the 'Networked systems security' and 'Computer networking' units. Clearly, with learners at home and our ability to support them remotely limited, we needed another option.

For us, that option was Amazon Web Services (AWS). This has allowed our learners to carry out their practical computing tasks in the cloud, completely free of charge. They connected to their self-configured virtual machines (VMs), containers, and databases via SSH, and of course, they could do this from anywhere they had an internet connection.

The free-tier minefield

Pre-pandemic, I'd experimented with Microsoft Azure, Google Cloud, and AWS to host and run VMs and databases, so I knew that this could be an option once I'd negotiated the difficulties of registering to use their free-tier services.

For security reasons, you almost always need a credit card to sign up, even for the free tier. As I did this, it made me a little nervous — what happens if I accidentally generate a huge bill because of an infinite loop? As I did more investigating, I found the Microsoft Azure student offering — it didn't require a credit card and was suitable for UK learners aged 16+. Then I read about AWS Educate, which specifically caters for learners aged 14+ and has plenty of free credits (US\$) available.

Once registered as a teacher, AWS Educate allows you to create a classroom and invite your learners to join and receive their own free credits up to \$100. This was my next worry. Would that be enough for the learners to complete all their practicals?

I didn't have to worry. In three months, my learners only used \$20 of their credit. The free-tier credits are more than enough if you are careful to close down any instances when you aren't using them. Even when I mistakenly left a VM running for a few weeks, it only burned through an additional \$10 of credit.

Through the Educate classroom, you can monitor student activity and assist them when needed — for example, if a student has a configuration issue, you can log in and see their view of the AWS dashboard and make changes or suggestions. You can also assign templates as a starting point, and learners can also earn AWS digital badges — Cloud Explorer, Cloud Inventor, and Cloud Builder, which are fantastic experiences for them and great for their CVs.

Our experience with AWS Educate

From our point of view, learners were able to set up their required practical sessions in the cloud, install and configure software, then tweak and test security settings. Overall, nearly 100 Year 12 and 13 students successfully used it for their practical computing tasks. As long as they had a device to access the internet, they could complete the required tasks. There was little strain on the device they were using to access it, as it was all happening in the cloud, so that helped with equipment accessibility. There was also no chance of the cloud being lost or misplaced, so the learners could always access what they needed, when they needed it.

The pandemic forced my college to investigate cloud learning options much more quickly than we otherwise would have done. As we now return to a more traditional classroom setting, we'll inevitably revert to physical computing for many of our activities,

but I will certainly continue to incorporate some elements of cloud computing. As enterprise IT continues to migrate nearly all services to the cloud, it makes sense to equip our learners with the right practical skills needed to flourish in the new normal. **(HW)**

WHAT CAN YOU DO?

Below are some suggestions of the educational practical activities and experiences that you can run in the cloud for free:

- Set up and host your own dynamic website
- Run your own VPN server
- Practise your command line skills
- Host a web app
- Access remote desktop Linux applications
- Host an SQL (or NoSQL) database
- Machine learning



STEVE RICH

Steve is the lead computer science practitioner and a BTEC Computing teacher at Ada, the National College for Digital Skills, in London (@adacollege). He is also an NCCE facilitator, an Isaac Computer Science facilitator, and a member of the Teach Computing A level subject practitioner panel (@sjrADA).

A WHOLE-SCHOOL COMPUTATIONAL THINKING INITIATIVE

An international school in Scotland has launched a computational thinking podcast to share ideas on how to integrate CT across the curriculum

What does computational thinking (CT) mean to me as a teacher? Is it only related to technology and computer programming? Or is it something bigger that extends beyond computers and has a place in any subject and at any grade level? If so, what would it look and feel like as a problem-solving strategy for students? What can I take from CT that I can transfer to my own content and context?

These are some of the questions that a small group of teachers at International School Aberdeen (ISA), including myself, has been trying to answer for a couple of years now.

Morio Kajiwaru, assistant principal in elementary at ISA, who is also a technologist, guided a group of teachers from across grade levels through the International Society for Technology in Education (ISTE) course on CT (helloworld.cc/ISTEcourse). This enthusiastic and curious group ranged from teachers of kindergarten to grade 12, so there was potential for immense diversity when using CT in their classrooms.

STARTING POINTS FOR CT

If you'd like to use CT in your subject area, start by identifying these four stages in your context and design pedagogy around them:

- **Decomposition:** breaking up a big problem into simpler, more manageable units
- **Pattern recognition:** identifying similar patterns so predictions can be made
- **Abstraction:** separating important core information from other extraneous details
- **Algorithms:** creating a step-by-step recipe for the solution

They then began to find ways of integrating CT into their lessons. I joined ISA in August 2020 and became part of this group. We have continued collaborating in different ways, such as visiting each other's lessons, sharing CT-integrated activities, and examining the student work that is being produced as a result of this approach.

Converting challenges to opportunities in a pandemic

With the pandemic a constant challenge, we continued thinking of ways in which we could collaborate despite staggered schedules and social distancing. And so CT Pod was born; inspired by other educational podcasts, we decided to launch a monthly series starting in January 2021.

CT Pod's main aim is to engage more teachers, from within ISA and beyond, to think about how CT's pillars — decomposition, pattern recognition, abstraction, and algorithmic design — can be embedded in all subject areas.

The teachers who are part of ISA's core CT team have found different ways of doing this. Isabelle Wolfe uses it in her French classes to break down language structure and sequencing. Joanne Curnock uses it in her science classes to organise lab planning by identifying various aspects of an experiment such as aim, materials, safety, testing, and analysis of the results. Elementary classroom teachers Lulu Davies (grade four) and Susan McWhirr (kindergarten) help students understand underlying concepts of different topics by using CT's approach of breaking down bigger problems into smaller, more manageable chunks. These include specific instructions on how to sort out winter clothes; break down a maths problem; or map routes from the stories they read.



SHASHI KRISHNA

Shashi is a middle and high school computer science teacher at the International School Aberdeen, Scotland (@skrish2017).

CT Pod is our way of sharing a friendly and reflective community conversation about these approaches that anyone can engage with.

Looking ahead

In addition to increased teacher engagement, another major focus of the podcast in future episodes is going to be student engagement. Students in these classes and at these grade levels are already familiar with CT and have used it to break down and chunk information in different ways. We are now working on getting them engaged with the podcast series. This will involve featuring their feedback on this pedagogical approach with questions such as: Is the CT approach helping them to break down problems more confidently? Can they identify patterns that they can pursue to start building a step-by-step solution? To help us with the next stage of identifying CT's strengths and features at different age and grade levels, we plan to create age-appropriate spaces where student voices can be represented.

All the episodes can be found at helloworld.cc/CTpod. 

IMPROVING ACTIVE PARTICIPATION IN THE VIRTUAL CLASSROOM

Actively engaging students in online classes has been challenging. Primary computing teacher **Sam Shallcross** discusses using Pear Deck, an interactive presentation tool

How great is it to be back in the classroom? Although I am so proud of what teachers around the world have achieved in delivering online learning from home, it was not without its difficulties! Apart from the obvious challenges of live teaching to a sea of tiny video images of students, like every teacher, I missed the spontaneous interactions and instinctive changes of direction that my lessons might normally take in front of a live audience.

I have always believed that computing is a journey of discovery, awe, and wonder for primary-aged students as they learn more about using technology for learning,

innovation, and exploration. Giving pupils the opportunity to work through their thought processes to problem-solve, suggest their own ideas, share with others, and evaluate are some of the most important tools to a computing teacher — often more important than the actual outcome of a coded game in Scratch. In the classroom, face-to-face, this is more straightforward to achieve; live on the screen, it felt so much harder to gauge every pupil's level of understanding and thought processes.

Investigating interactive tools

It was during the second wave of home learning that I really began to investigate how I could improve pupil participation beyond sharing in the chat window of a Google Meet. Of course, we all know that after one pupil has typed in the answer, everyone else can just copy it, so I felt there was a loss of independent thought... and those who typed quickest got there first!

I first tried using breakout rooms in Google Meet, and it was helpful to be able to have conversations with smaller groups of students. The older primary pupils enjoyed the independence and responsibility of running their own group while I was working with a different group. But it still felt like the strongest or most

vocal students in the group overpowered the quieter students, and I was still struggling to hear all the voices in written comments. Splitting the class up and not being present in each space to listen and guide was difficult with lower-primary pupils, so I still felt this was not a tool that was engaging this age group.

Another tool I explored using was Jamboard, as this online whiteboard works well with Google Meet and is now integrated into the platform. While I was able to control contributions by changing the sharing permissions on the Jamboard, it is not possible to prevent students from deleting content while in edit mode. There are some excellent ways to use Jamboard for whole-class or group interactivity, but I was still looking for something that would engage all participants privately to support pupil voice for all.

Discovering Pear Deck

Then I stumbled across Pear Deck. I had heard of it before, but had never had the opportunity to try it out, and now seemed as good a time as any. So I signed up to a training webinar one evening and found exactly what I had been looking for.

I use Google Slides to create most of my lesson content, sometimes sharing via Google Classroom so that the students can

TRY IT FOR YOURSELF!

The basic use of Pear Deck is free, and Pear Deck are also offering their premium features to educators on a 30-day free trial (peardeck.com/pricing).

For further information on using Pear Deck, I recommend a training webinar for beginners, run most Thursday evenings. Register at peardeck.com/learn-pear-deck.



©MIA Studio/stock.adobe.com

edit their own slides or follow along what I am sharing on the screen.

The Pear Deck add-on instantly made my online lessons more inclusive and interactive. Suddenly, I was able to apply a drawing feature to a slide that invited a student to annotate with a text tool or draw with coloured markers over the slide. But what was truly amazing was that I was able to see all the students' contributions to the slide in real time, while they were only looking at their own! I quickly picked up on the text box feature, which allowed all the class to add their thoughts to a question I asked, and also the polling slide feature, so

spot misconceptions quickly and discuss verbally with students through the Google Meet, or type a private comment in reply to the students' annotations. The discussions we had became more relevant to their own thoughts and understanding.

Bringing virtual tools into the physical classroom

I have only just started on my journey with Pear Deck, but now UK schools are back in the classroom, I have been able to use it a few times with some of my computing classes. Even face to face, with students working on their individual devices, it has

contribute to whole-class discussions. I feel I have a better insight into the understanding of the quieter members of the class and can challenge the thinking of the more confident.

By using Pear Deck, I have found a way to motivate, engage, assess, and connect with my students. When the power of learning is in their hands, the sky's the limit! **(HW)**

“ I HAVE ALWAYS BELIEVED COMPUTING IS A JOURNEY OF DISCOVERY, AWE, AND WONDER FOR PRIMARY-AGED STUDENTS

I could gauge their understanding before moving on. But these examples are just the tip of the iceberg! There are a whole host of features and ready-made content to help build a more inclusive lesson.

Using Pear Deck gave every student in my class the opportunity to share their thoughts and ideas independently, knowing that only I could see them. I was able to

been useful in increasing engagement from all students, by allowing them to contribute to certain slides to show their understanding, rather than offering a hands-up answer.

In particular, while students are typing or annotating their ideas, it has given me more time to be able to work one to one verbally with students who find typing challenging but would also often be reluctant to



SAM SHALLCROSS

Sam is a computing teacher at Bromley High Junior School, a Girls' Day School Trust school in the UK. She is also a Computing at School hub leader for Bromley and Hayes Primary Community (@bromleyJunComp).

INTRODUCING PHYSICAL COMPUTING REMOTELY USING VEXCODE VR

How can we develop physical computing outside of the classroom?

Graeme George reflects on his school's experience and shares options for remote exploration of the subject

Since the introduction of the GCSE computer science course in the UK, our school has taught computer science at both Key Stages 3 and 4 (ages 11–16).

At that time, the computing curriculum was only just being developed for lower secondary, and so we based our Key Stage 3 curriculum on the Open University's *My Digital Life* course. This was the first time I saw the benefit of using physical computing to teach computational thinking. The SenseBoard, developed by the Open University, allowed students to see that there were other input and output devices beyond just the keyboard and the monitor. We were able to create projects such as a voice game controller and a noisy classroom indicator, and even allow the students to send a word between two computers using infrared light.

The department has developed an ethos of giving students opportunities to use computational thinking and coding in as many different situations as possible, allowing them to find the area of computer science that interests them. As part of this, we had started looking at developing a robotics curriculum.

Challenges in the first national lockdown

When schools moved to remote learning, we needed to find a way to offer students an experience that was as near to that of the classroom as possible. Using our normal resources would not be an option, because we knew that not all of the students could run the Sense programming software on

their own computers. We also knew that we would not have sufficient resource to supply all students with a SenseBoard.

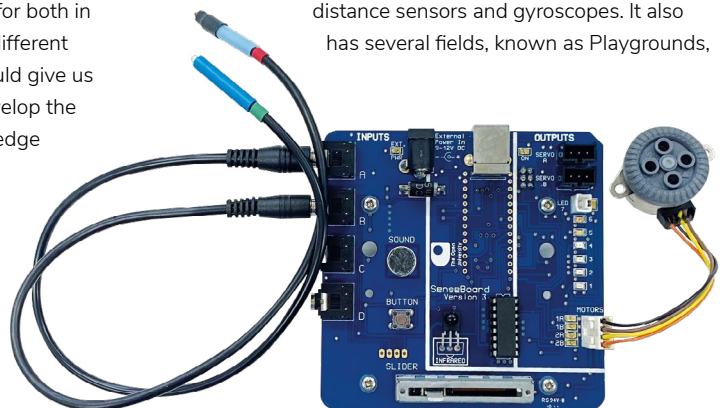
Therefore, we decided to find an online solution that would run on a reasonable broadband bandwidth and had a similar coding environment to the Scratch-based Sense software. For the first few weeks, we didn't teach live lessons, so the choice of coding environment was very important. Students needed to feel comfortable with the software, as they could not get spontaneous feedback to a question or reassurance from the teacher as they could in the classroom.

The other challenge was having an environment that recreated an authentic illusion of physical computing.

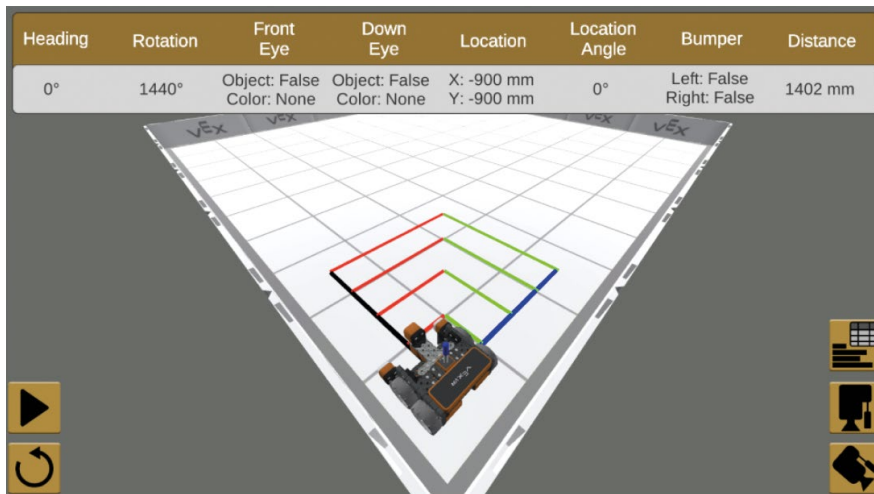
I had previously come across the block-based environments used by Snap! and Robot Mesh, and the department could see a use for both in a classroom, in different ways. Snap! would give us the ability to develop the students' knowledge using its more complex blocks and libraries, but it did not have the ability to use a variety of sensor inputs and outputs, like the SenseBoard.

The Robot Mesh software offered a possible solution to our problem in that it has a realistic display called a Mimic that allows you to run your program on a virtual robot. It also enables you to create your own obstacles and barriers, and can be downloaded onto an actual VEX robot. However, the software required us to create the robots and fields that we needed ourselves. From our experience, we knew that the Mimics needed to be created using a computer with a fast processor speed, which our laptops didn't necessarily have.

At school we have been involved with the VEX Robotics Competition for a few years, and through the VEX community I was introduced to the newly released VEXcode VR. Like Snap!, the software uses a blocks environment, but unlike Robot Mesh, it has precreated robots with a wide range of sensors from push buttons to distance sensors and gyroscopes. It also has several fields, known as Playgrounds,



■ The SenseBoard



■ Example of the live dashboard on VEXcode VR running a piece of code

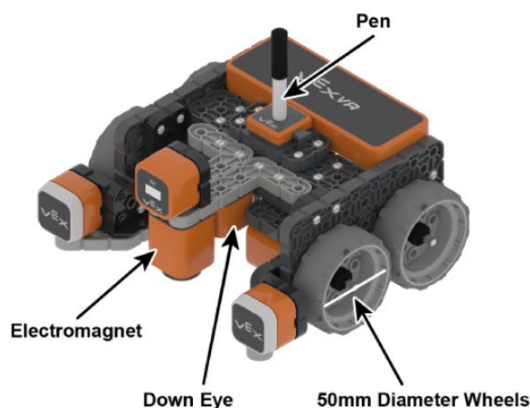
“ WE NEEDED TO FIND AN ENVIRONMENT THAT RECREATED AN AUTHENTIC ILLUSION OF PHYSICAL COMPUTING

with obstacles already created. VEX had also created a number of activity worksheets with differentiated activities linked to the Playgrounds.

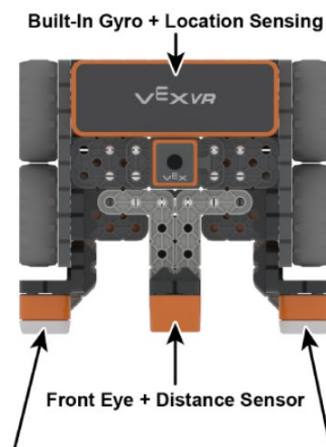
How we used VEXcode VR in lockdown

From our pre-existing worksheets and the activities created by VEX, we created a mini robotics course taking the students through the basics of sequencing and selection. The number of Playgrounds in VEXcode VR allowed us to create different exercises for each topic area and use them to scaffold the learning for the students. Additionally, we

developed a website containing videos to introduce the activities and give students support in completing the tasks. The students immediately engaged with the new environment and liked the way that it allowed them to see the solution working in a physical environment. They also liked the way that it gave realism to the problem — environment conditions meant the robot sometimes didn't perform exactly the same all the time, just in the same way as a real robot. As students became more motivated by the environment, we were able to send them the more complex activities already created by VEX.



■ Premade robot with fitted sensors



GRAEME GEORGE

Graeme is the subject leader for computer science at Simon Langton Grammar School for Boys in Canterbury, Kent, in the UK.

Using VEXcode VR back in the classroom

Throughout the pandemic, VEX have continued to develop the software and increased the number of Playgrounds and associated worksheets. This has meant that we've been able to look at the software as a more long-term part of the curriculum. Since September, we have altered our lower secondary curriculum so that the students are able to learn and develop their computational thinking skills using Snap! and VEXcode VR, choosing the most appropriate environment. We can continue developing the physical computing ethos in the lessons, knowing that students can continue to work on projects at home without needing access to physical hardware. By the time our students reach Year 9, they should have enough familiarity and confidence with the software, and coding skills, to further develop their prowess using the physical VEX IQ robots that we've bought.

The ability to write programs in Python and the complexity of the Playgrounds mean that we can use the software with Key Stages 4 and 5 (ages 14–18). For example, the Playgrounds will allow us to create maze algorithms in Key Stage 5. I can see us being able to extend the use of the VEX IQ robots across all year groups, which is a very good point to be able to make when trying to get funding for physical computing equipment! (HW)

COMPUTING OUTREACH DURING THE PANDEMIC

The University of Plymouth's computing department refactored its outreach activities during Covid-19, learning how to play to the strengths of online delivery

The University of Plymouth's computing department provides outreach activities for students and teachers across South West England. These activities are usually face to face and range from lectures presenting the state of the art in computing, to exposing students to topics in computing with practical lab sessions, to delivering CPD activities for teachers. As the pandemic arrived in early 2020, we were faced with having to figure out how we could refactor our sessions for online delivery.

Outreach before Covid-19

Plymouth first began working with the UK's grassroots Computing at School (CAS) community in 2014, providing training for local Master Teachers at schools, and running our first on-campus event. Between 2015 and 2018 we were a CAS Regional Centre, before becoming a regional delivery partner for England's National Centre for Computing Education. By 2019, we were running a multitrack conference for primary and secondary audiences and featuring workshops aligned with curriculum requirements, including hands-on activities and interesting and relevant presentations and discussions. Eighty-four people attended the 2019 conference.

Alongside our CAS work we participate in university-run events, presenting outreach relating to our specialisms of cyber security and artificial intelligence. Activities have included practical demonstrations of computer forensics tools, and talks on AI and big data.

Outreach during Covid-19

As soon as lockdown started, we knew that our outreach activities should continue being a priority, but we also knew we would have to change our approach. An early effort was to repackage some of the face-to-face talks as webinars. Early talks on AI and big data were successful, and it was clear from these initial activities that we were able to reach a much wider group of participants than before, with students from right across England taking part. Participants did not have to travel to Plymouth to attend, and it was much easier for them to find a spare hour in the day than it was to find a whole day. Schools were also able to use the talks as activities within their teaching sessions.

Running a conference posed a bigger challenge. We extended the event by including a student track in the conference for the first time — one that was designed to put cutting-edge information in front of students with an interest in computing. We also continued to ensure that the tracks for teachers would be relevant. Then, we invited relevant speakers who were comfortable with Zoom, and leveraged our outreach team's experience to set up the web infrastructure. Sessions were kept to 45 minutes, with plenty of breaks throughout the day to reduce Zoom fatigue. Each session had a university student ambassador who monitored the chat and ensured that everything was running smoothly. We received over 600



■ A student exploring a virtual-reality world



■ Students learning about data visualisation

registrations for the event, again with participants joining from as far afield as the north of England.

We also moved our coding challenge online. We had previously run Code Skool days for Year 7s and 8s (ages 11–13), and this time we adjusted the challenge to account for older, more experienced students. Again, running the event virtually made it accessible to a much wider range of students than when we ran it in Plymouth.

What we've learnt

This last year has been a learning experience like no other. A big takeaway for us has been how vital admin support has been. The one time we tried running a webinar without a moderator to manage the chat and admit people was easily the least slick! Having someone to concentrate on these areas, so that the person delivering the session can focus on their material and engage with the students and teachers who have come along, is better for everyone.

Another major challenge is battling Zoom fatigue. It's vital to break up the material so that attendees are guided through a set of activities beyond simply watching and listening — more so than in face-to-face sessions. The conference included a good number of activities, and in the webinars, reducing the material and extending the discussion time afterwards worked well.

Running the conference virtually really highlighted the difficulties of running practical sessions remotely. Despite our having prepared code snippets and provided an online JavaScript editor, the distribution of these did not go according to plan, and it was very clear that this was one activity that would not port well to the remote environment. Further, there was no opportunity for the student ambassadors to help those attending, as they would have done in a normal lab environment. The time slot was also too short for a workshop — with hindsight, scaffolded activities that built up during the day would have been more beneficial in the online environment. We've definitely learnt a lot about how to do this in the future, and will be putting this experience into practice as we design our future outreach activities.

We've also found practice runs are vital, to ensure that the extra infrastructure required for online delivery works properly. We discovered on more than one occasion that the university firewall stood between us and the successful completion of an activity — fortunately prior to the real event. It's really important to check simple things such as whether resources can be accessed remotely, or it can really throw off your session, and these are easy enough to check if you have access to a friendly off-site helper.



SHIRLEY ATKINSON

Shirley is the associate head for computing at the University of Plymouth. Her research field is in privacy, and her lecturing in software engineering builds upon years of industry experience.



DAVID WALKER

David is a lecturer in computer science at the University of Plymouth. Alongside teaching and research into AI and software development, he has for several years designed and delivered outreach for schools.

Outreach in the post-Covid world

As things return to normal, we expect that we will resume face-to-face outreach. With that comes the task of integrating the lessons learnt during the last year. One thing that we intend to keep are the webinars. They offer a much greater geographical reach, and give students the confidence to engage with us in a way that many struggle with when they are in front of other people.

As we carry on delivering webinars, we'll also continue working on new ways of interacting with attendees to further enrich the experience for them. 

Find out more about our outreach opportunities at helloworld.cc/plymouth.



USING JAVASCRIPT TO TEACH PROGRAMMING REMOTELY

When the pandemic sent students home, the challenge for **Laura James** was keeping them coding. JavaScript to the rescue!

When lockdown loomed and we found ourselves facing the prospect of remote learning, my Year 9 computing students (aged 13–14) were about halfway through a program of study learning Visual Basic (VB). Trying to get students to continue learning this language at home presented many difficulties: did the students have Windows computers? Could they download the Visual Studio software? Would students even want to program when, faced with home learning and no teacher guidance, the inevitable bugs appeared in their programs? All of these challenges made me consider whether there was a better language to use to teach programming concepts.

Switching to JavaScript

Before they began learning VB, my students were learning to build web pages using HTML and CSS, to get them used to the rigours of writing accurate text-based code. The students really engaged with HTML and CSS as they learnt techniques to build their own websites full of colour, images,

and embedded content that they could really make their own. This success with web-based HTML, and the virtual teaching barriers to VB, gave me the idea that rather than pursue the dated language of VB, we could attempt to learn JavaScript.

JavaScript has many advantages. It is regularly listed as one of the most popular programming languages in the world, as it is used on almost every website. It also has huge advantages as a teaching language, as it can be run in any browser. And most students have access to a browser. If you right-click in a browser, choose Inspect or Inspect Element, find the Console in the developer tools, and type `alert("hello world")`, then you've programmed your first bit of JavaScript!

Starting with jsbin.com

However, we needed to find a better integrated development environment (IDE) that students new to JavaScript could quickly get to grips with. Enter jsbin.com.

Discovering this simple site was a revelation: it allows you to write mini web pages with attached CSS and JavaScript

that can be saved with a unique URL, without needing to log in. During the lesson, I could give students the steps they needed to perform, either over a live video demonstration or as a prepared PowerPoint or prerecorded video, and students could follow along. If a student was really stuck, they could send me the link to their latest bin and I could help them spot the error.

After a couple of lessons, most students were confidently adding alerts and prompts to HTML buttons they had created on a fictional scam web page, pretending to ask for people's credit card numbers, or showing a pop-up saying "You've been scammed." We then moved on to manipulating the web pages themselves, changing background colours and images, and adding text.

Moving on with glitch.com

Eventually we outgrew using jsbin.com as a development environment, as it only allowed for coding of a single web page at a time, and we went back to using the site we had started out learning HTML and CSS with: glitch.com.

Glitch.com is another free site, and one that allows many more web coding opportunities than jsbin.com. Once signed up, you can create multiple projects, which act like mini websites: you can have multiple pages inside each site, and use shared resources such as CSS and JavaScript files.

Inside glitch.com, students created a range of projects, such as:

- A simple login form to introduce 'if' statements
 - A Love Tester, using random numbers to predict how well two names would be matched
 - A times table generator, using 'for' loops
 - A random task chooser to introduce the concept of arrays
 - A Rock, Paper, Scissors game, using Boolean operators such as AND and OR
- You can see my resources for this scheme of work at javascript.geekcandy.co.uk.

As with jsbin.com, students could send me a live link to their site on glitch.com if they were stuck, and I could help them spot their error.

The pros and cons of JavaScript

While JavaScript has been a good language for my classes during lockdown, it might not work for everyone.

The advantages of learning JavaScript are:

- It's a useful skill for the real world
- It links back to HTML and CSS, which my students had already studied
- The online IDEs make it accessible to almost all students
- There are plenty of resources online, so students can look up how to do other things such as add timers, sounds, and graphics
- In the CyberFirst Girls competition, a UK

ONLINE JAVASCRIPT EDITORS

jsbin.com

This is a simple online editor that doesn't require students to log in. They can quickly type their HTML, CSS, and JavaScript, and their web page appears on the right-hand side. It's similar to jsfiddle.net, but I like the ability to hide panels when not needed.

glitch.com

Here's another online editor, but one that requires you to register for an account (which is free). It allows multiple pages and asset hosting, and gives memorable unique site names.

“ WAS IT POSSIBLE THERE WAS A BETTER LANGUAGE TO USE? ENTER JAVASCRIPT

competition supporting girls interested in a career in cybersecurity, some of the challenges use JavaScript

The disadvantages of learning JavaScript are:

- It presents a relatively steep syntax learning curve, and the semicolons and curly brackets can be off-putting; training students to read their code properly, though, is a useful skill for whatever language they learn
- The errors are not always obvious; both the sites mentioned had some form of syntax error checking, but it wasn't always foolproof, and students have to be prepared to hunt down the bugs themselves

Next steps

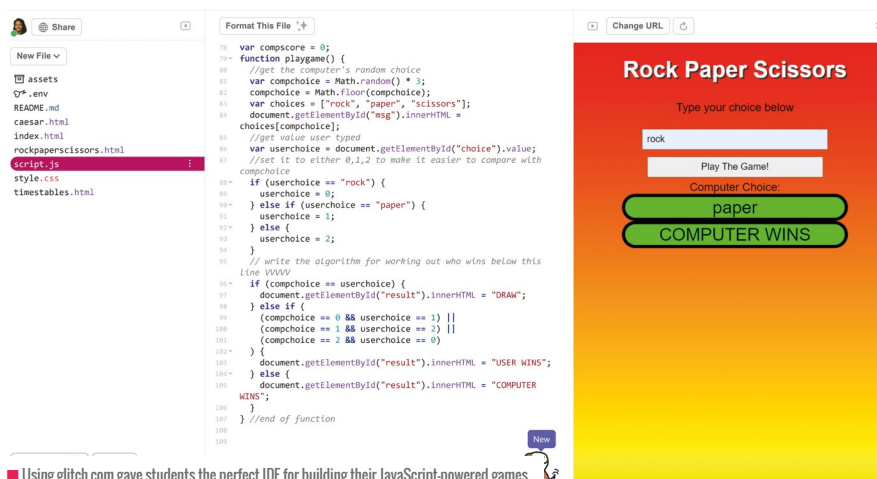
My students have really engaged with learning JavaScript. Many have spent extra time adding features to their JavaScript websites by researching other techniques. Even students who have found the coding part challenging have enjoyed making their websites look

unique by adding different images in the HTML and different styles in the CSS.

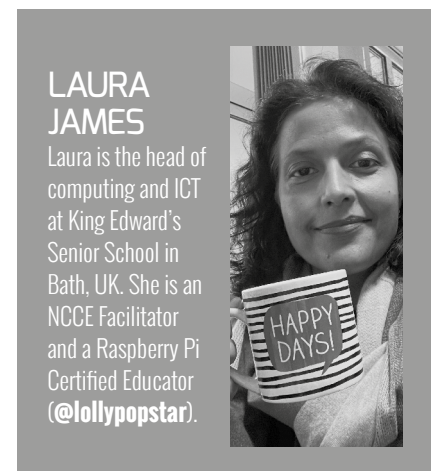
Having learnt the basics of JavaScript, there are many options for my students going forward. JavaScript has several extra libraries that can be imported easily:

- P5.js would allow artistic drawing and animating features
- Phaser.js is a gaming engine with sprites, physics, and game mechanics
- Three.js is a 3D library with lots of examples to ignite students' interest in 3D games and visualisation

Our school curriculum at this learning stage is very flexible, which I appreciate is not the case in all schools. Now we are back in the classroom, I could go back to teaching Visual Basic. However, this experiment has made me realise that online IDEs and a wealth of online resources make JavaScript a viable language to continue teaching in the physical classroom. (HW)



■ Using glitch.com gave students the perfect IDE for building their JavaScript-powered games





JULIA ROEBUCK LEARNTEC

DEVELOPING POLYGLOT PROGRAMMERS

Julia Roebuck discusses the benefits of encouraging students to learn more than one programming language and shares her top tips

In the computing industry, being able to program in more than one language is a valuable skill. Although employers are often looking for experts in a particular programming language, there is also a great advantage in having staff who can work in multiple languages. Furthermore, there is often a shortage in the most popular languages, so being able to swap can help with career progression. If you introduce this idea at school, students are already trained in switching between languages and seeing the similarities, and can ensure their skills never become outdated as technology evolves and new programming languages appear.

Q & A

WHICH OTHER LANGUAGE SHOULD I USE?

If you know another language well, use that. Otherwise, I would suggest JavaScript, Java, or C#. All of these languages have many syntax similarities, and C# is popular among teenagers because of its use in Unity game programming. W3Schools.com is a great resource and has tutorials in Python, JavaScript, Java, C++, and C#.

I DON'T KNOW ANOTHER LANGUAGE — WHAT SHOULD I DO?

You're not going to be able to go into too much detail, but you can point out that the basic constructs they are learning in Python are virtually the same in other commonly taught languages. If you have a friend or colleague who does know another language, you could ask them to translate some code for you.

Advantages in school

In school, you might think, why bother? Students already have enough to learn and some students may get confused. However, the aim is not always to teach another language per se, but to show pupils how similar certain aspects of high-level programming languages are, and plant the seed that they are capable of learning and transferring their skills from one language to another.

It also encourages students to teach themselves if they are interested in self-study, so they are not limited to the language set by the curriculum. It's also likely to keep students engaged and motivated to learn if they can transfer their knowledge to a language that is more suited to their interests.

How to introduce multiple languages

Between the ages of 11 and 16, we are just teaching the basic constructs of all high-level programming. Of all the commonly studied programming languages at school, all of these base features are the same, so this is the ideal time to start mentioning other languages — even if you don't use them in class.

As Python is the most popular teaching language, the suggestions that follow assume that is what you are teaching, but they will still be relevant if this is not the case.

Lower secondary (ages 11–14)

Before you first introduce a text-based language, find out if any pupils are already using a different programming language. You can then speak to them on an individual basis

LANGUAGE SIMILARITIES AND DIFFERENCES

SIMILARITIES A few examples

	Python	Java	C#
Print	<code>print("Hello World")</code>	<code>System.out.println("Hello World");</code>	<code>Console.WriteLine("Hello World");</code>
Setting variables	<code>variable = <value></code>	<code>variable = <value>;</code>	<code>variable = <value>;</code>
Loops	<code>while, for</code>	<code>while, for</code>	<code>while, for</code>

DIFFERENCES A few examples

	Python	Other languages (examples)
Instruction end	New line	Semicolon
Variable declaration	Data type not stated	Data type usually stated
do...while loop	Not available in Python	Available in Java and C#
Defining blocks of code	White space/tabs	Curly brackets

about how much they've done, and use this information when you teach concepts in Python. They should be familiar with the same concepts in the language they've been using. You may notice that those who've done programming in a different language will pick up Python quickly.

From the moment you first teach Python, reinforce that all programming languages do exactly the same thing, just with different syntax and sometimes slightly different key words: all high-level languages have variables, selection, loops, and data structures.

If pupils have already learnt block coding, show them how everything they've learnt in that (usually in Scratch) can also be done in a text-based language. England's freely accessible National Centre for Computing Education resources for Year 8 (ages 12–13) take this approach.

As you introduce each new programming concept, keep reinforcing that other languages all do the same thing, and that although this is the syntax for Python, other languages will often look similar. Conversely, pointing out differences in high-level languages can also be helpful. See the boxout for some examples of these similarities and differences.

If you are using Predict–Run–Investigate–Modify–Make (PRIMM) pedagogy, pupils will get used to examining code and predicting what it does. If you feel they can cope with it, show them the same code they are working on in a different language. This can be done on a continual or occasional basis, depending on how you feel the cohort can manage.

Upper secondary (ages 14–16)

With a set syllabus for this age group, you may not have the time to think about introducing an alternative language, but you can still reinforce whenever possible that what they are programming can be done in an almost identical way in other languages. This is the ideal

time to get advanced students to try a second language, especially if they have completed class tasks.

Advanced secondary (ages 16–18)

The fact they are studying computer science at this stage means many students at this learning level are often already programming in multiple languages.

Encourage students to try different languages in their own time, especially programming something that they are interested in — often this is gaming or apps. Good areas to try could be Swift (Apple apps), Kotlin (Android apps), and Lua (gaming).

Most A level textbooks have multiple language solutions, so encourage students to look at the solutions in the other languages, even if it's a brief read.

If you have already taught Python, you could use a spare lesson to show some code that your learners have already done in a different language and get them to try a few simple tasks. A browser-based editor is best, to avoid the issue of having to download integrated development environment (IDE) software. There are plenty of websites that let you try out different languages.

You can encourage students to learn multiple languages early in their learning and continue through all levels of study. It will help demystify other languages and build skills they can use in their future careers. **(HW)**

JULIA ROEBUCK

Julia has over 20 years' experience programming in multiple languages and technologies. She teaches lower secondary, IGCSE, and A level computer science via virtual classes with her company, LearnTec. She also teaches lower secondary and IGCSE at St John's School in Devon, UK.

ABSTRACTION: THE IMPORTANT BITS

Matthew Parry discusses the concept of abstraction and shares ideas on how to teach abstraction using the PRIMM approach and My Blocks in Scratch

The Collins English Dictionary defines abstraction as “the process of formulating generalised ideas or concepts by extracting common qualities from specific examples” or “something which exists as a general idea rather than as an actual example”. In other words, it is about removing complexity in order to increase understanding, without losing the core message.

We encounter abstractions around us every day. They can be shortened versions of something, to give an example of the whole thing. For example, the British television programme *Match of the Day* only shows highlights of football matches; music websites allow you to stream excerpts of upcoming albums; and the blurb on the back of a book gives only an indication of the whole story.

Abstractions can also be symbols that are used to convey meaning while reducing complexity. Examples are road signs, the London Tube map, class timetables, and infographics.

We also regularly entertain abstract ideas in our day-to-day living. For example, most of us have only a notional understanding of most of the machines we use. We use the terms ‘car’, ‘microwave’, and ‘mobile phone’ without really knowing what they do, and without having any concept of how they work.

Word problems in mathematics are a great example of abstraction. Pupils need to choose (abstract) the relevant information from the description in order to create the number sentence that solves the problem.

Abstraction in computer science

Within computer science, abstraction is one of the key elements of computational thinking. Barefoot Computing states that

“abstraction is about simplifying things — identifying what’s important without worrying too much about detail,” whereas educational researcher Keith Turvey says that “abstraction is a process by which any unnecessary detail is omitted in order to help us to solve a problem or achieve a specific outcome”.

One of the aims of England’s National Curriculum for Computing is to ensure that all pupils “can understand and apply the fundamental principles and concepts of computer science, including abstraction, logic, algorithms, and data representation”.

Abstraction allows us, as computer scientists, to concentrate on the important bits of developing a solution by ignoring the irrelevant detail — much in the same way as pupils do for maths word problems.

For example, we may have been asked to create an algorithm for making a cup of tea. In our algorithm, we can ignore some of the complexity that this task would entail, such as how to use a kettle, and use (abstract)

instructions such as “switch the kettle on” in order to make our algorithm simpler and easier to understand.

Abstraction in Scratch

The majority of programming languages are abstractions. They hide the complexity and, for us humans, the incomprehension of the binary instructions that computers use. A lot of the keywords and commands within programming languages, for example the print and input commands, also abstract more complex tasks.

Within Scratch, there are a number of blocks that abstract the processing that is being performed under the hood. Figures 1 and 2 below show two such blocks, and you can see the steps that are hidden from the user in their captions.

Abstraction lesson idea using Scratch

Using decomposition and the PRIMM (Predict–Run–Investigate–Modify–Make) pedagogical approach to structuring

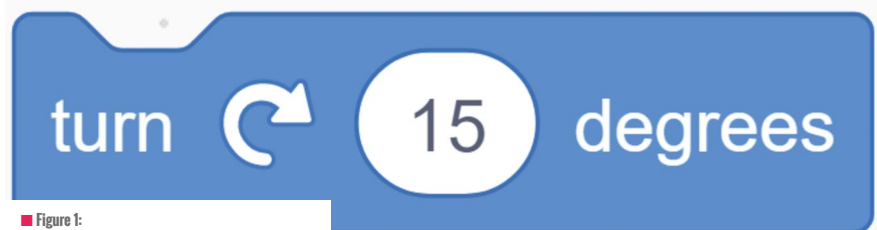


Figure 1:
1st step: Get current direction of the sprite
2nd step: Add 15 to the current direction
3rd step: Set direction to the new value

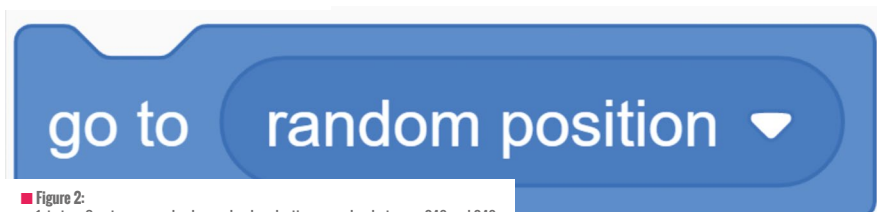


Figure 2:
1st step: Create new x value by randomly selecting a number between -240 and 240
2nd step: Create new y value by randomly selecting a number between -180 and 180
3rd step: Move to the new x and y position

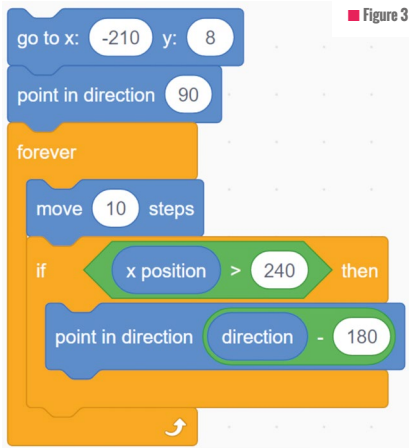


Figure 3

programming lessons (see the boxout for more details), we can unpick another of the Scratch blocks to begin to understand how abstraction works.

Take your class through the following steps, referencing Figure 3:

Predict: Ask your students to look at the

Modify: Ask students to alter the code to make the sprite bounce on the left-hand edge of the screen too.

Make: Get students to add code to make the sprite bounce at the top and bottom of the screen too. Ask them what they will have to change to test this code. For an added challenge, ask students to try and replicate the 'If on edge, bounce' block.

Abstraction using My Blocks

Using My Blocks in Scratch is an excellent way to develop understanding of abstraction. In creating an abstraction, pupils need to fully understand what information needs to be hidden from the user.

One of the complexities of teaching Scratch to younger children is the need for them to understand coordinates and the use of negative numbers to control the movement of a sprite around the stage. We can use abstraction to hide those

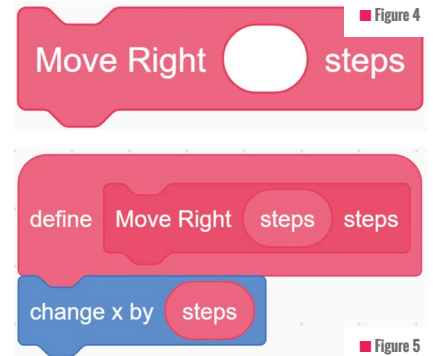


Figure 4

Figure 5

Modify: Can you create other blocks for moving left, up, and down?

Make: Can you create another block called 'Stay on screen', to stop the body of the sprite going off the screen? Can you then amend the 'Move right', 'Left', 'Up' and 'Down' blocks to include this 'Stay on screen' block?

Jeanette Wing, who is recognised as having defined computational thinking, refers to abstraction as the most important thought process in computer science. Abstractions are everywhere in computing: they hide the complexity of the underlying processes, whether that complexity is in an algorithm, a program, the computer itself, or your own understanding of what the internet is. As with all tricky concepts, the more practice and examples you engage with, the more concrete the notion becomes — even with something as abstract as abstraction. (HW)

“ ABSTRACTION ALLOWS US TO CONCENTRATE ON THE IMPORTANT BITS AND IGNORE THE IRRELEVANT DETAIL

code and ask them, “What does this code snippet do? What will happen to the sprite?”

Run: Get your students to run the program and ask them, “Does the code do what you thought it would? If not, can you work out why?”

Investigate: Ask your students to label and comment on the code. You can ask questions like, “What does each block do? What do the numbers represent? What happens if you change the numbers?”

complexities by creating new blocks to move left, right, up, and down. Take students through the following steps:

Predict: What do you think the block in Figure 4 will do?

Run: Using the 'When right arrow key is pressed' block, run the 'Move right x steps block'. Does it do what you thought it would?

Investigate: Investigate the block definition in Figure 5. How does it work? Will it work for all directions?

THE PRIMM APPROACH

PRIMM is a pedagogical approach to structuring programming lessons. It follows the below structure:

Predict: Discuss the program and predict what it might do: what will be the output?

Run: Run the program to test the predictions.

Investigate: Explore the structure of the code, using tracing, explaining, annotating, debugging, etc.

Modify: Edit the program to change its functionality via a sequence of ever more challenging exercises to gain confidence by extending the function of the code.

Make: Design a new program that uses the same structures but solves a new problem.

You can read more at: helloworld.cc/primm.



MATTHEW PARRY

Matthew is a senior lecturer in initial teacher training at the University of Derby. He is also a CAS Community Leader, a Raspberry Pi Certified Educator, a Barefoot Ambassador, and an NCCE course facilitator (@Matthew_Parry_).

PROGRAMMING AND COMPUTATIONAL THINKING THROUGH WEB DEVELOPMENT

Martin Hill shares how he has used web page creation as an engaging complement to block-based programming and computational thinking

Programming may be at the core of computer science, but web development offers an enjoyable complement, with overlapping content and the opportunity to build on ideas already encountered. Admittedly, HTML coding is not the 'real thing'; it's just a markup language, and not one of the imperative programming languages on which we generally focus our teaching. However, in creating web pages, children can apply and practise skills that aid their understanding of imperative programming

USEFUL RESOURCES

- helloworld.cc/weblab: Online HTML and CSS editing for beginners
- helloworld.cc/notepad: Free text editor for PC
- atom.io: Free text editor for Mac or PC
- w3schools.com: A fantastic resource for beginners, with interactive lessons on HTML, CSS, and JavaScript
- balsamiq.com: Wireframing software, free for schools through their Software Donation Program
- lucidchart.com: Online diagramming tool
- EraseAllKittens.com: Entertaining HTML, CSS, and JavaScript game for children aged 8+

constructs. It's also a great way to bridge the gap between block-based coding and text-based coding. More to the point, perhaps, web page creation is relevant and rewarding, and it encourages pupils to make the shift from being passive web consumers to participative and engaged web content producers.

I recently produced a web development unit for my students aged 9–10 and surveyed three classes upon completion of the topic, asking them to compare it to Scratch programming. I found that, despite my concerns that dealing with HTML and CSS syntax for the first time might be a disheartening step too far for them, most pupils considered web development to provide a very similar level of challenge to Scratch. A slight majority also expressed a preference for HTML coding, describing it as new, exciting, and fun. (I suspect that, for many children, it also provided a break from years of block-based programming.)

Introducing HTML

Web development is packed with opportunities to revisit, or even introduce, programming concepts and computational thinking skills. For younger pupils, the big change is likely to be the move to text-based coding. When I first introduced classes to HTML tags, I gave them two near-identical segments of HTML code — one was structurally and syntactically correct, while the other was riddled with errors — and let them play spot the difference. This led to a



■ In this unplugged task, pupils sketch out the page as it would be rendered by the browser

discussion of HTML syntax and the need for precision and attention to detail. I then asked these pupils to look at the source code for a simple preprepared web page and try to identify the purpose of the various HTML tags, based on the browser's rendering. This was also an opportune moment to discuss an important computing concept that appears in all sorts of contexts: hierarchy. HTML code has a clear hierarchy of elements, and vocabulary such as 'parent' and 'child' can be established at this point. To emphasise the idea of hierarchy, pupils could create a hierarchical diagram to show the parent and child elements on a web page.

Moving on, I borrowed aspects of the PRIMM approach to teaching programming (Predict–Run–Investigate–Modify–Make — see helloworld.cc/primm). For example,

TRY IT FOR YOURSELF!

Below is a summary of the web development unit discussed in the article.

Age group: 9-10

Timings: 40 minutes per lesson

Topics already covered before starting this unit:

- A short unit of block-based programming: an understanding of sequential execution is helpful
- A short series of lessons about the internet and the WWW, including what the internet is, what the WWW is, hyperlinks/URLs/search

engines and the credibility of websites: an understanding of how web pages are delivered to computers is helpful

Unit breakdown:

- Lesson 1: What is HTML: what are tags
- Lesson 2: Basic structure of an HTML document: how to use Code.org Web Lab
- Lesson 3: Basic text tags (<p>, <h1>, <h2>, etc.)
- Lesson 4: Reinforcement of previous lessons
- Lesson 5: Intro to CSS: changing colours using CSS
- Lesson 6: More CSS practice
- Lesson 7: Adding hyperlinks
- Lesson 8: Adding images

I asked pupils to look at HTML code and predict its appearance when rendered by the browser, followed by a short period of directed investigation (for example, moving lines of code about). The key intentions here were to improve understanding of the tags and reiterate the importance of sequencing code correctly. Other tasks involved debugging incorrect code, with a focus on common syntactical errors. For an unplugged lesson, you could ask pupils to sketch or write out the page as it would appear in the browser.

Making HTML pages

Learners will naturally be itching to start coding a page themselves. Code.org's excellent Web Lab provides a friendly environment suitable for younger pupils, by offering a basic HTML template and an easy file management system, thereby eliminating a number of common pitfalls for beginners. For older or more advanced pupils, free text editors such as Notepad++ and Atom are ideal.

After some initial code-modification tasks, pupils can be tasked with making a whole page. At appropriate points, following the introduction of new tags, you can issue a wireframe design for a page for pupils to follow. Once they're ready to design and make their own page or site, they can work through a decomposition process by planning the organisation of the information on the website. If it has more than one page, they can create a diagram to represent its navigational

structure. Pupils can then identify the elements (headings, paragraphs, images, etc.) and plan the layout for each page by creating wireframe diagrams (beginners should stick to a simple 'one thing under another' layout — putting elements side-by-side is much more difficult) before finally implementing. During implementation, we have the chance to

efficient CSS code. We can even introduce inheritance, as children inherit the styles of their parent elements.

Of course, all the programming concepts I've mentioned, and more, are encountered in their typical context if we go deeper into web page creation and start experimenting with JavaScript. We can also identify a huge variety



MARTIN HILL

Martin has been teaching computer science in state and independent schools in the UK for 20 years. He currently works at an independent school in Edinburgh where he teaches classes from primary 6 to secondary 6 (aged 9-18).

“ WEB PAGE CREATION HELPS PUPILS TO ADAPT FROM BEING PASSIVE CONSUMERS TO ENGAGED CONTENT PRODUCERS

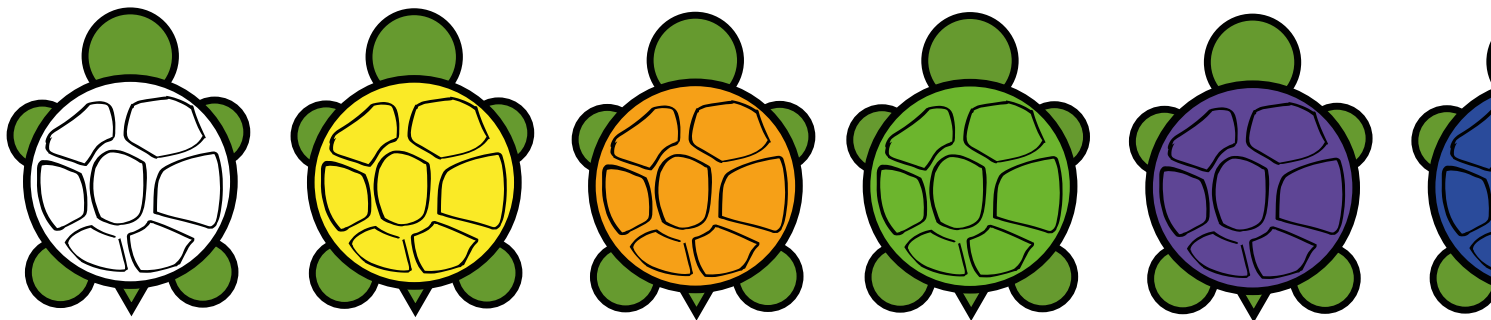
discuss readability and maintainability of code; pupils may have added comments to block-based code before, but they can now also use indentation and white space.

CSS and beyond

Younger pupils love to brighten up their pages using some basic CSS code and, although this involves a whole new language, the syntax is relatively straightforward. Linking HTML code to a separate CSS file relates to the idea of abstraction and, even at its most basic level, CSS involves selection (CSS selectors are used to tell the browser which styling rules should be applied to which elements). By identifying patterns in the styling of the page, such as the same colour of text being used for all headings and subheadings, pupils can write more

of links to other aspects of computing; during the web development topic, I've found myself delivering impromptu lessons on number bases, networks, colour theory, censorship, copyright, databases, agile development, graphic file formats, compression algorithms, and more. As an added bonus, various basic ICT skills such as cut/copy/paste, common keyboard shortcuts, and file management can be developed naturally in this context.

Ultimately, I believe pupils will relish the creative freedom offered by web development and be engaged by its relevance to the world around us. Whether or not they continue to explore this vast topic, they will benefit from the connections they are able to make with other areas of computing and the reinforcement of existing knowledge. (HW)



BLOCKLY TURTLE PLAYGROUNDS

Chris Roffey introduces a new set of free resources for teaching programming at primary and lower secondary level and makes suggestions about how to use them

Students do not all learn in the same way. Even for an individual child, something that was engaging at first can later become boring and frustrating. One of the challenges teachers face is gauging the engagement of their class and swapping activities to ensure that all the children in the class keep making progress. There are many useful resources for teaching programming to primary school students, each providing a different learning experience: floor robots, Scratch, ScratchJr, and Code.org are all very popular. This article introduces another programming resource and provides suggestions of how it may be used.

Programming playgrounds

Blockly Turtle Playgrounds provide free resources to teachers globally for teaching programming concepts. These can be accessed from the ukct.org.uk website. There are eight programming playgrounds, progressing from Playground A to Playground H. Each playground adds a few more carefully chosen blocks to the last, enabling children to learn new programming techniques gradually and hence become more creative.

Playground A has just three blocks. It provides the same commands as those

available on a floor robot, and has its own mat where teachers can set challenges customised to individual children, or use some of the challenge cards provided (introduced later in this article). The idea is that students will meet familiar problems to those they have solved with Bee-Bots (or any of the other available floor robots) in a new environment that is not at all overwhelming. To keep children focused on the task at hand, don't share the index page to the Blockly Turtle Playgrounds with them, but instead give them a link directly to the playground they are going to use.

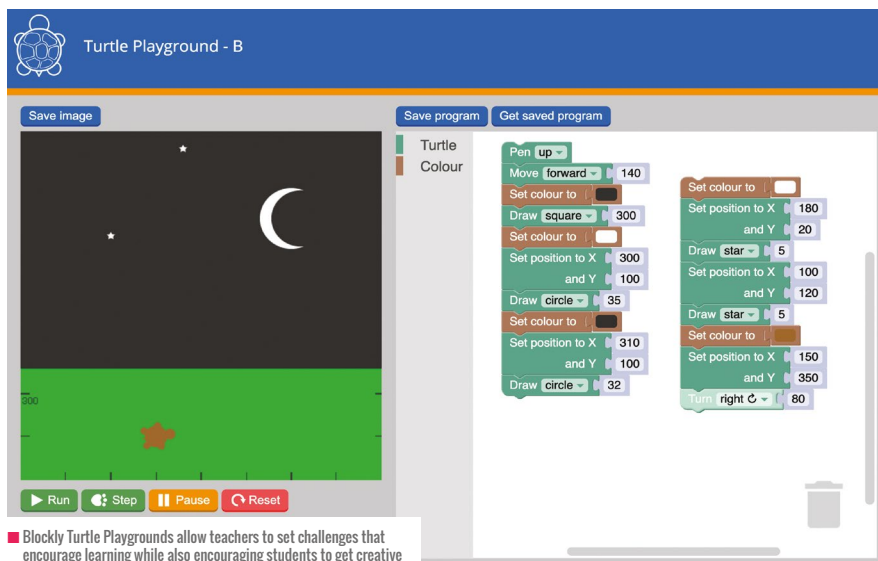
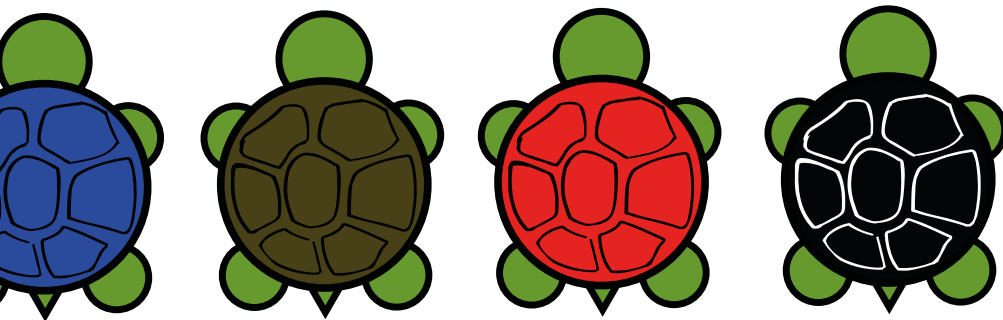
Playground B adds new movement blocks and colour-changing blocks, and replaces the floor robot mat background with a generic drawing background. It provides all the same buttons as those in Playground A and adds the ability to save and upload programs.

Playground C introduces a new loop block and a more technical background and turtle for more precise positioning. Logic blocks are brought into Playground D so that 'if' statements can be introduced. Playground E adds maths and variables, while Playground F allows students to create their own functions. Playground G introduces children to some simple robotics.

Progressing through Playgrounds A to G will allow teachers to cover all the programming concepts required for ages 5–11. Students who want to go further can be directed to Playground H or the Oxford University Turtle System. Playground H provides a sophisticated environment in which turtle commands can be combined with all of the standard Blockly blocks. It allows students to use sequencing, selection, iteration, decomposition, and recursion in a block-based language, where typing and formatting errors are avoided. The Oxford University Turtle System lets students try a text-based implementation of Turtle, where turtle commands can be combined with a selection of programming languages, including Python.

Supporting resources

Supporting resources are gradually being released and are freely available to download and print. Challenge cards provide a range of challenges that teachers can set their students to help them learn about each new tool and the programming techniques possible in each playground. There are also grade cards provided for children to keep track of their own progress. Each completed grade card awards students with a coloured turtle badge — and the pinnacle of



■ Blockly Turtle Playgrounds allow teachers to set challenges that encourage learning while also encouraging students to get creative

“ THIS DISCOVERED LEARNING IS MUCH MORE REWARDING FOR CHILDREN THAN BEING TOLD HOW TO DO EVERYTHING

achievement is becoming a Black Shell Turtle Programmer. Auto-marked quizzes are also currently being developed.

Guided discovery learning

The turtle playgrounds give non-specialist teachers a structure they can rely on and the power to select appropriate learning experiences for their students which has, at its heart, a philosophy of guided discovery. Instead of being overwhelmed with a new system, children will be enthused to find out what they can do in each new playground. For example, in Playground A, the movement blocks provided move 80 pixels forward or back, and turn right 90°. This is so the students can use them to navigate from square to square on the provided mat

without editing the numbers. However, the numbers are editable, and some children will be very pleased with themselves when they discover that they can move from the centre of a square to the grid lines by moving 40 instead of 80 pixels. They may find this out by using maths or through experimentation. These students will be developing a better sense of the linearity of numbers.

I believe this discovered learning is much more rewarding for children than being told how to do everything and, as the students own the learning, it will last longer. Children do not need to understand that moving 80 pixels is moving a fifth of the distance across the turtle space. They can discover for themselves that reducing 80 pixels in the forward block means the turtle won't go



CHRIS ROFFEY

Chris organises the UK Bebras Challenge on behalf of Oxford University and the Raspberry Pi Foundation, and the Oxford University Computing Challenge. He is also the author of several programming books for children (@coding_club).



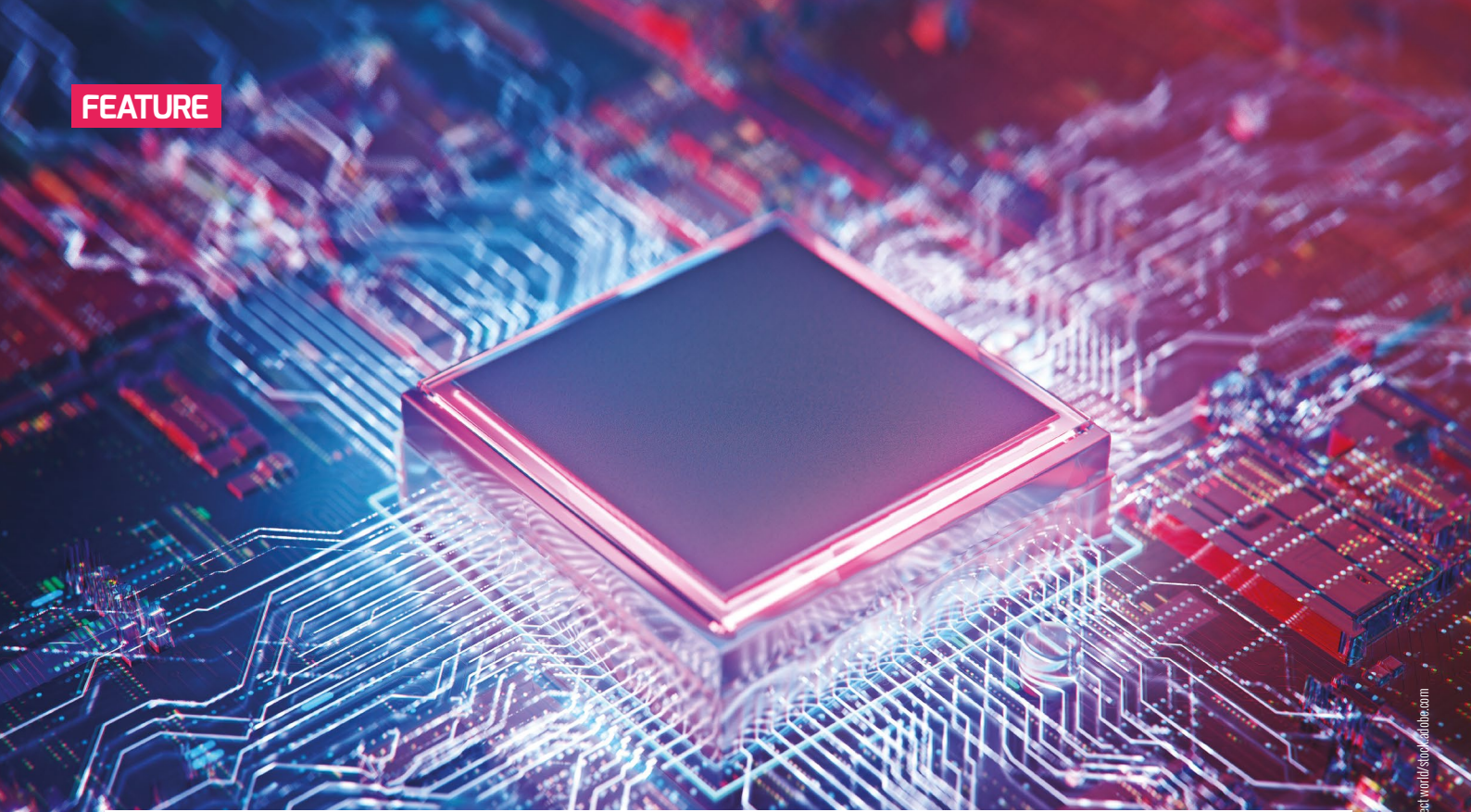
5. Visit the log

■ This challenge card can be given to children working with Turtle Playground A

as far. I am very excited to find out whether teaching computing principles in this way will result in children finding numeracy easier to learn in their maths lessons.

Students learn in different ways. I hope this new resource proves useful for educators who want a system that organises the learning of the key programming principles in primary and lower secondary, and works alongside more familiar resources. It would be great to hear a child who is trying to make a project in Scratch ask: "How do I use functions?" This is the kind of flexible learner I am hoping that the Blockly Turtle Playgrounds can help develop.

The playgrounds and resources can be found at ukct.org.uk. Please email chris@codingclub.co.uk if you would like to give feedback or ask questions. (Hw)



ASSEMBLY LANGUAGE ON THE PI: “LEARNING HOW TO WALK AGAIN”

Simon Humphreys shares how Raspberry Pi's ARM processor makes it a great tool for A level students to get their hands dirty with assembly language

A ssembly language programming is on every university computer science course and also in numerous advanced level courses — so it must be important! The instruction set for the UK's AQA A level is akin to the ARM instruction set used on the Raspberry Pi, making it a great tool for bare-metal programming. In this article, I discuss taking a PRIMM (Predict–Run–Investigate–Modify–Make) approach as one way of introducing the topic, and look at how to encourage students to use a real processor to build programs using ARM assembly.

Why learn assembly language?

Programming in assembly is the closest we ever come to the CPU and its architecture, as each assembly instruction is shorthand for the binary code the CPU executes. There is no better way of understanding the role

of the processor, how it executes programs and manages memory, than by attempting some low-level programming using a given instruction set. It helps us to answer questions such as:

- How is an integer stored in memory?
- How does the computer carry out an 'if-then-else' statement?
- What happens when one function calls another function?
- How does the computer know where to return to after running a function?

There are many brilliant simulators, especially the new ARMLite Assembly Language Simulator by Peter Higginson, that can help students to take what they learn in that sandbox to a real processor (helloworld.cc/ARMLite). I highly recommend using it, along with Richard

Pawson's book *Computer Science from the Metal Up* (helloworld.cc/pawsonbook).

If your school has a set of Raspberry Pis, it's easy to make a start with real-world assembly and give your students insights into how their high-level code is ultimately made to work on a given architecture. Putting into practice some of the fundamental concepts of computer science (for example, binary arithmetic, allocation of memory, working with the stack, character set encoding, handling of interrupts, and so on) should help to make them better high-level programmers.

We'll now look at a simple example, following the PRIMM approach to teaching and learning programming, which can be equally applied to low-level languages as high-level. The following overview should take one lesson to cover, with additional time for extensions.

A SIMPLE EXAMPLE (CONTINUED OVERLEAF)

PREDICT

Ask students if they can work out what the following ARM assembly language program does.

```
1. .data
2. first: .word 5
3. second: .word 8
4.
5. .text
6. .global _start
7.
8. _start:
9.     ldr    r0, =first    @ get address of first value
10.    ldr    r0, [r0]       @ get value at that address in r0
11.    ldr    r1, =second   @ get address of second value
12.    ldr    r1, [r1]       @ get value at that address in r1
13.
14.    cmp    r1, r0         @ compare: is r1 > r2
15.    bgt    _greater       @ if true, branch
16.    b      _exit          @ jump over to exit
17. _greater:
18.    mov    r0, r1         @ true, put return code of 1
19. _exit:
20.    mov    r7, #1
21.    swi    0
```

OK, perhaps it's an unfair question if they've never encountered ARM assembly instructions before, but the comments have been left in place for students, and should provide some clues!

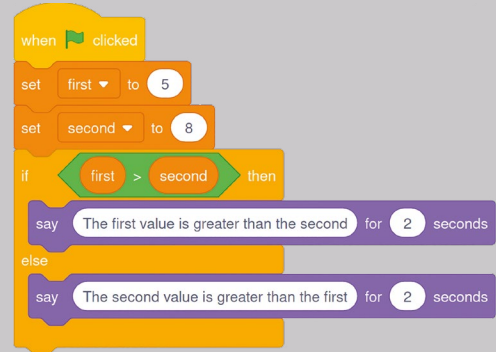
RUN

Ask students to enter the code in an editor of their choice and save it as `max.s`, then in a terminal window, enter the following commands (the dollar is the terminal prompt):

```
$ as max.s -o max.o
$ ld max.o -o max
$ max ; echo $?
```

If all goes well, the value 8 will be displayed (note: the command `echo $?` displays the value in register 0 on the screen).

Does the answer provide any further clue as to the program's purpose? If in doubt, the Scratch example pictured should make it clear.



```
int first = 5;
int second = 13;
```

Lines 5-6

This is the code section. Our instructions start here and the entry point, `_start`, is given by another label.

Lines 8-21

This is the code for our program. Each line has either a label or an instruction, with labels placed on their own line to help with readability. Each instruction is an operation, or opcode (operation code), followed by the operation's operands.

Lines 9-12

These lines fetch the data values from memory. We start by getting the address of our first value and copying it into register 0, then we fetch the item of data at that address and copy it into register 0 (overwriting the address). The next two lines do the same thing for our second value using register 1. Note that the direction of operation, `ldr r0, =first`, will copy the address of our data item into register zero, and not the other way around.

The CPU has a number of registers. Some are general-purpose, and others are set aside as special-purpose, for example, the program counter. The ARM CPU has 16 of these, with `r0` and `r1` being two of the general-purpose registers.

Now we have our two values in two registers, where they can be processed by further instructions.

INVESTIGATE

There may be several activities we can do with our students to dig into the code, for example, reminding them about the role of the CPU in fetching, decoding, and executing instructions. Here, we'll just deconstruct what is happening in the code from the Predict stage, section by section, line by line.

Lines 1-3

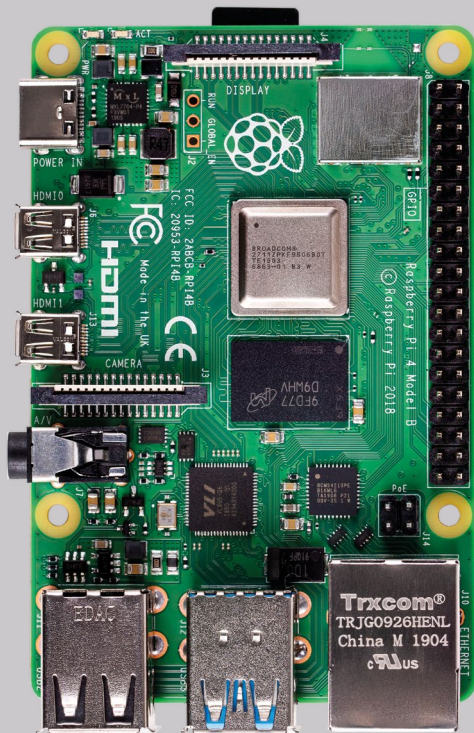
This is the `.data` section, and here we declare our data. Each is given a label, a size, and a value. The first label, `first:`, will allocate 4 bytes of memory, and assigns the value 5 to that address. The label is a symbolic address, meaning it points to a memory address. The second label does the same thing, but with a different value.

This is equivalent in a high-level language to:

A SIMPLE EXAMPLE (CONTINUED)

Line 14

The two values are compared using the `cmp` instruction. Internally the processor will subtract one value from the other and update another register, the flags register, based on the outcome of that operation. For example, if the result is negative, the negative flag will be set to 1. Other flags include the zero flag and the carry and overflow flags.



Line 15

The instruction `bgt` means 'branch if greater than'. As high-level language programmers we've been taught to avoid using the GOTO instruction, but we cannot avoid it in assembly! It's the only way we can force the CPU to jump to a new address and fetch data/instructions from that address. In this case, we jump to the address defined by the label `_greater` if the second value is greater than the first.

Line 16

Line 16 contains another branch instruction. This is unconditional: just jump or GOTO the address specified, `_exit`.

The branch instructions can take a number of conditional commands, e.g. `eq` branch if equal, `ne` branch if not equal, `lt` branch if less than, etc.

The combination of the compare instruction, `cmp`, and a branch, `bxx`, is the equivalent of an 'if

then-else' statement, so either line 16 or line 18 will be executed.

Line 19

We then fall through to the `_exit` section, which ensures we exit gracefully from the program.

Thus, the program is a classic introductory programming problem, returning the larger of two given values.

MODIFY

We can then invite students to carry out a number of potential activities, such as returning the smaller of the two values, or providing three values to compare against each other.

MAKE

Once the students know how to write an equivalent 'if-then-else' construct and how to branch, it becomes relatively trivial to introduce loop constructs such as 'do-while', which have a conditional expression at the bottom of the loop and need to jump back to the top of the loop.



PROGRAMMING IN ASSEMBLY IS THE CLOSEST WE EVER COME TO THE CPU AND ITS ARCHITECTURE

➤ The above exercise does need to come with a health warning! One of my students described programming in assembly as learning how to walk all over again. Assembly does not come with nicely prepared features such as variables, loops, selection constructs, or functions, and even the data types and manipulation of memory are left to us to manage. But stick with it — the pay-off is rewarding.

Where next?

In a short article, all we can do is scratch the surface of such a fascinating area. This example provides many insights into the role of the CPU; how a compiler would take source code and translate it into machine code; how

memory is allocated; and that a variable is just an address of a fixed size. The registers can be inspected using a debugging tool such as GDB (helloworld.cc/gdb), and students can also see the role of the program counter and how it is updated as the program progresses.

Assembly language programming is fascinating, tricky, and irritating in equal measure. For A level, the problems set are usually no more difficult than rewriting a simple high-level algorithm that uses one or more of the constructs such as our example. Taking students further, to build functions or manipulate the stack, provides genuine insights and will make them better high-level language programmers — but perhaps that's for another article! (HW)



SIMON HUMPHREYS

Simon is the vice chair of Computing at School (CAS). He is also a computer science teacher at Hills Road Sixth Form College in Cambridge, UK.

CODING WITH MOONHACK

Moonhack's 2021 coding challenge celebrates Indigenous cultures in Australia and New Zealand

Moonhack is an annual international campaign run by Code Club Australia from 24 to 30 May 2021. It brings together kids from over 70 countries virtually to code planet-saving solutions to sustainability problems across the globe. Over 40,000 kids have joined the Moonhack adventure over the past six years, coding everything from moisture-sensing microcontrollers to animations of forest-regenerating drones. This year, the coding projects celebrated Indigenous Australian and Māori stories; Code Club Aotearoa and Code Club Australia are committed to ensuring that their lessons are welcoming and representative of all peoples. In this article, we'd like to introduce you to two of these projects.

© Tess Reading



■ Coders can animate the journey of the Dhinawan across the sky

'The Dhinawan in the Sky'

Code Club Australia was honoured to work with Gomeroi artist Tess Reading to feature an Australian Indigenous story, 'The Dhinawan in the Sky', in Moonhack this year.

Australia's Indigenous people are the oldest living culture in the world, having existed on this land for 60,000 years or more. Among many firsts, Aboriginal and Torres Strait Island peoples are the world's oldest astronomers, and we were thrilled to acknowledge this as part of our Moonhack celebrations around the world.

The Gomeroi (Gamilaroi/Kamilaroi) people are based in north-west New South Wales. In this Moonhack project, Code Club Australia worked with Indigenous people around Tamworth to tell the story of the Dhinawan, or emu, that appears in the night sky in the southern hemisphere. It tells the story of the cycle of birth of the emus, from the females laying their eggs to the males sitting on the nests.

In this project, coders animate the journey of the emu across the sky, as well as its nest on land, to trace the connection between land and stars; the cycle of celestial and animal movements is heavily

tied into Gomeroi lore. 'The Dhinawan in the Sky' is just one of the many stories told by the Gomeroi people, and is also shared across many language groups. It is used by Gomeroi to mark the cycle of their lives.

Matariki: Māori New Year

As well as a Gomeroi story, Moonhack 2021 featured a Māori story about Matariki, with the help of Code Club Aotearoa, who worked with community members to share this story.

Matariki is a constellation of stars that rises in midwinter in Aotearoa (New Zealand), signifying the start of the Māori New Year. Also known as the Pleiades or Subaru, this constellation has cultural significance across the globe. Even within Aotearoa, the stories and celebrations around Matariki differ between regions and iwi (tribes). Code Club Aotearoa's project allows participants to create an interactive animation to tell their own stories of Matariki.

The Matariki Moonhack project is available in both English and te reo Māori, a language spoken by over 185,000 people, giving tamariki (children) an opportunity to explore their culture in their own Indigenous language.

The work undertaken by Code Club Aotearoa and Code Club Australia is ensuring that Indigenous voices are celebrated and shared near and far, so that children of all backgrounds are able to participate in Code Club in a way that is supportive and relevant to their lives.

Even though Moonhack is over for 2021, you can still code the projects. They are available at helloworld.cc/moonhack. **(HW)**



NICOLA CURNOW & REBECCA DONNELLY

Nicola is the national programme manager at Code Club Australia (@CodeClubAus). Rebecca is the education manager at Digital Future Aotearoa, a charity that runs Code Club Aotearoa (@CodeClubNZ).



■ Team S.P.A.R.K.S. were delighted to win a Judges' Award for Enthusiastic Newcomer in Duxford in 2020

MORE THAN ROBOTS

Huzaifah Zainon discusses her school's motivation for joining a global challenge for students aged 12–18; more than just a robotics competition, it equips pupils with soft skills such as motivation and resilience

The FIRST Tech Challenge (FTC) is an annual global robotics competition that challenges teams of students aged 12–18 to design, build, and code robots. Each year, participants around the world build their teams in September and receive their mission via an epic launch video.



© FirstUK.org

The first round of regional competitions take place in February, with the national finals scheduled for April. Pre-pandemic, one lucky team would win a place in a US-based competition in May. This year's challenge is called Ultimate Goal, and asks teams to build a robot that will deposit game pieces in goals of varying heights.

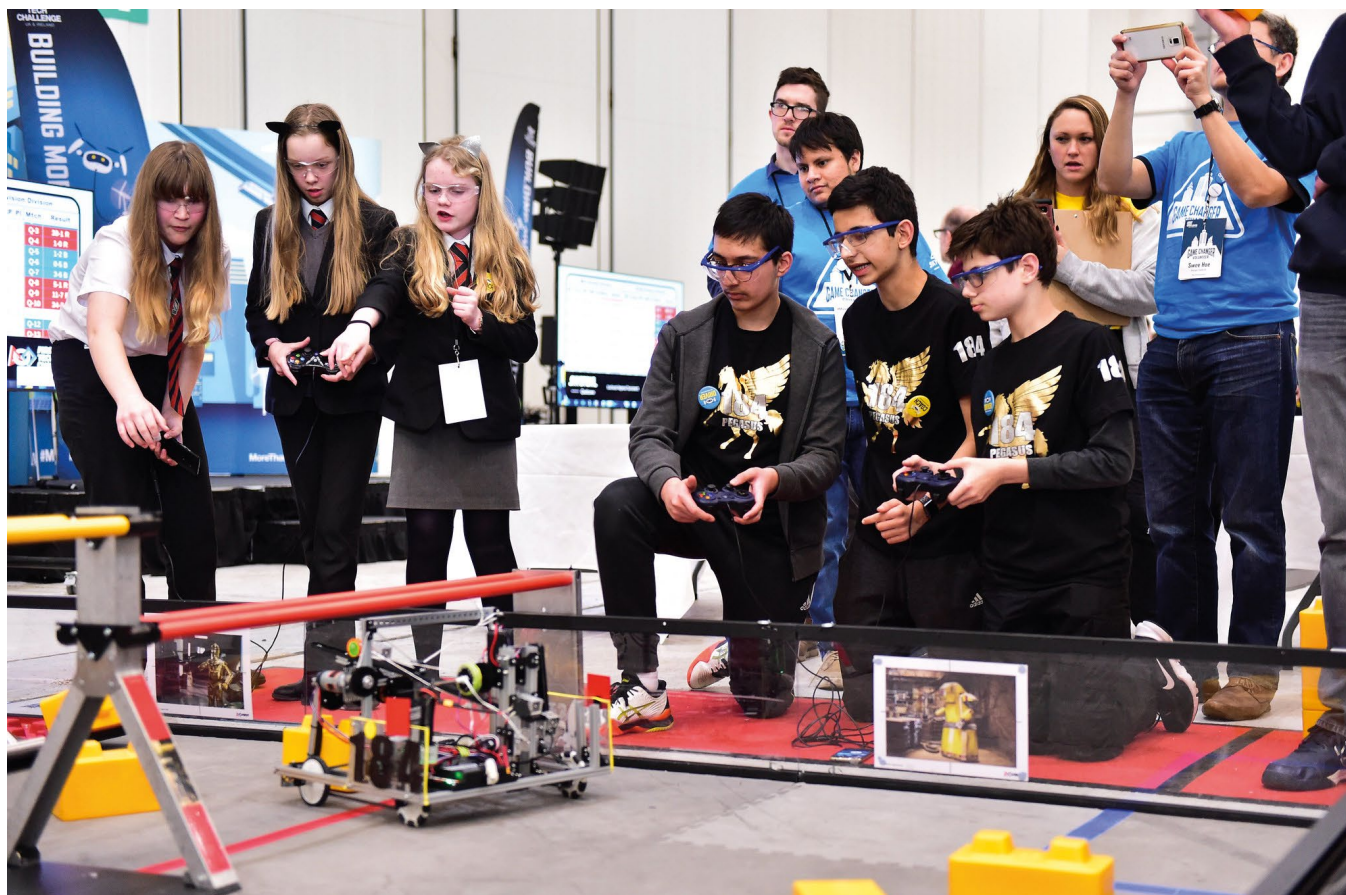
Why I signed up

I run a robotics club at my school. I started with the Raspberry Pi and then moved to kit-based VEX robotics, before discovering FIRST UK at a Computing at School meeting.

The first reason I signed up was the cost. For a small annual entry fee (at the time £100, but now £49) I was able to secure a bursary for the reusable kit. After three years, I have spent very little on the

kit because I don't believe in throwing money at a problem. I also like that the competition is organised by a single body, FIRST UK. This means there is a single deadline. With VEX competitions, I never got around to signing up for a competition as there were multiple possible dates, and it was easy to postpone signing up. As a team, you are never ready to compete!

The challenge also appealed to me because it is so much more than a robotics competition. Of course, students have to learn to think like engineers, asking themselves questions about how they can complete challenges such as picking up and dropping pieces into a specified scoring area. But they also have to think like a business, considering their pre-existing kit and the budget they have available to



© FirstUKong

purchase additional hardware to solve the challenges. They also have to be creative — the likelihood is that their solution won't work the first time and they will have to reconsider their approach. Resilience is a key skill here, too, as students have to be prepared to keep modifying their design.

Another benefit of this competition is that the FIRST Tech Challenge has been running in the US for many years. This means there are a huge number of ideas on the web posted by teams that have competed. The companies that produce the robotic kits, such as REV Robotics, also post a lot of helpful resources on their website. Having said that, it was still rather overwhelming in our first year of the competition, so we followed the instructions to build the kit robot. It has a limited capability, but it moved, and we were able to participate in a competition, which was a good experience.

The competition is sponsored by big names in technology, which made it feel really worthwhile to students. In our school, we are fortunate to have had a

mentor from Qualcomm and to have been inspired by volunteers from technology company Arm during competitions. As we are a rural school, pupils might not appreciate that the Silicon Fen around Cambridge is just on their doorstep!

Lastly, I like the ethos of FIRST: gracious professionalism, which links closely to our school's own PRIDE values of positivity, resilience, integrity, drive, and empathy.

Why pupils signed up

A team can include up to 15 pupils with multiple skills. This allows a group of friends with different interests to join together. As part of the challenge, a pupil could be designing the team's logo, or promoting the team to the local community. One student might also be a good record keeper, who keeps a log of all the activities, design changes, and decisions that the team makes throughout the year.

The challenge is an opportunity to represent the school in a competitive, team-based environment in the same way as a sporting team, but appeals to

those who might not normally be included in team sports. It is an opportunity to be creative problem-solvers. It is also an opportunity to come up with a cool project name. The team is currently called S.P.A.R.K.S. (Software Programmers Acquiring Robotics Knowledge at Soham).

Logan, a pupil, said, "FTC is a unique competition; I have not only gained knowledge in robotics, but I have improved my teamwork, 'gracious professionalism', and problem-solving skills. I have gained lifelong friends, and memories I will never forget. The memories and experiences I've had have helped shape me as a person, and I wouldn't trade them for the world."

If you're interested in signing up, find out more here: helloworld.cc/firstuk. 

HUZAIFAH ZAINON

Huzaifah is a software engineer who is now enjoying a second career educating the next generation of innovators at Soham Village College in the UK.



ROBOTICS: PREPARING OUR YOUNG GENERATION FOR THE FUTURE

Andrew Perrin shares how his school aims to prepare students for an uncertain future through robotics

In the two years I have been at my school, I have lost count of the number of times we have changed the curriculum. Technology is one of the most difficult subjects to teach, mainly due to its ever-changing nature and this need to constantly upgrade and redesign the curriculum based on the latest and newest advancements in the subject. Among all the new topics that we teach, such as 3D design and printing, or app and web design, one of the most prolific topics, and one that has found its way into most of our year groups, is robotics.

A recent addition to our curriculum has been our Future Skills programme. This is a rotation of activities and lessons that allow our children to develop skills, outside the standard curriculum, that will benefit them in the future or give them an insight into a particular vocation and help

them to flourish in the modern world. We sometimes use external experts, but most of the sessions are provided by teachers who share skills (maybe outside their normal teaching qualifications) with the children. My role is to expound the values of technology, and particularly robotics.

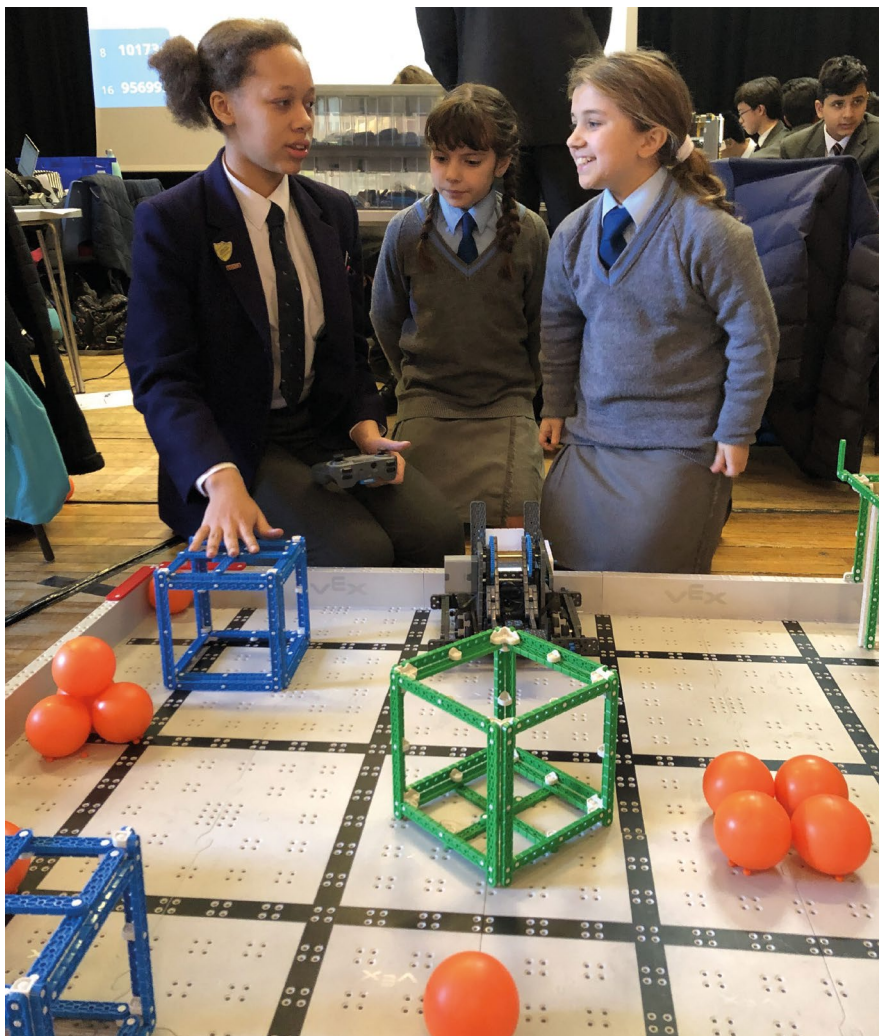
Why is robotics so important?

It is fair to say we stretch our pupils when it comes to teaching technology and computing — and robotics is a big part of that. We aim to prepare our students for careers in a technological world, and robotics and coding are huge areas, with many implications and opportunities for future employment. It's estimated that between 400 and 800 million people worldwide will be displaced by automation and have to find new employment by 2030. We try to connect our robotics teaching

to these real-world examples, inspiring students with outstanding technological achievements — from Amazon's automated warehouse robots to the first aircraft to fly on Mars. The considerable increase in the use of automation, and the associated skills required in robotics, programming, and AI, means that a growth in the employment market is inevitable, and the time to prepare our pupils for it is now.

VEX Robotics

We use the VEX Robotics system across our school (vexrobotics.com). It has hardware and software capabilities for all our age groups, and this provides a beautiful linear progression of learning, using something that is already familiar to our pupils as they progress. Through events such as Bett, we have seen a huge increase in the range of robotics products



■ Belmont students competing in a VEX IQ regional competition

available, such as Invent-a-Bot and Bit:Bots. However, I decided to use VEX after seeing how successful it was as a club-based activity in a previous school I worked at, allowing pupils to compete in national and international VEX Robotics competitions.

At these competitions, pupils could form teams and progress from Year 7 to 13 (ages 11–18) through the various categories, and develop innumerable skills and qualities that are completely separate from the normal teaching curriculum. When I finished training, I worked with a group of three boys who started doing VEX in Year 7 and kept it going throughout their time at the school, before leaving to go to one of the country's top universities to study robotics. There is no doubt that they benefitted from all their education. But this achievement rested solely on the background of the voluntary VEX club they had been part

of for five or so years. For me, that meant that when a Future Skills curriculum was proposed, I jumped at the idea because it allowed me to take the robotics program from being just an after-school club to being part of every student's education.

The future

It's still early days at Belmont Prep School, but we've already seen robotics move from being just a Year 7 club to being incorporated into the Future Skills curriculum. This includes a focus on the making and building side of VEX IQ and also on the coding side of robotics, with Year 5s (ages 9–10) attempting to master the art of programming a virtual robot.

Our goals are now twofold. The first is to incorporate robotics into other year groups, to create a linear approach to the complexity of the subject. This



ANDREW PERRIN

Andrew served for twelve years as a telecommunications engineer in the British Armed Forces. He was one of the pioneering students who took part in the Troops to Teachers scheme, qualifying as a design and technology teacher. He is now the head of computing at Belmont Prep School, part of the Mill Hill Foundation, in London, UK.

“ WE INTEND TO TEACH OUR STUDENTS THAT ROBOTICS IS TRULY A CAREER FOR THE FUTURE

may include trying to incorporate a different coding language, stepping up to the most advanced level of VEX, or even seeing what other products are available. The options are wide-ranging.

The second target is to see if we can introduce robotics earlier. Can pupils handle coding at a younger age? I hope that the answer is yes, with the introduction of VEX Go into our Future Skills curriculum in September 2021. This will give our youngest pupils their first taste of robotics and also start them developing other key skills such as teamwork, problem-solving, and programming.

There is only so much time in a busy curriculum, but we intend to promote robotics and coding within our computing programme and teach our students that this is truly a career for the future. 

GAME-BASED LEARNING IN THE COMPUTING CURRICULUM

Can playing educational video games help children develop their understanding of computing concepts, or are they just a tool for engagement?

In the previous issue of Hello World, I examined the use of retro video games in the classroom and discussed opportunities to use them to teach elements of the computing curriculum.

While commercial video games such as these can enable children to develop their understanding of computer science concepts, many educational games also exist that can support children to enhance their knowledge further. This pedagogical approach is referred to as game-based

learning. Educational games often include a range of gaming elements, such as rewards, scores, and levels. This differs from the gamification of learning, which involves using gaming elements to enhance the classroom environment, such as awarding points for demonstrating certain behaviours. In this article, I'm going to be discussing the benefits of game-based learning for motivation and engagement, and looking at when and how to use games to best aid learning.

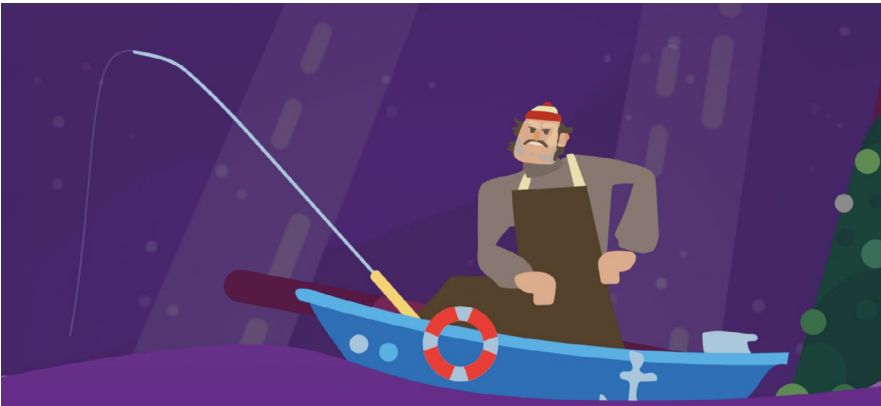
Motivation and engagement

As gaming is an activity regularly enjoyed by children, with those aged 8–11 in the UK spending 9–10 hours on average gaming each week, it is often cited as a way to motivate children when learning a subject, and children regularly outline how much they enjoy game-based learning as part of their computing lessons. For example, Barefoot Computing, which includes a number of web-based games for teaching aspects of the computing curriculum, has received excellent feedback from teachers on how engaged children are when undertaking the activities. Educational games increasingly feature characters and settings already familiar to children, which can help with initial engagement. Examples of these are the Minecraft and Frozen-based games used to develop basic programming concepts within Code.org. Games from Barefoot Computing and Code.org can be used both in the classroom and at home.

The use of game-based learning has also been observed to help with engagement in teenagers who have experienced difficulties with accessing traditional educational settings. For example, by requiring the player to develop their IT knowledge in order to increase their score within a specially designed game, users can be motivated to undertake further study of that subject area.¹

Enhancing knowledge

The National Foundation for Educational Research's 2013 report, titled 'Game based learning: latest evidence and future directions',² outlined how game-based



The Phisherman

AN UNDERWATER ADVENTURE

Start

■ Barefoot Computing's Phisherman game includes detailed lesson plans to use alongside the game

REFERENCES

1. Schmitz, B., Czauderna, A., Specht, M., & Klemke, R. (2011). Game Based Learning for Computer Science Education helloworld.cc/games1
2. Perrotta, C., Featherstone, G., Aston, H., & Houghton, E. (2013). Game-based Learning: Latest Evidence and Future Directions (NFER Research Programme: Innovation in Education). Slough: NFER helloworld.cc/games2
3. Papastergiou, M. (2009). Digital Game-Based Learning in High School Computer Science Education: Impact on Educational Effectiveness and Student Motivation helloworld.cc/games3
4. Takeuchi, L. M., & Vaala, S. (2014). Level up learning: A national survey on teaching with digital games. New York: The Joan Ganz Cooney Center at Sesame Workshop helloworld.cc/games4
5. Mathrani, A., Christian, S., & Ponder-Sutton, A. (2016). PlayIT: Game Based Learning Approach for Teaching Programming Concepts. *Educational Technology & Society*, 19 (2), pp.5-17 helloworld.cc/games5

learning can improve student engagement and motivation, although significant research is still needed into how video games impact on learning. In particular, further investigation into the impact of specific game mechanics, rather than games per se, would allow for a greater understanding of how games can be used to enhance pupils' knowledge.

Despite this need for further research, some studies have already outlined how using gaming can help students improve their knowledge of computing. With secondary-aged pupils, the use of games has been shown to be more effective in developing students' knowledge of computer memory concepts, such as the role of secondary storage, when compared to a non-gaming pedagogical approach.³ More wide-ranging studies, such as those published by the US-based Games and Learning Publishing Council,⁴ have outlined how results on improving learning in STEM subjects have been mixed, with learning gains in game-based activities related to mathematics significantly higher than those seen in science.

When to use games

When choosing games for use in the classroom, it is important to ensure that the academic content is an integral part of the game, rather than an add-on. This will help ensure that children are clear about what they are learning and not solely focused on other parts of the game. Learners also need to be given guidance in conjunction with the game, which could

IT IS VITAL THAT THE GAMES FORM PART OF A COHERENT SCHEME OF WORK, WITH PROGRESSION CLEARLY IDENTIFIED

include further details around content introduced, or perhaps a discussion around how to implement a programming concept. In addition to this, where a game contains a number of activities, it could be used to provide a focus for other learning within class, such as unplugged activities away from the machines, or the production of digital artefacts to demonstrate learning. For example, Barefoot Computing's recent Phisherman game, which helps children learn about phishing scams, includes a detailed lesson plan for teachers to use in conjunction with the resource.

Studies examining games focusing on programming concepts, such as Lightbot, have outlined the benefits of using such games when introducing a topic, which, due to students' enjoyment of the game, is then likely to motivate them to learn further.⁵ Alternatively, the games could be used for pupils to revisit a topic after a period of time, for example, if they haven't undertaken programming activities in their computing lessons during the previous term. Either way, it is vital that the games form part of a coherent scheme of work, with progression clearly identified, rather than children simply using the games to have a play. Similarly, for younger children, games could form part of the continuous

provision of a class, although the learning objectives from the activities need to be carefully identified, along with appropriate supporting tasks.

Finally, it's worth noting that game-based learning does have its limitations. Most games related to computing follow a linear path, which means all learners experience the same problems, in the same order, rather than having questions tailored to the next stage in children's learning. The extensive use of games can also mean that opportunities for children to creatively develop digital content of interest to them are limited, which can limit children's learning experiences and could lead to the

full potential of computing as a subject area not being realised.

What are your thoughts on using game-based learning to teach computing? Have you introduced games into your lessons effectively, or do you prefer other pedagogical approaches? Do get in touch on Twitter [@computingchamps](https://twitter.com/computingchamps). **(HW)**



NEIL RICKUS

Neil is a senior lecturer in computing education at the University of Hertfordshire, a primary education consultant for the BCS, and the founder of Computing Champions. He is a CAS community leader, a CEOP Ambassador, and a Raspberry Pi, Google, and Microsoft Certified Educator ([@computingchamps](https://twitter.com/computingchamps)).

HOW GAMES CAN INSPIRE THE NEXT GENERATION OF CHANGE-MAKERS

Kim Chouard shares how game-based learning can inspire youth to become agents of change building a more equitable and sustainable world

We are facing unprecedented global crises, and growing challenges like climate change won't make it any easier for the next generation to build an equitable and sustainable world. Despite this, young people globally have shown us their determination to make a positive change.

In my role in the philanthropy team at Salesforce, we've leveraged the power of purpose-based and game-based learning to create a free and open-source gamified curriculum for educators and volunteers: Age of Makers. It aims to empower youth from middle to high schools to use new skills, such as coding, 3D printing, digital literacy, and music production, to positively impact the world around them. In this article, I would like to share the benefits of these approaches to creating change-maker students, and look at how you can try both the game and the approaches out with your learners.

GET INVOLVED

Whether it's in the classroom or after school, you can run an Age of Makers programme locally and for free, with in-depth guidance from our online guide (helloworld.cc/ageofmakersguide).

The game is open source, so you can contribute by coding it, translating content, or envisioning a new curriculum connecting new goals to new topics like AI or painting (helloworld.cc/ageofmakers).



Purpose-based learning

The traditional approach to education starts by trying to teach students specific skills, without necessarily showing them why they should care about acquiring them. Without context, students often lack the intrinsic motivation they need to grasp a topic deeply. Edtech entrepreneur Lee LeFever offers an alternative to this approach in his book *The Art of Explanation*, describing how understanding can be seen as a balance between the 'how' and the 'why':

- The less you understand about a topic, the more you need context to understand why it matters
- As you start understanding the topic better, you can start digging deeper and look more precisely at how it works

Purpose-based learning flips the traditional

education system's paradigm by starting with the 'why'. The Age of Makers game and its programmes are built on these principles. The game starts by introducing some of the world's biggest problems, using powerful and simple frameworks such as the UN's Sustainable Development Goals. Students then choose a specific topic they want to address. Only then do they start learning new skills in order to solve the problems they care about. As a result, students have agency in their learning and they see a direct connection to why they are learning specific information and skills.

For example, one group of students 3D-printed boat prototypes designed to collect and recycle the ocean's trash. Another student produced a song denouncing systemic racism. Lars Blok, a teacher in the Netherlands,

THE AGE OF MAKERS GAMIFICATION PRINCIPLES



Learning Maps open up whole new worlds

One challenge for students is not a lack of places to learn and upskill independently, but rather the difficulty in connecting topics together in a meaningful way.

The Age of Makers game solves this challenge by providing a Learning Map: a visual way to see the different topics at hand and how they all relate to each other. Try to create your own maps to guide your students through their learning. It could be as simple as something printed out on paper or on a slide.



Journeys tell their own story

Journeys can be seen as adventures that students go on. In Age of Makers, each Journey follows a specific pedagogic order. After finding a challenge they want to solve in their communities, students teach themselves different skills that they'll use to build a concrete solution. Finally, students showcase their solutions in front of an audience, where they shift from being consumers to creators.



Ages guide learning journeys

It's important to phase how much you share with your students at each step of their learning journey, to avoid overloading them. In the game, we've built in the concept of Ages. At each Age, students see only the part of the map that's relevant to how far they are in the game. At first, only a very small part of the map, with just a few Quests, is visible. As students complete Quests and straighten out their 'why', they reach new Ages, and more of the map gets visible each time, showing them more of the 'how'.



Quests offer limitless learning opportunities

To make learning more gradual, it's best to break it down into bite-sized learning adventures for each Journey. In Age of Makers we call these Quests, and they can take different forms, from a video to an interactive showcase with examples to inspire the students. The game can embed a guide on top of any external website (e.g. MakeCode, BandLab, Tinkercad, etc.), so as an educator, the amount of content that can be leveraged through the game is pretty much endless.

“ WITHOUT CONTEXT, STUDENTS OFTEN LACK THE INTRINSIC MOTIVATION THEY NEED TO GRASP A TOPIC DEEPLY

explained how this benefitted his students: “It was inspiring to see the students become more confident while finding their own voice.”

Gamification

On top of a purpose-based approach, Age of Makers incorporates gamification principles. Gamification creates similar experiences to the sense of challenge, thrill, and accomplishment people feel when playing games, to motivate and engage users. It is the application of game-design elements and principles, such as scoring points, appearing on a leader

board, and earning badges (see boxout).

The game itself is only part of the equation, though. Programmes are built around it so that educators can run multi-week activations locally, while partnering with volunteers wherever possible. These programmes have been used by 600 students all around the world, from Paris to Sydney passing by San Francisco, with students ranging from middle to high school.

What about you — are you already using games in your teaching as a force for change? Are you empowering your students to become change-makers? [GHW](#)



KIM CHOUARD

Kim works in the philanthropy team at Salesforce. He is a social entrepreneur and artist on a journey to imagine an alternative education approach, leveraging games and technology to empower young people to be critical thinkers (@KimChouard).

GOING BACK TO BASICS

Rebecca Franks looks at how to get beginners started on a physical computing project

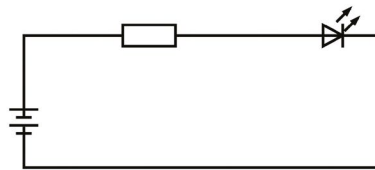
My journey into physical computing has taken me in all sorts of directions, but it really started to click with me when I went back to basics. I have decided to try this approach with a group of beginner digital makers, who are learning to power LEDs and set up simple circuits to make an art project, to see if it helps them become more independent with their own projects. Over the next few issues I will share my session ideas and progress with you in the hope that it will encourage you to have a go yourself!

It all starts with a breadboard

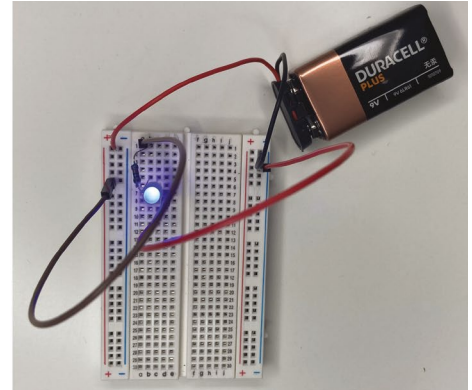
Many physical computing projects begin with a breadboard. A breadboard is typically made up of terminal strips to hold the main components, such as LEDs, and bus strips to power the components. A breadboard allows creators to prototype their ideas before making them more permanent by soldering — sticking wires and components together using melted metal.

Each row of terminal strips, marked using numbers, is electrically connected. There is a channel in the centre of a breadboard which provides an airflow gap for components that need them. Electricity doesn't flow through the channel; this is a common misconception that beginner learners have, and I often forget, too!

Breadboards are great once you understand how they actually work, but



■ A circuit diagram for an LED (left) and the same circuit created on a breadboard (right)



this is often the first barrier that newbie makers face. A basic circuit diagram can look very different when it is created on a breadboard (see images above).

Session one

My goal for the first session with my beginner maker group is to develop their understanding of how the breadboard works. They will start with a circuit using a battery, a resistor, an LED, and male-to-male jumper wires; see the images for examples. They will then add further LEDs to develop their series circuit. This will show how the brightness of the light is reduced with each new LED.

Next I will introduce a parallel circuit so that the group can see how the current flows through branches. This will also help to build their knowledge of the breadboard, as they will need to think carefully about where to position the components.

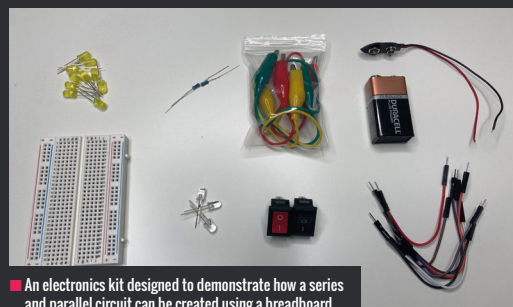
My plan is to use circuit diagrams along with breadboard images to support the group. I think that learning circuit diagrams is important, because you might want to learn something new later on, or add something extra to your circuit. Electronics tutorials often include circuit diagrams, which can be quite off-putting to beginners if they have never seen one before.

I often find myself asking 'why' or 'how' when digital making, and it is not always easy to find the correct answer. By starting at the very beginning with my group, I hope that they will learn this as they go along. This should help with their confidence and make them more independent when they create their final projects.

Join me in the next issue of Hello World, where I will give feedback on how the session went and reveal my plans for session two. **(HW)**

SESSION ONE KIT LIST

- 10 x yellow LEDs
- 3 x multicoloured LEDs
- 2 x 1K resistors
- 3 x crocodile clips
- 1 x small breadboard
- 2 x on/off switches
- 10 x male-to-male jumper wires
- 1 x 9V battery connector
- 1 x 9V battery



■ An electronics kit designed to demonstrate how a series and parallel circuit can be created using a breadboard



REBECCA FRANKS

Rebecca is a learning manager at the Raspberry Pi Foundation. She writes resources for the Teach Computing Curriculum and has over 15 years of experience teaching computing.

REFLECTIONS ON DECOLONISING THE COMPUTING CURRICULUM

Yota Dimitriadi discusses the importance of widening global perspectives in computing education

© Nazara / iStockphoto.com

Decolonising the computing curriculum sits within broader considerations around equity, diversity, and inclusion. It stems from the need to cast a wide perspective on how the curriculum (the ‘what’ and the ‘why’) and associated pedagogies (the ‘how’) include and represent all students. This wide perspective embraces considerations around knowledge sharing, construction, and representation. Such considerations should not be add-ons, but integral to an accessible curriculum, making the educational environment equitable, inclusive, and welcoming for all.

However, decolonising brings discussions around culture to the forefront of the educational agenda. It focuses specifically on addressing racism and colonialism, and the sharing of knowledge that can be

viewed as unrepresentative for its recipients, or privileged as it excludes crucial narratives. An approach that does not acknowledge these considerations can alienate groups, as students may not see themselves reflected in the topics, and does not equip them well to work in a diverse world. For instance, data from Kemp et al. (2018) points to Black students being proportionally under-represented in studying computer science. In this article, I will focus on representation and symbolism as a way of reflecting on how global perspectives and marginalised narratives can be included in our computing curriculum planning.

Representations

These ideas form part of wider discussions around “culturally relevant pedagogies” (Ladson-Billings, 2006)

and “culturally responsive teaching” (Gay, 2000). For instance, reflecting on the diversity of the demographic profile of the examples and role models we discuss in our computing classroom has been a key consideration in gender-balance interventions. Extending our intersectional examples further, with a focus on ethnicity and race, and acknowledging the contributions of African philosophers and Indian and Arab mathematicians in the development of technology, for example, can encourage discussions about the sociopolitical context of how knowledge has travelled across the world, and questions about equity and social justice can emerge. A typical example is referring to the history and etymology of the word ‘algorithm’, derived from the name of the Persian



■ Students can create an algorithmic design for the ceremonial haka

► mathematician Muhammad ibn Musa al-Khwarizmi, which can act as a starting point for discussion of current algorithm structures and possible inequalities.

The history of cryptography and steganography also offer opportunities to celebrate multicultural perspectives in our computing lessons and discuss not only the techniques used (e.g. a Caesar cipher), but also the cultural and historical context where such practices were located (e.g. the Roman Empire). Using examples that include respectful consideration of other cultures — for example, programming Islamic art patterns, or algorithmic design for choreography that extends to discussions around the Māori ceremonial

“ WE NEED TO CAST A WIDE PERSPECTIVE ON HOW THE CURRICULUM AND PEDAGOGIES REPRESENT ALL STUDENTS

haka or the Chinese Dance of a Thousand Hands — provides a platform to explore cultural and heritage practices as resources for learning. Such examples have the potential to go beyond cultural appropriation towards a global outlook in which a dialogue around cultural understanding of the history and meaning of these resources, especially around the history of Indigenous cultures, can be promoted.

Ethnocomputing considerations

This relationship between culture and computing is discussed extensively in the field of ethnocomputing. The term is associated with Professor Ron Eglash's research (1999, 2006) which explores the fractal geometry of traditional African architecture and develops interactive simulation tools that students in American inner-city schools used to

REFERENCES

- Eglash, R. (1999). *African Fractals: Modern Computing and Indigenous Design*. New Brunswick NJ: Rutgers University Press.
- Eglash, R., Bennett, A., O'Donnell, C., Jennings, S., & Cintorino, M. (2006). Culturally Situated Design Tools: Ethnocomputing from Field Site to Classroom. *American Anthropologist*, 108(2), pp. 347-362.
- Gay, G. (2000). *Culturally responsive teaching: Theory, research, and practice*. Teachers College Press.
- Kemp, P., Berry, M., & Wong, B. (2018). *The Roehampton Annual Computing Education report: Data from 2017*.
- Ladson-Billings, G. (2006). "Yes, but how do we do it?" Practicing culturally relevant pedagogy. *White teachers/diverse classrooms: A guide to building inclusive schools, promoting high expectations, and eliminating racism*, pp. 29-42. Stylus Publishing, LLC.

and culture in the British textile industry.

Reference to culture and 'cultural capital' are included in the intent of the UK Ofsted framework and its focus on "a broad, rich and deep curriculum". The debate surrounding the aim and interpretation of 'cultural capital' in that framework is well documented. However, the inclusion of examples and resources that reflect local communities, acknowledgement of minority perspectives around unequal access, engagement in a dialogue around equity, and a curriculum offering intersectional and accessible solutions and pedagogies



■ A page from al-Khwarizmi's *Algebra*

create computational models of fractals. They either replicated original Indigenous designs or developed their own new creations. Examples used with students included replicating computational models of cornrow hairstyles and Native American weaving patterns. These themes gave the students opportunities to reflect on their own cultural identity and its relationship to maths and technology. This intervention raised the achievement levels of participating minority students in maths, and positively influenced their technological career aspirations.

The research promoted the complexity of those Indigenous cultural processes and artefacts that may traditionally

“ REFERENCES MIGHT HIGHLIGHT EMPIRE CIVILISATIONS BUT ALSO INCLUDE CITATIONS TO INDIGENOUS CIVILISATIONS

have been considered primitive. It is an important consideration in our selection and discussion of cultural examples.

It is also relevant to consider the diversity in the global examples we choose to include as part of decolonising the curriculum. There is scope to reflect further on how the references we make highlight empire civilisations, but also include citations to Indigenous civilisations. For instance, referring to origami art to introduce algorithms, or the Antikythera mechanism to discuss computers, are examples taken from two colonial empires that have contributed and shaped the complex history of technology innovations. It is important to include such examples and broaden our global perspective, but also to acknowledge the context and background from which they emerge. Cultural perspectives can be relevant to our target audience and can be combined with discussions around local histories and realities. For instance, the Native American weaving example in Eglash's article, which was of social relevance to his participants, can provide a starting point for examples with more social relevance to different audiences. It can include discussions around the weaving traditions from the north of England and considerations about technology, power,

can support students to make connections between knowledge and power, and empower themselves and their communities. Support for educators and acknowledgement of possible challenges in incorporating such approaches needs to be promoted and developed further.

Ron Eglash discusses *Computing for generative justice: decolonizing the circular economy* (13 July) with Raspberry Pi at helloworld.cc/equityseminar. (HW)



YOTA DIMITRIADI

Yota is an associate professor of technology enhanced learning and computer science education at the University of Reading. She is a national teaching fellow of Advance HE and the Reading CAS SEND Community Leader (@yotadimitriadi).



■ The virtual event brought together 800 volunteers, teachers, and students

EXPANDING ACCESSIBILITY IN COMPUTING WITH GOOGLE'S CODE NEXT

Omnia Saed from Google's Code Next team shares the story of engineering teacher Nick Anaya and his school's effort to expand accessibility

Nick Anaya was a full-time engineer when he began volunteering with Robert Lindblom Math and Science Academy's robotics club in Chicago, USA. The work transformed Anaya's career trajectory. "Just so you know, Nick walks the talk. He volunteered for seven years at Lindblom before he decided to flip over and become a teacher," says Matthew Houghteling, a computer science teacher at Lindblom. "For one full school year, he had two full-time jobs."

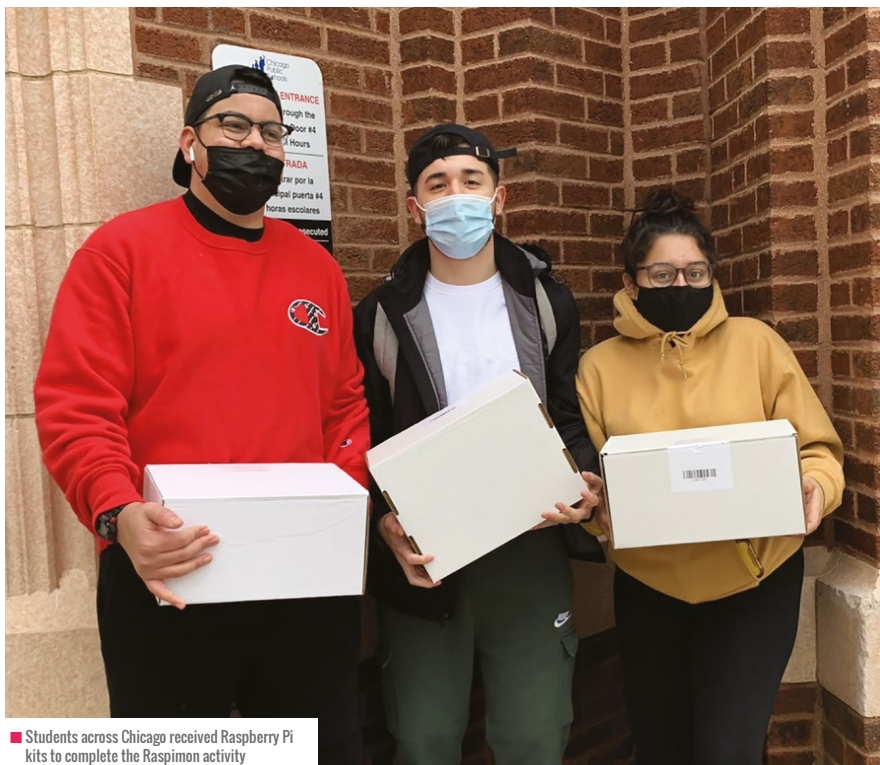
Today, Nick supports the roughly 1,400 students who attend the academy as an engineering teacher. He's the type of teacher that is excited by the work and

in many ways, it is a reflection of the students he teaches. "Lindblom is a school where whatever you want to be great at, there are great programs for that," Nick says. "We have the best kids doing the coolest stuff."

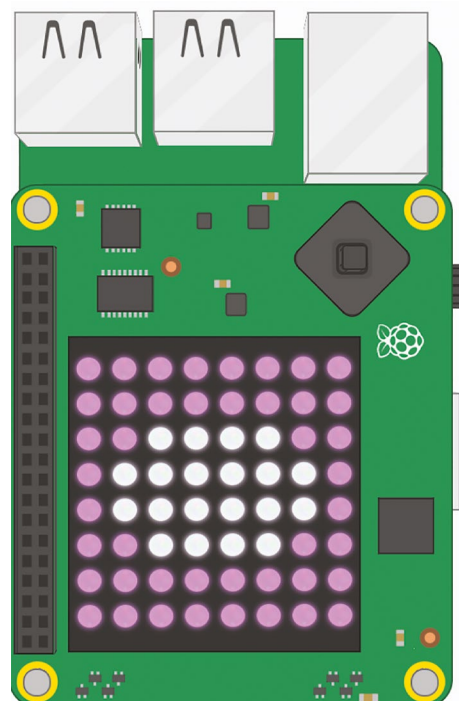
Google's Code Next and Raspberry Pi

Last December, Nick, alongside 800 students and staff from across Chicago public schools, participated in Computer Science Education Week (7–13 December). One event in the week was hosted by Google's Code Next (codenext.withgoogle.com) in partnership with the Raspberry Pi Foundation. Code Next is a free computer

science education programme in the USA for Black and Latinx high-school students, with a mixture of in-person and online programmes and events. Between 2011 and 2018, Black and Hispanic college students each only made up 3 percent of computer science graduates in the USA — and Code Next is working to change that. The programme provides students with the skills and inspiration needed for long and rewarding careers in computer science. "We aim to provide Black and Latinx students with skills and technical social capital — that web of relationships you can tap into," said Google Diversity STEM Strategist Shameeka Emanuel.



■ Students across Chicago received Raspberry Pi kits to complete the Raspimon activity



■ Students created a virtual monster powered by a Raspberry Pi

STUDENTS WITHOUT PREVIOUS ACCESS TO HARDWARE WERE ABLE TO INTERACT WITH THE CURRICULUM IN A NEW WAY

The event tasked participants to create their own Raspimon — a virtual LED-animated monster powered by a Raspberry Pi using Python. It opened the eyes of educators about the possibilities that the Raspberry Pi offers. “The problem we’ve gone through with remote learning is that the kids aren’t able to interact in the same way as they did before,” Nick explains. “But with new platforms, the goal is to bring the students together.”

Expanding accessibility

In preparation for the event, Google donated over a thousand Raspberry Pi kits to students across Chicago. As a result, students without access to hardware were able to interact with the curriculum in a way they had never been able to before. For teachers like Nick Anaya, the possibilities that level of access brings are endless: “Lindblom has four different

tracks related to computer science and engineering that kids can take. We run a web track, a video game programming track, a networking track, and an engineering track,” says Nick. The goal in a virtual environment is to get students engaged and working with one another. “We had an aha moment when we realised we could incorporate Pis in all four tracks,” says Matthew. “We realised with the event, and the programming in the curriculum, that we could leverage the hardware in a much bigger way than we had thought about before.”

And that’s the lesson here for the Lindblom staff, and for teachers across the country trying to engage with students: we need to make room for innovation, discovery, and curiosity when we’re apart, so that the tradition can continue when students are back together. “The hope is when the kids are back in person ... and

someone in the video gaming platform is going to have a kit and they’re using it for a very specific function, they might walk over to a networking kid or robotics student,” explains Nick. “With all of these different types of applications, we end up moving towards an understanding of platforms in a very different and interesting way, which hopefully gets the kids excited.”

As schools work to expand accessibility to resources in a distance learning environment that disproportionately impacts economically disadvantaged students, it’s teachers like Nick who drive the movement forward. “None of this stuff gets fixed. None of it gets any better unless we have that kind of dedication,” says Nick. “We don’t educate a kid in a day; we expose a kid and get them excited in a day, and then we need, you know, we need every Saturday. The one thing I would say, and this is always — we need more people. The hardware’s awesome. We really, really appreciate it. It gives us an opportunity to do something we couldn’t before, but we need more — more engineers, more business-minded entrepreneurs, and so on.”

For a copy of the activity and curriculum, visit helloworld.cc/raspimon. (HW)

LOOKING FOR THE NEXT PROFESSIONAL CHALLENGE AS A COMPUTING EDUCATOR

At a certain point in our careers, we all ask what's next; **Val Quaye** explores what the next steps of her teaching career could look like, and the power of conversations

I have been teaching ICT and computer science to students aged 11–18 for the last twelve years. During my teaching career, I've been a learning coach (focusing on supporting educators and sharing best practice) and a pastoral head of year. I'm now at the stage of my career where I'm getting itchy feet and asking myself the big question, "What comes next?" This article is a summary of my research and my thoughts on the pros and cons of each option. I hope that this article will help to show you some of the options available if you are asking yourself the same question.

More qualifications?

I started by updating my CV to identify any areas for improvement in my professional development. I was looking to explore the different options that would refine my teaching practice and as a result, make learning even more enjoyable for my students. The first route I explored to do this was gaining more qualifications.

I first looked at pursuing a master's degree in education. This is a globally recognised qualification which many teachers have, and is often seen as a desirable qualification when applying for a teaching position. A master's involves a big investment of time and money, and I wondered if it would help me to develop my teaching practice and inject more fun into my classroom, or if it would be focused more on pedagogy and educational policy.

I looked to other formal qualifications, and a colleague suggested that I find out more about the National Professional Qualification (NPQ) for Middle Leadership (helloworld.cc/NPQ). My initial reaction was, I'm not a middle leader, so I'm not eligible to even consider this qualification. However, my colleague showed me that the qualification is open to both current and aspiring middle leaders (for example, subject leads and heads of department).

This qualification will be reformed from September and broken down into three specialist areas. The area which stood out for me was Leading Behaviour and Culture. I wondered if, through the NPQ, I could design a project that would intertwine my interest in coaching and positive psychology, and measure the impact on the whole school community. Some of the questions I had were: What would learning look like if coaching conversations and positive psychology were part of everyday language? What impact could this have on students' mental health? How would individuals feel if when they asked for help, they were empowered to decide on their next steps? I believe that, through this route of the qualification, I could help raise academic attainment in the classroom, and could give students their own success formula.

Upskilling

With the route of gaining more qualifications being so costly and time-intensive, I also investigated upskilling myself in

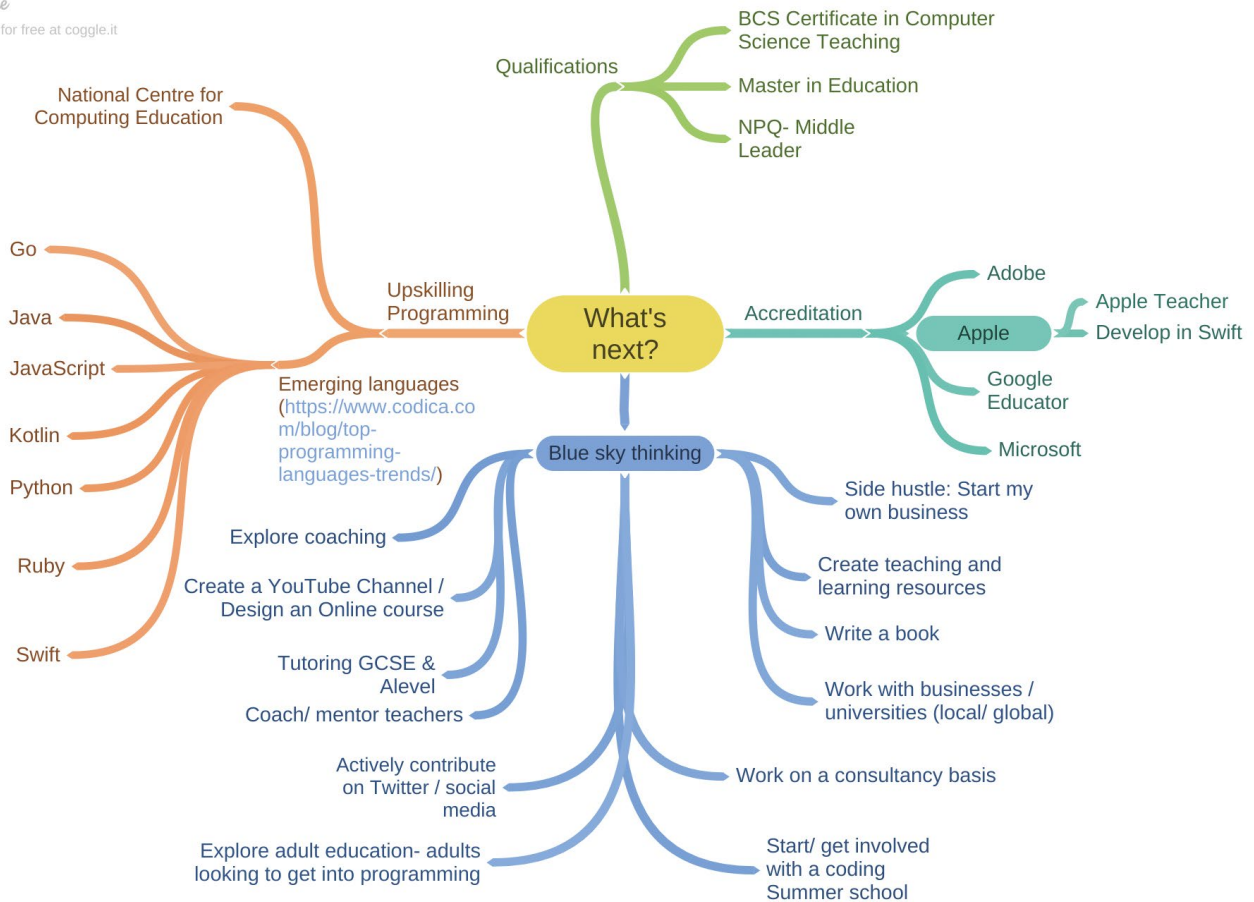
programming. As an educator who is an avid Apple (iOS) lover, I have dabbled with Swift Playgrounds and Xcode, but hadn't invested much time into getting to grips with them properly. I found a free Apple Teacher webinar that introduced educators to the basics of using both. I found it really valuable to hear from educators about the strategies they used to overcome some of the hiccups their learners faced. They also shared how learning this language had given their students the knowledge and confidence to create their own apps and release them publicly. I thought that was really exciting, and could help learners see how what they are learning in lessons can permeate into life outside of the classroom.

The power of conversations

With such a large selection of possibilities, I sought out a learning coach at my school to help. We used the GROW model to structure the conversation, which works through your Goal, the Reality of your situation, your Options, and your Way forward. For part of the conversation, I was asked to close my eyes, visualise myself twelve months from now, and answer questions such as:

- If there were no consequences, what would you do?
- Which choice energises you the most?
- Imagine a friend came to you with this situation — what would you tell them?

The visualisation and questioning helped me to clarify what I wanted to explore.



■ Mind mapping my potential next professional steps

I felt that the most powerful question was: What I would do if there were no consequences? As a result of this blue-sky thinking I answered this question with many possibilities, which you can see in my mind map! From this list of possibilities, the coach asked me to identify three ideas that I would be excited to explore further. I also spoke to other colleagues about my options, including my head of department, who has been teaching for more than 30 years. I asked him what advice he would give himself, if he had a second chance. He reminded me that the terrain of computer science is constantly changing; new and emerging languages and technology are always on the horizon.

What's next?

Before both of these discussions, I believed that the master's was a compulsory requirement to be a successful teacher. However, the conversations helped me see that the three ideas I really wanted to invest

in were my programming skills, coaching skills, and sharing ideas and resources online. I plan to work through the *Develop in Swift* professional learning course offered by Apple (helloworld.cc/swift). To make myself accountable, I will create a range of classroom resources focused on teaching students Swift, and share these through Twitter, the Computing at School hub, and with the South East Asia Computer Science Teacher Association (seacsta.net).

I want to develop coaching and positive psychology in my classroom and school community, so that students continue to strengthen their growth mindset and become empowered to make even more positive choices in their lives. In the longer term, I am going to find out more about the NPQ Middle Leadership qualification.

If you are asking yourself what's next, take a deep breath and ask yourself the question, "What would you do if you could do anything?" You might just be surprised by your answer. (HW)



VAL QUAYE

Val teaches ICT and computer science to students aged 11-18. She is currently teaching at the Alice Smith School in Kuala Lumpur, Malaysia. Val is a Raspberry Pi Certified Educator and completed ILM's Performance Coaching for Leadership: Executive Coach Training in 2020 (@ValQuaye).

EDUCATING EDUCATORS

Sujatha Gunja and **Daniel Hickmott** share tips for teaching computing by drawing on their collective teaching and research experience in schools and tertiary education

Every Australian child between the ages of 5 and 14 must learn computing through the Australian Digital Technologies curriculum. We have supported Australian teachers with learning the content, pedagogy, and technical knowledge of the Digital Technologies curriculum since it was first endorsed in 2015, in a variety of contexts. These contexts have included professional learning programmes, tertiary education, outreach programmes, and research. Most recently, we have taught in units in the Master of Education (Digital Technologies), or the MEd as we call it for short, at the University of Sydney, Australia. In the MEd we have taught educators with a variety of computing experience, including computing professionals who have recently started teaching, and teachers with decades of teaching experience who have never taught computing before.

Our tips for teachers

Teaching computing can feel intimidating, particularly when you are new to teaching in this area. In this article, we have identified six tips for teachers with any level of computing experience, drawing on research and reflecting on our experiences of educating educators.

1 ACKNOWLEDGE THE EXPERT BLIND SPOT

Computer science has many abstract concepts and technical skills that students need to navigate. As educators get more familiar with a subject discipline, they may not be able to see the learning with novice eyes, and may not recognise the difficulties involved in learning. This phenomenon, termed the “expert blind spot” (Guzdial, 2015), can make it difficult for teachers to spot their students’ misunderstanding of important concepts.



© Shutterstock/Andrew Rybakov and Craig Carroll, Gink Academy

Learning programming can be challenging for many students, and encountering misconceptions is an important part of their learning journey. Knowing what these misconceptions are and how to identify when students hold them is important. Developing pedagogical content knowledge expertise in the discipline will help teachers spot misconceptions in their students’ learning.

2 DIFFERENTIATE LESSONS FOR YOUR LEARNERS AND USE PEER INSTRUCTION

We have observed that students and teachers often have a wide variety of experience and interest in computing, particularly when it comes to programming. We recommend providing a variety of activities that are targeted towards different levels of computing experience and include some opportunity for self-directed learning. For example, in our teaching we pair students with similar levels of confidence and give them a range of activities, including Parson’s Problems and code-reading

exercises, that they can choose to complete.

We also recommend encouraging peer instruction by using strategies such as pair programming, particularly with larger classes where time to support students individually is limited. We suggest including students’ participation in discussions in their assessment. We did this in the MEd and have found our students to be very engaged in discussions, even finding that they would often answer their peers’ questions without any intervention from us — perhaps because the discussion was part of the assessment.

3 FOCUS ON A SMALL NUMBER OF KEY CONCEPTS IN EACH LESSON

The subject of computing encompasses a variety of concepts. In Australia, the Digital Technologies curriculum is underpinned by ten key concepts. It may be tempting to address several concepts in lessons to achieve a wider coverage of the curriculum, but this can lead to overloading students and to lessons lacking a clear focus.

We recommend focusing on two to three key concepts in lessons and assessments and showing students the relationship between them. For example, some of our assessments in the MEd units involve our students designing resources and units of work with a focus on just a small number of concepts. This approach allows teachers to align their teaching more directly with curriculum outcomes, and allows students to explore the concepts and the relationships between them more deeply.

taught explicitly, or an expectation of complexity that is not aligned with the time provided to do the task. Our MEd students design computing assignments and create sample solutions aligned to different levels of student achievement, which enables them to assess and evaluate how they have aligned curriculum, task, and assessment objectives. This process, while time-consuming, can give teachers valuable insights that can be used to improve the task for their students.

REFERENCES

- Guzdial, M., (2015). Learner-Centered Design of Computing Education: Research on Computing for Everyone. *Synthesis Lectures on Human-Centered Informatics*, 8(6), 1-165.
- Meyer, J.H.F., and Land, R. (2003). Threshold concepts and troublesome knowledge: linkages to ways of thinking and practising. In C. Rust (Ed.), *Improving Student Learning - Ten Years On*. Oxford, OCSLD.

“ FIND OUT WHAT INNOVATIONS AND TEACHING STRATEGIES HAVE BEEN TESTED AND CAN BE ADOPTED FOR YOUR TEACHING

4 SEEK OUT EDUCATION RESEARCH

We recommend seeking out computing education research (and education research more generally) as more becomes available with the growth of the subject, to find out what innovations and teaching strategies have been tested in research and which can be adopted for use in your own teaching.

Our MEd students have compared and discussed a variety of research, which has helped them learn what is happening in the teaching of computing outside of their context. There are sites, such as the International Journal of Computer Science Education in Schools and CSEdResearch.org, that publish and summarise research in publicly accessible formats. Organisations such as the Education Endowment Foundation in the UK and Evidence for Learning in Australia, publish summaries of research evidence and guidance for interpreting research.

5 DO THE ASSIGNMENTS, ASSESSMENTS, AND PROJECTS YOURSELF

When designing tasks or assignments for students, it's important for teachers to take the time to do the task themselves. This enables a teacher to experience the task from a different perspective and gauge whether the task and instructions have been pitched appropriately.

This process will assist with finding issues — for example, assumptions about skills that may not have been

6 KNOW AND PLAN FOR THRESHOLD CONCEPTS

Threshold concepts are the concepts in a discipline that present new and transformed ways of understanding, without which the learner cannot progress to a deeper level of knowledge (Meyer and Land, 2003). Threshold concepts are essential to mastering a discipline and provide a significant challenge for learners.

There is ongoing research to identify threshold concepts in computing, particularly in school-based contexts. What we know is that variables, abstraction, functions, and object-oriented design are among the important threshold concepts that are difficult

to learn and can also be difficult to teach. Each of these complex ideas transforms the way in which a student understands programming and computing more broadly, and once learnt, cannot be unlearned. The identification of threshold concepts has significant implications for how teachers design learning opportunities and learning materials for their students. We recommend keeping up to date with computing education research, as outlined in tip four, to help you identify these threshold concepts.

Final thoughts

Computing education is an exciting area that poses both challenges and opportunities for teachers. We hope the tips provided enable you to seek out the research literature and professional development opportunities that you need to enrich your teaching practice. (HW)



DANIEL HICKMOTT

Daniel is an associate lecturer at the University of Sydney. He completed his PhD at the University of Newcastle, Australia. His study investigated the impact of teacher professional learning programmes in computing and how primary school teachers integrate computing across different subjects (@dan_hickmott).



SUJATHA GUNJA

Sujatha is a lecturer at the University of Sydney. A secondary teacher and ex-software developer, she has taught computing to students from kindergarten to 12th grade. She has run professional learning in computing for teachers across Australia and developed learning resources in use by thousands of students (@sgunjal).

► The following three lesson plans are taken from the 'Data and information' unit of different stages of the Teach Computing Curriculum (TCC). The goal of teaching 'Data and information' is to ensure that learners can use technology to answer questions as well as collecting, analysing, and presenting data. Within the TCC we teach a unit of work in each year group, to develop a different angle of the learners' understanding of data and information.

ABOUT THE TCC



The Teach Computing Curriculum is part of England's National Centre for Computing Education and supports students aged 5-16.

Every unit of work in the TCC contains: a unit overview; a learning graph to show the progression of skills and concepts in a unit; and lesson content, including a lesson plan, slides, and formative assessment opportunities. Find them when you sign up for a free account at nccce.io/tcc.

GROUPING DATA: LABEL AND MATCH

Labels and groups are the first steps to understanding how computers allow searching and sorting of large amounts of information

This is the first lesson in a 'Grouping data' unit that focuses on assigning data (images) with different labels, to demonstrate how computers can group and present data.

Labelling objects is an accessible way to begin exploring how data can be organised, as learners of this age are still building their vocabulary and learning that objects can be both 'a pencil' and 'made of wood', depending on how you are grouping them. It also helps learners begin to make sense of the world around them while they are laying the foundations for how computing-specific technologies work and the fact that computers don't have brains — we have to give them the labels!

These activities do not focus on the use of a computer, but instead build the fundamental understanding of how computers work so that it can be built on over the course of learners' education. (HW)

AGE RANGE

5-6 years

REQUIREMENTS

- An assortment of physical objects for grouping
- Hoops or string to separate groups

ACTIVITY 1: LABELS AROUND US 10 MINUTES

Explain to pupils that they will be finding labels around the classroom. Give them time to move around the room looking for labels. These might include:

- Drawer labels (pencils, pens, paper, etc.)
- Book labels (English book, maths book, etc.)
- Room labels (Class 1, Lion class, etc.)

Discuss some of the labels that pupils have found, and ask, "Why do objects have labels?" Take some suggestions and explain that we give labels to objects to make them easy to find and discuss.

Before moving on, start making the link between the labels we have in the classroom and computers, by modelling using a search engine to find images. What do the learners think the computer will show if you type 'cat'?

Show the pupils what happens and ask, "Does the computer know that this is a cat in this image? How does it know?" Take suggestions and explain that the computer doesn't recognise that the images are cats; it is showing the images because a human gave them labels, like those that they found around the classroom.

ACTIVITY 2: MATCHING OBJECTS TO GROUPS 15 MINUTES

Show learners a group of toys that you have labelled, such as 'cars'. Question them so that they recognise that the cars are all different, but they are still cars, so they can be grouped together. Follow this up with some other objects, perhaps fish and cars, and ask learners to place the objects under the correct label. This is just like what someone did to the computer when they told it that all those images were of cats.

Set up three hoops, or specify areas on tables, with three group labels such as: 'Things we play with', 'Things we write with', and

'Things we eat'. Place a number of physical objects that fit into these groups in a pile near the hoops. Ask pupils to place the objects into the correct group, for example, "Can you find an object that goes with the 'Things we play with' label?" Allow pupils to move the objects until all the objects have a label.

Explain that by putting the objects with a label, they have made groups. This is a way of putting similar objects into groups, and it's why the computer showed us lots of different cats when the word 'cat' was searched.

OBJECTIVE

To label objects:

- ✓ I can describe objects using labels
- ✓ I can match objects to groups
- ✓ I can identify the label for a group of objects

ACTIVITY 3: LABELLING GROUPS 10 MINUTES

In the previous activities, the label for the group was given to the learners and they had to work on finding examples that could be labelled in the same way. For this activity, you should present learners with groups that have already been created and ask them to decide what label would include all of the objects.

You may choose to create your groups as a worksheet (this is what we included in the TCC); or, if you have enough objects in your classroom, you could create groups in different spots for learners to label. At this point in the unit, it's important to keep the groups clear enough that there is

one obvious label, to ensure the concept isn't confused with describing attributes of those objects. For example, if you had a group of just green cars, learners may label the group 'green' or 'cars'. We will come on to this later.

RELEVANT LINKS

TCC 'Grouping data' unit:
helloworld.cc/groupingdata

PLENARY ACTIVITY: MATCHING OBJECTS TO CATEGORIES 5 MINUTES

For the final group, show learners three objects, such as a cat, a duck, and a tree, and three groups, such as 'cat', 'tree', and 'animal'. Ask them to think, pair, share: "Which object goes with which group?" Take suggestions and discuss any differences in opinions around whether the cat should go in 'cat', 'animal', or both. Conclude that an object can

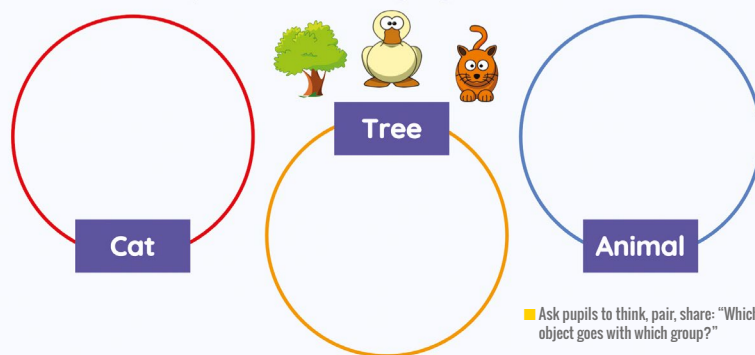
fit into more than one group and so we can give it more than one label.

Conclusion

This lesson is only the very beginning of the learning progression that will take the learners from labelling single objects to recognising the range of attributes for

a specific object and using this data to understand concepts such as 'how many' and 'more than'. The next lesson in the unit (not shown here) moves on to the use of groups to aid counting. This allows the learners to generate new data about their objects, such as how many cars there are, or how many animals.

Match the object to the category



AGE RANGE

8-9 years

REQUIREMENTS

- Paper or dry-wipe boards and pens
- No computers needed!

DATA LOGGING: ANSWERING QUESTIONS

What questions can and can't be answered using data, and how do you know if you're collecting the data you need to answer your question?

This is the first lesson in a 'Data logging' unit that focuses on what data can be collected and whether that data can answer the questions we have.

In the adult world, computers are regularly used in the collection and processing of data. As learners get older, they begin to question how devices work, asking questions such as, 'Do the traffic lights know whether or not I am here?' This is a great opportunity to explore the role of sensors in computer systems and look at what questions can be answered by collecting data

about the world around us. Data logging gives learners hands-on opportunities to experience the role of computers in automated data collection while also making connections between their learning experiences in maths and science.

These activities focus on how technology can aid the collection, analysis, and presentation of data. They look at how that data can answer questions we have about our environment, and begin to expose learners to the idea that we can store and organise data to answer even more complex questions. (HW)

OBJECTIVE

To explain that data gathered over time can be used to answer questions:

- ✓ I can choose a data set to answer a given question
- ✓ I can suggest questions that can be answered using a given data set
- ✓ I can identify data that can be gathered over time

WE CAN USE DATA TO ANSWER QUESTIONS ABOUT THE WORLD

Weather			
Day	Summary	Temperature (°C)	Rainfall (mm)
Sunday	Sunny	21	0
Monday	Partly cloudy	17	0
Tuesday	Overcast	16	2
Wednesday	Overcast	15	1
Thursday	Partly cloudy	16	0
Friday	Sunny	20	0
Saturday	Sunny	22	0

Fitness tracker	
Day	Steps
Sunday	10307
Monday	12139
Tuesday	9844
Wednesday	12015
Thursday	7053
Friday	11413
Saturday	4341

Sports				
Team	Played	Won	Lost	Drawn
Archers	7	2	2	3
Badgers	7	4	1	2
Coasters	8	3	4	1
Flyers	7	1	3	3
Ringers	8	3	4	1
Zoomers	7	5	0	2

■ Activity 1: Which table would you look at to find the warmest day?

ACTIVITY 1: DATA TABLES 10 MINUTES

Show learners some tables of data and ask them to discuss what information they are being shown. The data could be anything, but things that are familiar to learners work best, such as the weather on different days of the week, fitness tracker data, or data from a recent sports event. It would be ideal to use three different tables of data.

Once the learners have spent some time looking at the tables, they can begin to discuss which one would be the most appropriate to answer specific questions, such as, 'What was the warmest day last week?' Here, the focus of the activity is not on the answer, for example Wednesday, but on encouraging learners to recognise that the data has been organised under headings to give it meaning. They should also be able to dismiss looking for data about the weather in a table about someone's step count!

Repeat this activity a few times, asking which table the learners would use to look for the answer to specific questions, and then hide the data that is shown in the tables so that just the headings are visible. Ask the learners if they would still be able to answer the questions about which table they need to use, even without knowing what the data is. The answer should be yes. Table headings should provide a good indication of what can be found out from the table. This is important, as learners need to understand the importance of choosing appropriate headings when they begin to organise their own data into tables.

ACTIVITY 2: CLASSROOM DATA 5 MINUTES

Now show learners an empty table that you have created. Add the column headings Time, Temperature, Light, and Sound, and explain that they could collect data from the classroom at different times to show these things. Ask learners to think about questions they could answer if they had data in this table. Encouraging learners to think about what answers are possible given a limited data set helps them to consider the limitations of certain data sets, as well as ensuring they have a plausible question to investigate at a later date.

The questions that learners choose will mainly start with 'what' or 'when'. They could be questions such as, 'What was the highest temperature in the classroom today?' or 'When was the room quietest?' To aid your learners with their question creation you may want to provide them with some helpful words, such as hottest, coolest, loudest, quietest, brightest, duldest, highest, lowest.



ACTIVITY 3: COLLECTING DATA 15 MINUTES

Now that learners have an idea of the importance of table headings and the types of questions they can answer with data, it's time for them to start collecting some data of their own. We chose to collect data about vehicles on a fast-moving road, because we wanted to show how easy it is for people to make errors when they are recording data manually. In the TCC a video of the road is provided ([helloworld.cc/carvideo](https://www.helloworld.cc/carvideo)). However, if your school looks out on to a main road, learners could collect data from there.

Each learner should be asked to focus on a certain colour of car and keep a tally of how many they see. However, it's important that you first highlight that some car colours are harder to see than others, so learners

should just try their best to record what they think. After they've collected their data, they can work out the total number of cars that they saw of their colour.

Now consider the totals that the learners collected. It is highly likely that learners will have different totals. Ask them to consider why this might be. They may suggest ideas such as: maybe different people saw the colour differently; perhaps the cars moved quickly and someone miscounted; or someone may have counted the same car twice by accident. These conclusions are important, as they encourage learners to think about the kinds of error that could occur in their data.

Finally, explain to the learners that you're going to look at the questions they

can answer with their data, just as they did in the earlier activity. Questions that they can answer are things like, 'How many red cars were there?' or 'Were there more blue cars than black ones?' However, they can't answer questions such as, 'How many buses were there?' or 'How many cars had two people in?' because they didn't ask anyone to collect data on those questions. It's a good exercise to ask learners to answer these questions themselves, so that they are able to recognise that they need to think carefully about the questions they want to answer, before they collect any data. This will prepare them for later lessons when they're planning their own investigation to answer a question.

ACTIVITY 4: DATA GATHERED OVER TIME: PEOPLE 10 MINUTES

Collecting a snapshot of data as we did with the traffic can be useful, but often, the data we hear about was not collected all in the same hour, or even on the same day! Introduce learners to the census and explain that it is a way for a government to collect data about everyone in a country. Tell them how often it is carried out in your country, and that data is recorded on things like who lives in each house, how old they are, and what they do for a job. The data is collected by each household answering questions on a form. Encourage the learners to think about the fact that this data is collected over a long period of time. For example, the first census in the UK was conducted in 1801 — over 200 years ago! Can the learners think of any other data that is collected over different time periods?

It may be useful to show learners some pictures of different types of data used in different industries. For example,

health data could include thermometers, heart rate monitors, or step counters. Each of these can be used to collect data over time. The sports data example from Activity 1 might track all the goals that have been scored in a certain tournament since its inception; or weather data might allow us to know that this is the hottest summer on record. These are all examples where data is collected regularly over a period of time and then used to answer questions, such as whether a football team scored more goals this year than last year.

It's important to consider how long data has been collected for, because when learners use data loggers, they're going to have to decide how frequently they want them to collect data: does it make sense to take a temperature reading every second? What about every minute? To make these choices, they first need to decide what question they're trying to answer, and then they need to think about

■ An example of a census form

how long the total data collection time will be. Collecting data once an hour over an hour-long time period would not give you much data at all! We will pick this up further in later data logging lessons (not shown here).

RELEVANT LINKS

TCC 'Data logging' unit
helloworld.cc/datalogging

“ DATA CAN BE COLLECTED IN A FEW MINUTES, OR OVER CENTURIES, AS WITH THE CENSUS

PLENARY ACTIVITY: ANSWERING QUESTIONS 5 MINUTES

Finally, to end the lesson, we want the learners to consider whether they should use a small amount of data to answer a question. Let's take the data that it was sunny yesterday and again today. Now let's ask a question: 'Do we live in a sunny place?' With the data we have, the answer would be yes. However, it might have rained for 50 days before those two sunny ones. Getting learners to ask and answer questions based on small amounts of data helps them to see the need for a wider data set.

Conclusion

This lesson is the beginning of the learning progression taking learners from understanding the role of data, to logging data on a data logger, to analysing what questions that data can answer. The next lesson in the unit focuses on the role of sensors in collecting data. This allows learners to explore how computers can be used to measure things like light, sound, and temperature in their classrooms, and discover what factors impact these measurements.



LESSON PLAN

INTRODUCTION TO SPREADSHEETS

AGE RANGE

10-11 years

REQUIREMENTS

- Paper or dry-wipe boards and pens

Choosing the context carefully makes a spreadsheet more than just boxes on a screen

This is the first lesson in the 'Introduction to spreadsheets' unit, which focuses on how we can use spreadsheets to organise data and create new, calculated data.

Spreadsheets are powerful tools that allow people to analyse their data quickly, create calculated data, and present that data, all in just a few clicks. However, understanding when and why you should use them, and getting learners invested in the context of spreadsheets, is vital. Are you trying to keep track of what percentage you need to score on a test to get the overall grade you

want? Or tracking your scores in a video game so you can see whether you're improving over time?

These activities focus on how computing can make us more efficient at answering data-related questions by calculating data on our behalf. They show learners the merit of organising data to ensure that sorting and filtering are effective and to make answering more complex questions possible. They culminate in learners budgeting their own celebration event (not shown here), using a spreadsheet to allow them to make choices about the best way to spend their budget. (HW)

OBJECTIVE

To identify questions which can be answered using data:

- ✓ I can explain the relevance of data headings
- ✓ I can answer questions from an existing data set
- ✓ I can ask simple, relevant questions that can be answered using data



	A	B	C	D
1	Fruit	Cost	Number sold	Subtotal
2	Apples	£0.40	4	£1.60
3	Bananas	£0.50	9	£4.50
4	Kiwi	£0.80	3	£2.40
5	Grapefruit	£1.00	1	£1.00
6			Total income:	£9.50

■ Our fruit stall spreadsheet encourages students to ask questions about data

ACTIVITY 1: WHAT'S THE ANSWER? 15 MINUTES

To get learners thinking about spreadsheets, it's important that they see examples that relate to their own experiences. Start by showing them a spreadsheet with a list of names in column A and numbers in column B (no data headings!) and ask them what they think this data is showing. The point here is that they don't actually know; it's just data — words and numbers without any meaning. They might suggest it's test scores, the outcome of a game, ages, or a whole range of other things. The conclusion: this is why we use data headings when we organise data.

The more practical uses that learners can see for spreadsheets, the more they'll understand why they are used. The example spreadsheet used in the TCC was in fact about attendance data — the number of days off some (fictional) children had taken during the school year.

Now you can show a spreadsheet with a few more data headings, although still a limited data set — for example, if your school provides fruit, you could provide data on the amount of fruit bought and thrown away. Again, linking it to learners' experiences is a great way to support them in asking questions that they can answer with data.

Now show learners a more complex spreadsheet: we chose one from a school fruit stall, with the fruit name, cost, number sold, and subtotal, and five rows of data. We can ask learners to navigate it to answer simple questions. Examples might be a mixture of organisational information such as 'What is the third data heading in

this spreadsheet?' or 'How do you think the subtotal was calculated?' and contextual information such as 'How much do apples cost?' or 'Which was the most expensive fruit?' However, both types of question reinforce the role of data headings, as well as the idea that data organised into a spreadsheet can help us answer questions.



■ Sports data can help to engage learners with spreadsheets



■ Choose an inspiring context for your students, such as the cost of tech devices

ACTIVITY 2: SPREADSHEET QUESTIONS 20 MINUTES

As learners have understood that spreadsheets can be used to answer questions, you can now ask them to think about questions they could ask. By doing this, the learners are now having to consider what questions can and can't be answered given the data that is available to them. It also offers them an opportunity to explore the range of information that can be extracted from data when it is organised in this way.

This time, we introduce another different spreadsheet, but it should be similar to the one the learners explored when answering questions about the fruit. The structure is then familiar, and they should be able to use some of the questions that we asked them to get them started; however, the data should be larger in size, and the context

should be different. For this second data set, we choose to use a technology shop tracking the items that they are selling, as learners are often unaware of the prices of the devices they use, and so this context might interest them. Another merit of this approach is that the cost of some of the items is very high, so learners can see the value of the computer creating the calculated data, rather than themselves. Large numbers would make it more challenging for the learners to do the more complex calculations manually. As always, you should choose a context that inspires your learners!

When you introduce learners to this new spreadsheet, give them some time to familiarise themselves with it. Can they work out what the spreadsheet is for, who would use it, or why it was

created? All these questions get learners thinking about the merits of using a spreadsheet. It doesn't matter exactly why they think the spreadsheet was created. Is it managing stock, profit and loss, or sales trends in certain periods? It matters that the learners can see the application of spreadsheets, and data, to answer these wide range of questions.

Encourage learners to create a wide range of questions that focus on different parts of the spreadsheets. Some can be simple 'How much is x?' questions, for which the learners are just navigating the cells, but for others some simple analysis could be required, such as 'What's the most expensive item?' This will be a great reflection point when learners cover sorting and filtering later in the unit.

	A	B	C	D
1	Device	Cost	Number sold	Subtotal
2	Tablet	£50.00	2	£100.00
3	Laptop	£325.00	5	£1,625.00
4	Notebook laptop	£69.99	10	£699.99
5	Computer	£649.99	4	£2,599.96
6	Mobile phone	£49.99	1	£49.99
7	Games console	£249.00	3	£747.00
8	Games console XL	£499.99	40	£19,999.60
9	Gaming laptop	£999.00	5	£4,995.00
10	Satellite navigation device	£249.00	1	£249.00
11	Smartphone	£729.00	52	£37,908.00
12	VR computer	£399.00	0	£0.00
13	E-book reader	£118.98	1	£118.98
14	Smart speaker	£159.00	3	£477.00
15	Smart speaker with screen	£239.00	2	£478.00
16	TV	£99.99	20	£1,999.80
17	Smart TV	£799.00	35	£27,965.00
18	4K Ultra HD video camera	£129.00	2	£258.00
19				
20			Total income:	£80,270.72

■ Can learners work out what this spreadsheet might be showing?

PLENARY ACTIVITY: ANSWERING QUESTIONS 5 MINUTES

Learners can now put their questions to the test by asking their peers to use the spreadsheet to answer them. Discuss how they found the correct answer. This activity is most effective if you pick some key questions that you've seen the learners create in the previous activity, to ensure a variety of complexity in the questions that they're asking.

Conclusion

This lesson is only the very beginning of the learning progression that will take the

learners from understanding the purpose of a spreadsheet to being able to use one to organise their data and use formulas to create new data.

The next lesson in the unit progresses to learners building their own data set that looks at the distance of popular tourist attractions from their school. This will allow them to explore different number formats and judge the effectiveness of their headings, as they are using online maps to collect their own data.

RELEVANT LINKS

TCC 'Spreadsheets' unit
helloworld.cc/spreadsheets

AGE RANGE

11-13 years

REQUIREMENTS

- A computer or tablet that can run Scratch 3.0
- A webcam
- An internet connection (Teachable Machine does not work offline)

AN INTRO TO PRACTICAL MACHINE VISION

This project provides a short introduction to machine learning concepts by using Teachable Machine and Scratch to create a project that will recognise different objects

In this project, learners will use a cloud-based machine learning engine called Teachable Machine (with Scratch) to create a project that will recognise a single image class and tell them when it detects this image. We have used a hotdog mascot for the image, but you will be able to detect any image or object you like in your project.

Completing this project allows learners new to machine learning to explore the concepts of image recognition and model training. It can act as a good lead-in to a discussion about the limitations of machine learning, particularly the fact that classifier models such as this one can only recognise what they have been trained to identify. **(HW)**

OBJECTIVES

- ✓ How to create classes for a machine learning model
- ✓ How to use the Video Sensing and Teachable Machine extensions in Scratch 3.0

LEARNERS CAN EXPLORE IMAGE RECOGNITION AND MODEL TRAINING

RELEVANT LINKS

helloworld.cc/machinevision



MARK CALLEJA & PHILIP HARNEY

Mark and Philip work for the Raspberry Pi Foundation, where they create educational projects, which you can find on the foundation's website.

DIFFERENTIATION

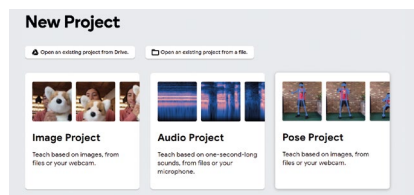
If you want to stretch your learners, you can do so in Teachable Machine by adding an additional class. Remember, you might need to make changes to your 'Not_Hotdog' class to deal with this, if you originally included the object you're now making a class for.

You can also stretch learners in Scratch, by:

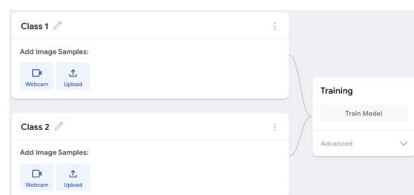
- Adding sounds to your project for each class
- Making the sprite change costume for each class
- Creating a project that requires a certain image to be shown to solve a riddle
- Making a game where you need to show a specific picture when prompted

ACTIVITY 1: CREATE YOUR MACHINE LEARNING MODEL

First go to the Teachable Machine website (helloworld.cc/teachablemachine) and choose 'Image Project' from the three options presented.

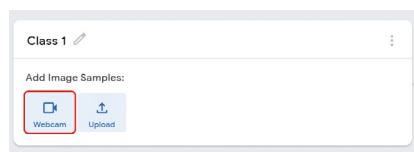


Here, you can see your model workflow.



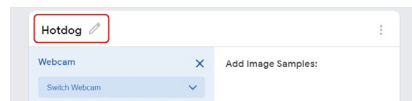
First, you will add some classes on the left-hand side. Classes are the different sets of data that we give to the Teachable Machine. Once you have created your data sets as classes, you will train the Teachable Machine to recognise similarities and differences between your images in the class. When it is trained, you will test and export your model so it can be used in Scratch.

Choose an object or picture to use as your subject. In this example, we use a printed image of a cartoon hotdog, but you could also use a toy or other image. In the box labelled 'Class 1', click the 'Webcam' button.



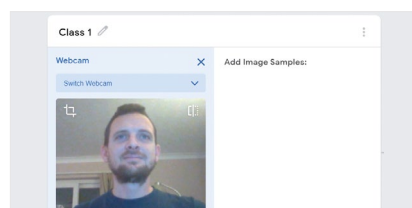
Take your object or image and hold it in front of your webcam. Hold down the 'Hold to Record' button and move your object around in front of the camera so that it captures all the sides and odd angles that you can show the camera. Try not to sit too still if you are in the shot — if you are too still, the Teachable Machine might recognise you, instead of the thing that you are holding up!

Rename the class to the thing that you are holding up in the picture. This will help when you begin working in Scratch. In this example, we have called the class 'Hotdog'.



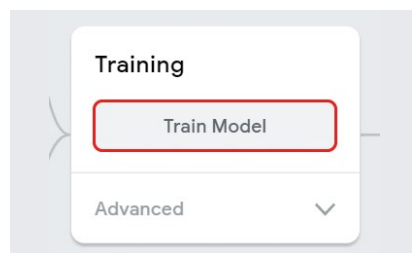
The Teachable Machine will not work with just one class. It needs a second class to compare against, so you will create that now. Change the name of your second empty class to 'not' plus the name of your object. In our example, we would call the class 'Not_Hotdog'.

Hold down the 'Hold to Record' button under your empty class and move around in front of your camera, or hold up several things that are not your object. This will teach the Teachable Machine things that are not your object and allow it to compare.



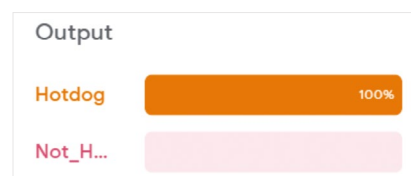
Now that you have created your two data sets, you will train the model to recognise the difference between them and tell you if it sees your object or something else.

To the right-hand side of your data sets, there is a grey button labelled 'Train Model'. Click the 'Train Model' button to begin training the Teachable Machine.



Do not close or change your browser tab while you are waiting, or all your careful training will not work!

When you have clicked the 'Train Model' button, a progress bar will appear where the button was, showing how far through the training process the Teachable Machine has progressed. Once you have trained your model, you can test that it works correctly: hold up the objects or images one at a time and watch the changing bars on the right-hand side of the screen. These bars show the model's certainty that it is looking at the object shown in one of the classes that you have made.



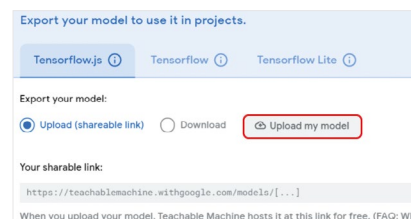
Switch between different objects and watch the model predict what it is seeing!

The next step is to export your model. This means saving the model online so you can use it in your Scratch project. To export your model, click the 'Export Model' button on the top right-hand side of your screen.



This will show you a pop-up asking you to choose a few options.

Select 'Upload (shareable link)' and click on 'Upload my model'. This will save your image recognition model to the cloud and return a URL that you can then use in Scratch. Copy the URL shown under 'Your shareable link'.



Leave this tab open, or paste your URL into a text file so that you don't lose it — you will need it in Scratch in the next step!

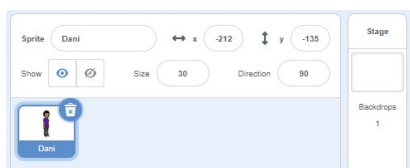
▶ ACTIVITY 2: MAKE AN IMAGE DETECTOR

In this step, you will take your exported model and make an image detector in Scratch.

Start by opening a blank machine learning Scratch project by going to rpf.io/ml-scratch.

You should see the Scratch environment that you are used to, with a few small differences. The project should contain a single sprite named Dani.

Make sure that the Dani sprite is selected before you start to code.



On the left-hand side, you will notice some new Blocks menus that do not exist in the version of Scratch that you are used to. These are special extensions (forks) created by MIT to work with some really cool new technology, and they make this fork incompatible with the version of Scratch that you usually use.

That means that any projects that you make in this fork of Scratch will not work in the other version of Scratch.

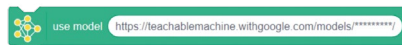
How to save your work

As this fork of Scratch is different to the version of Scratch that you are used to, you will not be able to save your work in the same way as in Scratch, and your project will be incompatible with your Scratch account. So you will need to download your project to your device, and then open it again in this specific online version of Scratch to use or edit it.



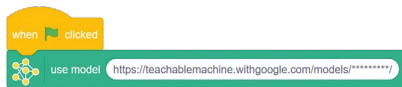
For this project, you need to open the Teachable Machine blocks menu.

Paste the URL that you copied from the Teachable Machine in the previous activity into the use model block at the top of the Teachable Machine blocks menu.



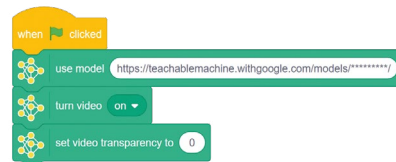
This will direct Scratch to access the model that you just created and stored on the cloud, so that it can detect your images.

Add the updated **use model** block, along with a **when green flag clicked** block, to the Code area.



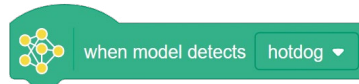
This means that when you start your program, Scratch will immediately access your model from where it is stored. This takes a little time to initiate the first time you use it, so be patient when you run your project later.

Now add a **turn video on** block to make sure that your camera can detect things, and a **set video transparency to** block so that the picture that it provides is as clear as possible.

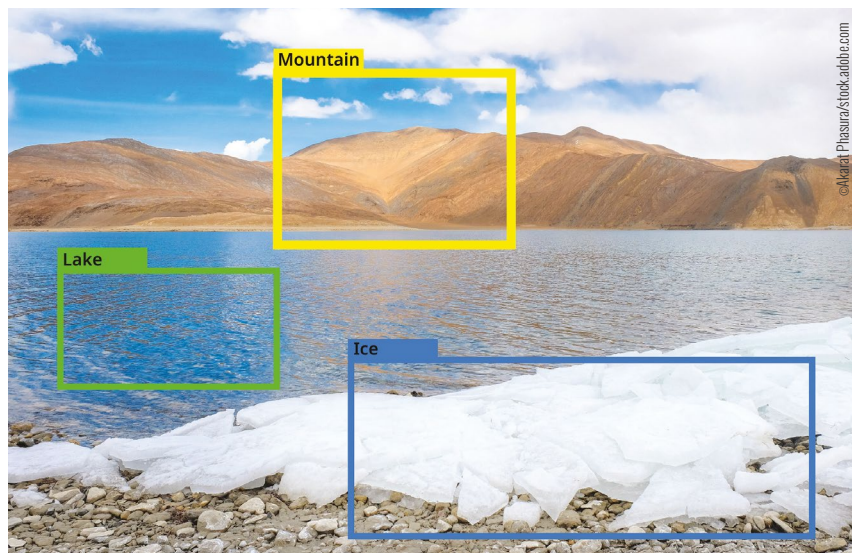
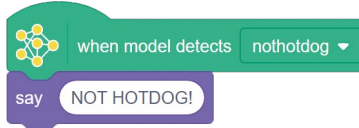
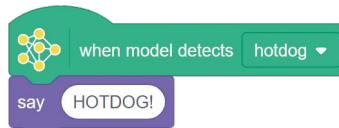


Click on the green flag to load your model and check to see if the camera is working. Now you need to create the script that will detect your two different classes, using the model that you have just linked.

Drag a **when model detects** block into the Code area and choose the name of your first class in the drop-down menu. In this example, it is 'Hotdog'.



Drag a **say** block into the Code area and add a message that will say what is detected. Repeat this step for your second class. In our example, it was 'Not_Hotdog'.





ME AND MY CLASSROOM

Bruce Ahern shares how he has made his computing classroom displays inspiring and engaging for students

At my school, students have their own Chromebooks, so we don't have a set of class desktop computers. As a result, it was important to me to give the room a real computing feel. In this article, I share some of the ways I've done this and hope that it will give you some inspiration for your own classroom.

One easy way to make the space feel like a computing classroom was to have computer parts displayed, along with an open computer, so the students could see all the parts together. Even simply opening up and showing the inside of a keyboard fascinates many of my students. The addition of LED lights was a cheap and easy way to add the wow factor and make the display stand out and look modern.

Many students are instinctively drawn to computing, but are often unsure what a computing career might look like. I have tried to create a display that shows a variety of roles in inspirational companies, to help students imagine their different career options. I also have a poster about teenagers who have done various incredible things with computing.

Computing is one of the subjects that students often want to pursue in their own time. Unfortunately, many don't know where to go to develop their skills. My classroom has an 'Upgrade myself' area, which is a bookshelf with printed guides they can select and take home. I have included micro:bits with all the elements they need for particular projects.

Seeing the past is a good way to help students understand modern computing. It's also something that frequently interests them. In my classroom, I have a section displaying a few old floppy disks, including an 8-inch floppy disk, as well as a Walkman, a cassette, an old games console, and a typewriter. These catch students' attention and are good at illustrating how fast technology is changing. Students particularly like having a go on the old typewriter.

Finally, I believe the classroom should change as the year goes on. I rotate the computing careers posters and I have a new quote about computer science displayed each half term. The current one is from Stephen Hawking: "Whether you want to uncover the secrets of the universe, or you just want to pursue a career in the 21st century, basic computer programming is an essential skill to learn." Code.org also has many great posters that can be used.

We'd love to see how you bring your computing classroom to life!
Email contact@helloworld.cc if you'd like to be featured. **(HW)**



BRUCE AHERN

Bruce is a subject leader for computing at the John F. Kennedy Catholic School in the UK.



RUNNING SUCCESSFUL COMPUTING CLUBS

Alan O'Donohoe shares practical tips for running computing clubs, with case studies from clubs around the world to help you follow in their footsteps

In your teaching, have you ever found yourself experiencing any of the following feelings:

- Frustrated that an interesting and important topic is not included in the curriculum?
- Disappointed at the amount of curriculum time timetabled for your subject?
- Worn down by the requirement to measure and report progress?
- Dispirited by teaching topics your students don't value or appreciate?
- Exhausted searching for solutions to increase participation levels?
- Worried about risking something new with a class that won't be a success?
- Regretful that enthusiastic students are being held back in lessons?
- Guilty that some students lack access to the resources that others take for granted?

If you have experienced any of these sensations, you'll perhaps appreciate how an extracurricular club offers a sandbox environment to mitigate against these issues. A computing club can be a flexible vehicle to test and try out the resources, projects, and programmes that you are not yet ready to incorporate into your regular everyday teaching. As you're reading this, you're probably thinking that a club is low down on your list of priorities right now — however, it might just be the thing that brings back that sense of joy and reward that inspired you to teach in the first place. Clubs can provide teachers with an enormous amount of fulfilment and offer freedom that is harder to orchestrate within your regular teaching.

In this guide, I provide advice and share case studies of successful examples, to help ensure that you reap the fruits of your efforts.

Tips for running a successful club

1. Put your own needs first. I've spoken to many teachers who, like me, have organised clubs for different reasons. Some teachers have apologised for having selfish motives for running a club —



but I don't think anyone running a club could be described as selfish. A club can be a massive drain on a teacher's time, energy, and availability, especially if not managed well.

2. Do it for the right reason. This is the single best piece of advice I can offer to any teacher considering running a club. No teacher should consider doing so unless they are absolutely clear on their reason and confident that the reason is compelling enough. Start by deciding exactly what your primary goal is for organising a club.

CLUBS CAN PROVIDE AN ENORMOUS AMOUNT OF FULFILMENT AND FREEDOM

Once you have decided why you are doing it, everything else will fall into place. Consider what the consequences would be if there were no club; weigh up the positives and negatives. Some teachers have asked me for advice on starting a club after a colleague suggested it to them, which makes me wonder — if it was such a great idea, why did the colleague not start the club themselves?





RESOURCES

- Virtual visit to a club at Caroline Chisholm School, Northamptonshire in the UK, recorded in 2017: helloworld.cc/chisholm
- A tour of Preston Raspberry Jam in 2019: helloworld.cc/rjam
- Amanda Haughs' virtual coding club blog post: helloworld.cc/mshaughs
- Digital Schoolhouse Tournament, engaging students to develop digital skills and soft skills: helloworld.cc/digitalschoolhouse
- British Esports Championships, competitive video gaming tournaments for students aged 12+ in the UK: helloworld.cc/BritishEsports
- National STEM Clubs, practical STEM activities to support learning in a club-style environment, for primary and secondary: helloworld.cc/STEMclub
- iDEA, a free programme that helps develop digital, enterprise, and employability skills: helloworld.cc/idea
- Code Club, a free club for young people aged 9-13 organised by teachers and volunteers: codeclub.org
- Pi Wars, an annual robotics competition for schools, clubs, and individuals: piwars.org
- Astro Pi, an annual programme giving young people the chance to run their computer programs in space: astro-pi.org
- Tim Bateup's Geek Tutorials, which students in a club can follow for GameMaker, Scratch, and Python: helloworld.cc/geektutorials
- Joanna Wakefield's resources to support a micro-bit robot club for lower secondary (you must be a CAS member to view): helloworld.cc/microbitresources
- CAS Community Discussion Forums on clubs (you must be a CAS member to view): helloworld.cc/clubdiscussion
- Learn about historical computer clubs Amateur Computer Club (helloworld.cc/amateur) and Homebrew Computer Club (helloworld.cc/homebrew)

➤ The list of questions at the start of this article provides some suggestions, but here are the most popular reasons teachers organise clubs:

- To raise the profile of the subject in school and increase interest levels
- To offer experiences and opportunities that can't easily be included in lessons
- To try out new resources and ideas before using them in lessons with large groups

"We've run a computing club aimed at pupil premium children [more disadvantaged pupils] to help boost their confidence and self-esteem. The club was open to children in all year groups and ran for six-week blocks." — Dawn Jones, digital skills officer, Education Scotland.

3. Establish a time-limited programme. One practical suggestion is to limit the club to an initial six-week programme, for example, after which there will be a planned break and a period of review. This exit strategy helps protect you from committing yourself to give up every Thursday afternoon for the next 33 weeks! It also gives you the option of either starting anew six weeks later with a fresh group of students, or resuming with the previous group. Building in breaks every few weeks provides opportunities to reflect, review, and evaluate how successful the club has been in meeting your primary goal. As teachers, we already have so many demands on our precious



time and resources that we need to be clear that anything else we volunteer for is making a positive difference. If the club is no longer meeting the intended purpose, one option is to cancel any future plans, but the review period gives you an opportunity to modify it instead, to match the planned outcome more closely.

4. Choose your audience carefully and start small. This will probably determine the success of your club far more than any other factor (other than yourself, of course!). No matter how well planned and well resourced the club may be, if the same children do not attend the club week after week, if you have different faces showing up every week, how will you be able to move the group forward?

I would go so far as to suggest that in the first instance you invite those students you think will enjoy and benefit from the club the most, before opening it up more publicly later on. The purpose of your club may have already determined who the target audience will be, but there are advantages in starting with a small, mature, enthusiastic group of learners that you already know well, before opening participation up to a wider, unknown audience. This will help ensure things go to plan, with the positive attitudes and behaviours of these highly motivated students spreading to others when you eventually open the club up.

“ DON'T TREAT THE CLUB LIKE LESSONS, WITH OBJECTIVES AND ASSESSMENTS

5. Get your students to run the club, and place more emphasis on fun and exploration than learning and achievement.

Practically, if you only have to do a small amount of work to get the club up and running, this will make it easier for you to commit to running it in the longer term. This does rely on you relinquishing the reins somewhat, but providing you have invited reliable and passionate students in the first place, it should be easier to trust them with the responsibility. Even if the students make some mistakes early on in running the club, these are great life lessons for them to learn from, providing they are encouraged to reflect on what worked well and how it could be even better. In your first club meeting you could make it clear that this is their club, and it will only succeed if they are prepared to make it succeed.

Don't fall into the mistake of treating it like a series of lessons, with learning objectives and assessments. This defeats the point of having a club in the first place. ➤



➤ *"Clubs are part and parcel of the school I worked in. As computing lead, I was asked to run a computing club in the infant school. I wish I had given less structure. I did the club almost like a series of lessons, and I should have given more time for children to explore, tinker, and unpick ready-made programmes." — Rhian Roberts, Stanbridge Primary School, UK.*

6. Follow in the footsteps of others. Find out from other schools and teachers what has worked well, and try replicating their example. There are national programmes that could provide a good basis on which to start or a template to follow. CoderDojo, Code Club, TeenTech, STEM Learning, Digital Schoolhouse, and the British Esports Association all provide support materials, structure, case studies, and content that can help reduce your workload. In theory, you could work your way through this list until you find the one that best helps you meet your primary goal. The Resources section of this guide includes links to their various sites and the case studies that follow should help give you inspiration and ideas. (HW)



ALAN O'DONOHUE

Alan has over 20 years' experience teaching and leading technology, ICT, and computing in schools in England. He runs exa.foundation, delivering professional development to engage digital makers, supporting computing teaching, and promoting the appropriate use of technology (@teknoteacher).

CASE STUDIES

Raspberry Jam. Around the time that the Raspberry Pi computer was first released in 2012, I was looking for a way for me and my students to learn more about the potential of this low-cost, single-board computer. After I announced our plans to organise monthly meetings with the aim of creating a safe space for people of all ages to meet, share, and support each other, news quickly spread around the world. I was amazed to receive messages from groups in places like Silicon Valley, Tokyo, and Melbourne who were inspired to start their own local jams. My wife suggested calling these clubs Raspberry Jam.

Nearly ten years on, we're still holding our monthly events: in 2020 we celebrated our 100th Preston Raspberry Jam. Each month our community of children, their parents, teachers, and enthusiasts share news of their projects and their interests in affordable technologies; and in that time, we've seen our community grow. Covid forced us to change our event format, and now our events continue online, with live video presentations that are recorded. Schools use recordings from our jams to support their clubs, and we've had presenters join us from clubs around the UK and beyond. Each month, girls from Essex Steamettes have presented projects they've developed, and Sean Raser, a teacher at California High School, organised students from his club to record presentations for one of our jams. Sean told me, "The students are all very excited to showcase their work!"

Amanda Haughs is a primary teacher in San Jose, California, USA. "I've run a few different Code Clubs in our district in the past several years. With the first lunchtime and after-school clubs we started, our goal was to provide a space for interested students to play, experiment, and learn more. We hoped that starting up a Code Club would provide more coding exposure to our students, and additional opportunities for them to learn.

I teamed up with a few other educators at my site to launch the after-school Code Club online in the spring, soon after our campuses closed due to Covid and we shifted to distance learning. The goal of last year's club was really about socialising around a shared passion and having a creative outlet during a stressful time.

I knew for sure that last year's virtual club was a success when I received an email from a student wanting to know when we were launching the next club! We had a great turnout last year. We opened the club up to second- to fifth-graders and promoted the club as a place to play and learn together, and prefaced that there would be no explicit teaching. Something like 40 students signed up and attended our regular weekly session!

While the 'teach yourself' model worked for many, there were a couple of students in the club who showed up with very little experience and really wanted to learn. Thinking more about different learning styles, just giving them a project manual to read wasn't working for them. Luckily, we had a few students offer to teach small groups. Moving forward, I think I would plan for both models in our club: a self-directed path, and a more directed instructional path, still led by other students, but planned in advance.

There are a lot of great resources out there for educators who want to start a club but don't have time to design their own materials. Code Club International resources have been my go-to. I like that I can find everything from teaching materials, to project guides, to student name tags, to sign-up sheets, as well as coaching on how to run a club, all in one place."



Jayne Oakley has been running a Minecraft Club at an after-school centre in Westhoughton, UK. Jayne had originally used Minecraft in her classroom teaching, but felt constrained by time in lessons. She spotted opportunities to educate children through the medium of Minecraft, so took the bold move of setting up an out-of-school club in a centre where there's a strong sense of a community and other activities on offer. Children attending the club learnt how to get the most out of the platform: programming, networking, and safety.

One challenge Jayne faced was resourcing hardware powerful enough to meet their needs. She managed to source funding for low-priced PCs and licences from the local council and a housing developer, but as the sessions were not in a listed school, she couldn't use lower-priced education licences, which significantly increased her costs.

Jayne recommends that other teachers be clear about the aims of the club and set reasonable expectations, especially with parents. She found parents' expectations varied dramatically in terms of what they wanted their children to achieve. Jayne advises, "Decide what the goals are, whether that is giving back to the community or even financial reward. I was fortunate with both. Most important was that it worked around me and my family."

Lorne Pearcey, a computing teacher and author in the West Midlands, UK, advocates that any club should have a clear purpose. She loves art, she says, but "If someone said 'come to Art Club and paint', but there was no purpose or guidance, I probably wouldn't stay for long." She believes that competitions like First Lego League and VEX Robotics are marvellous at getting groups of pupils working together to problem-solve and program their robots. She thinks the concept of a code club immediately suggests one thing, though, and might be off-putting for some.

Tim Bateup, a computing teacher on Australia's Sunshine Coast, suggests that teachers consider organising an e-sports club. Tim says, "If you're ready to be bombed with students, an e-sports club will be very popular. Sure, the children don't really learn anything academic as such, but it's a huge industry and it helps with all sorts of things, like respectful relationships and cyberbullying. This worked a treat at my school." The club, which took place twice weekly at lunchtimes, had been paused due to Covid, but has just started again as restrictions have eased in Australia.

Ben Goodwin, a computing teacher at a girls' school in Kent, UK, has organised a variety of clubs including: iDea award, Girls Who Code, BAFTA game design, Game creation with Construct 3, a Python programming club for keen students, virtual-reality game development, AWS GetIT, CyberFirst Girls, and a motor sport club.

Covid has recently impacted the running of these clubs, but previously they helped to increase uptake at GCSE level. Ben's future plans include e-sports, a streaming and video games club, and a 3D design and print club in collaboration with colleagues teaching design technology and media studies.

Shane McVeigh, a computing teacher in County Tyrone, Northern Ireland, had previously organised a FIFA e-sports tournament in his school, but really wanted to expand this to other schools. The Digital Schoolhouse programme enabled Shane's students to compete with other schools using Super Smash Bros on the Nintendo Switch. Shane says, "I found it helped build friendships throughout all year groups, which the school loved, so my school never saw any problems with running it."

Dominic Luther, a computing teacher in Sandbach, UK, found that "Code Club works well as it has a set of projects, gives students a view of a variety of different skills, and can be done fairly independently. A Key Stage 4 or 5 [aged 14-18] volunteer or two can do most of the helping necessary and manage the workload."

Nigel Hydes, a STEM Ambassador in Ipswich, UK, supported a Code Club at a local primary school. Beginners placed the highest demands on him, and he found Code Club projects a useful resource. When he demonstrated the use of turtle graphics in Trinket to plot polygons in colour, children were inspired to create patterns with overlaid polygons. Nigel found he needed to support the teacher with debugging until he encouraged the pupils to become experts at helping each other out. At the beginning, many children just wanted to sit and play games until Nigel intervened and gave stronger direction.

Graham Bridge, another STEM Ambassador, in St Albans, UK, recommends laying down ground rules: for one, playing games is forbidden. The only games allowed are the ones they write themselves. Graham also advises being bold: "We started mine with Lua on Minecraft/ComputerCraft, followed by Crumble, some Python, and finally Windows Forms with C# using SharpDevelop IDE. Some Year 5 [aged 9-10] children progressed to Unity/C# after a YouTube tutorial."

THE BEBRAS PUZZLE PAGE

Each issue, Chris Roffey shares a computational thinking challenge for your students

THE CHALLENGE: WELL-PLACED TOWERS

A tower is said to be well-placed if all the towers to the left of it are shorter and all the towers on its right are taller. Select all the well-placed towers shown.

Solution on page 95.

Further information

Placing objects in order is called sorting. The quicksort algorithm is one of the most famous and frequently used sorting methods because, on average, it tends to be very fast.

A key step of quicksort is to partition objects around a



pivot: an object is chosen as the pivot; all objects less than it are placed to its left, and all objects greater than it are placed to its right. In this problem, the towers represent a possible result of the partition algorithm where the

pivot is a well-placed tower. If the towers to the left of the pivot are further sorted, and the towers to the right of the pivot are further sorted, eventually all eleven towers will be sorted. To sort these smaller groups of

DOMAIN

Algorithms and programming

SKILLS

Algorithmic thinking, evaluation

AGE

8-14 years

DIFFICULTY RATING

Ages 8-10 hard; ages 10-12 medium; ages 12-14 easy

towers, we can use the same algorithm! When an algorithm uses itself to solve a problem, we call this recursion.

ABOUT BEBRAS

Bebbras is organised in over 55 countries and aims to get students excited about computing and computational thinking. Over the last two years, there have been a quarter of a million participants per year in our annual challenge. Our archived questions let you create your own auto-marking quizzes at any time during the year. To find out more and to register your school, head to bebras.uk.



COMPUTING KEYWORD SPOTLIGHT: RECURSION

Defining everyday techniques used by problem-solvers

Recursive algorithms are ones that repeatedly call themselves. Here are two problems that can be solved recursively.

Problem 1

Russian dolls are colourful wooden dolls that can be repeatedly opened to find smaller dolls nesting inside. The total number of dolls equals the outside one + all those inside. The next doll in contains the new outside one + all those inside. Thus, a recursive

algorithm could be written to find the total number of dolls. The algorithm must stop when there are no dolls in the one being counted. Recursive processes must know when to stop.

Problem 2

The factorial of any positive integer is defined as the product of that integer and all the integers below it. For example, 5-factorial = $5 \times 4 \times 3 \times 2 \times 1 = 120$. To find

n factorial, we might start with:

```
def fact(n):
    return n * fact(n-1)
```

However, if $n = 5$, this will try to calculate $5 \times 4 \times 3 \times 2 \times 1 \times 0 \times -1 \times -2 \times \dots$. In other words, it will never stop and, if it did, the answer would always be 0. We need to add a test to tell the recursive function when to stop:

```
def fact(n):
    if n == 1:
        return 1
    else:
        return n * fact(n-1)
```


SURVEY RESULTS

Since the last issue of Hello World was released, we have continued to send out a short monthly survey to random samples of Hello World subscribers.

Here, we are sharing some of the comments we have received. We read every comment — so if you get an email asking for your views on the magazine, please do take five minutes to let us know what you think. Or if you'd like to get in touch at any time, email contact@helloworld.cc. We are always interested in reader feedback and would love to know what you'd like to see more of.

And if you would like to write for the magazine, please do drop us an email too!

Our surveys are managed and analysed by Joshua Crossman.

Such a valuable asset, even though I teach half a world away from the UK — the material is still relevant to me and my context.

I am developing lesson materials on computational thinking and AI for secondary schools. I like to read the tips and points of view in Hello World as a manner of educating myself.

I always look forward to receiving the latest magazine. It is always inspiring and has at least one article that is just what I need for my lessons. It makes you feel part of a computing community.

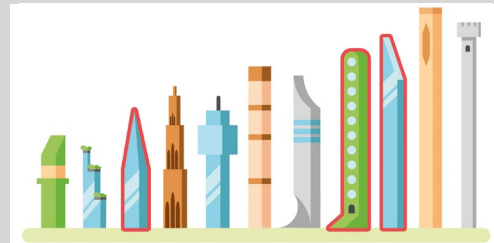
Hello World is a must-read for anyone who wants to understand the role of tech in education.

You could split content for primary and secondary, or use a colour system to differentiate.

This is definitely something we'll look at to make navigation easier! For now, we're trying to make the learning level prominent in the first paragraph and title.

BEBRAS PUZZLE SOLUTION (PAGE 94)

The three towers highlighted here are well placed. Choosing any one of the three well-placed towers, shown below, you can see that all the towers to its left are shorter than it, and all the towers to its right are taller than it.



I would like to see more 'computing for good' examples.

Check out this issue's Data science for social good Insights article and Ciira's profile in the Data science jobs article. We'll keep our eye out for other topics, but please write in if you'd like to highlight anything we can feature!

It can be hard to find accessible articles on CS topics that I can share directly with my students. I feel like Hello World can be a great resource for that.

Our articles are primarily aimed at educators, but we try to make the writing as accessible as possible, so you could share relevant articles with students. This issue, we've featured the Data science jobs article for you to share directly with your students.

I would like to see short resumes of research findings, with operational tips on how the findings can improve our classes.

In this issue's Insights articles we've included examples of how to put the findings into practice, but this is definitely something we'll look at doing more!

100 IDEAS FOR SECONDARY TEACHERS: OUTSTANDING COMPUTING LESSONS

A pocketful of inspiration for your next computing lesson

INFO BY Simon Johnson | PUBLISHER Bloomsbury | PRICE \$18.23 | ISBN 9781472984418 | URL helloworld.cc/100ideas

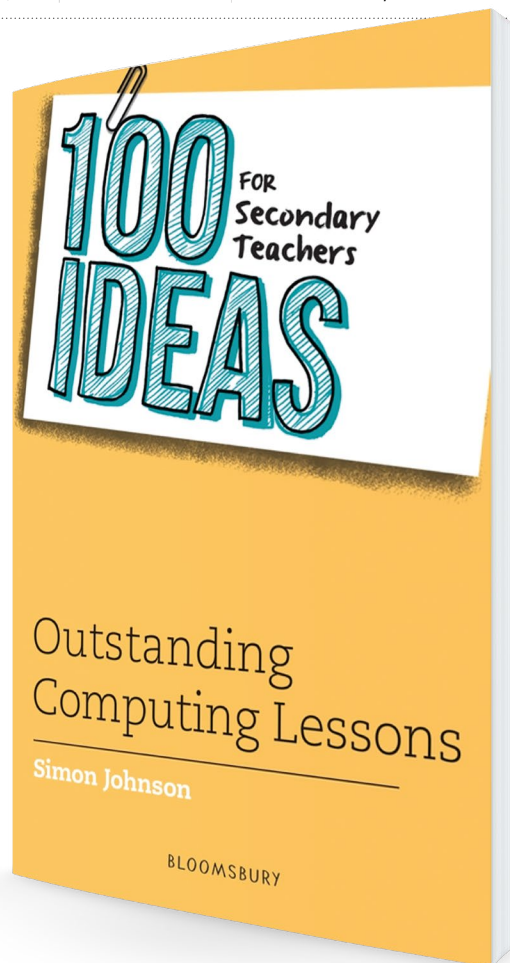
Rebecca Franks

This title is part of the popular 100 Ideas for Secondary Teachers series of books with lesson ideas for various subjects, and this book focuses on computing lessons. It is broken down into ten parts to make it easy for teachers to navigate to the topic areas that interest them.

Computing teachers will find an excellent curation of traditional ideas with strong examples of great pedagogical practice. They will also be pleasantly surprised by the uniqueness and creativity of the lesson activities.

The book begins with a focus on programming strategies. Here, teachers will find quick introductions to practices such as pair programming, Predict–Run–Investigate–Modify–Make (PRIMM), Parson’s Problems, and Use–Modify–Create. These are all strategies used in the Teach Computing Curriculum and are popular research-based approaches. The author does a fantastic job of summarising the approaches into a single, short page.

The book then moves onto more general approaches to teaching computing that involve strategies that could also be used in other curriculum subjects. There are some lovely ideas for teaching ICT and digital literacy. I particularly like the Mario Kart™ spreadsheets, but that might be because I am a Mario geek!



I can see the Exam Preparation section being helpful for supporting learners with revision; teachers are always on the lookout for ideas to help with revision, and this has a great list that includes a variety of approaches. I remember using the revision speed-dating activity with my Year 11 classes (ages 15–16),

and they found it engaging and helpful for remembering key terms.

Whether the reader is new to the subject or an experienced computing teacher, this would be a great addition to their library. It is good for curriculum planning ideas and will be useful when teachers want to try something different. **(HWF)**

WILLOW FINDS AN EGG

Encourage young readers to go on an adventure with Willow and take their first steps into coding with Scratch

INFO BY Rebecca Franks | PRICE \$12.99 | ISBN 9798715803757 | URL willowcodes.com

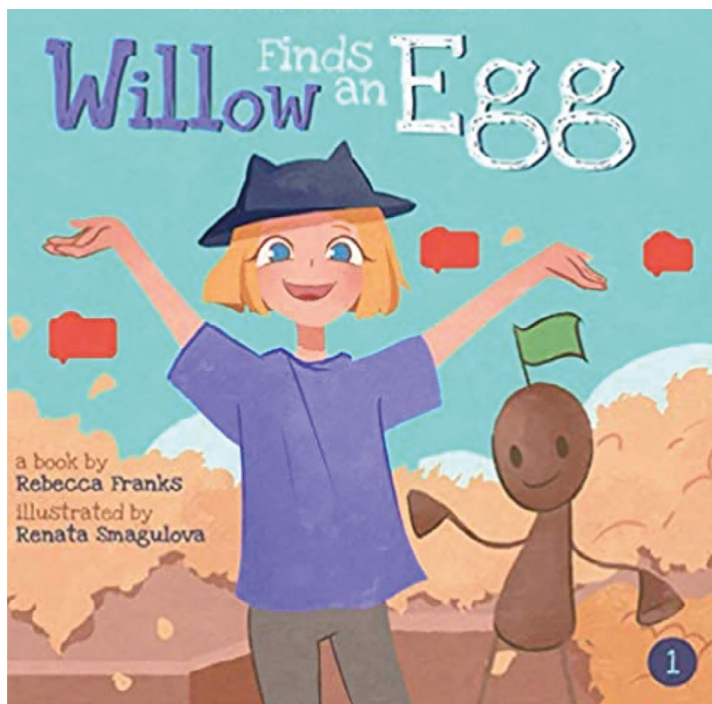
Ben Garside

Aimed at ages four to seven, *Willow Finds an Egg* is an engaging and beautifully illustrated story in which the main character takes the challenge of tidying her room and turns it into a fantasy adventure to find a lost egg. Willow is faced with challenges that the reader is encouraged to work with her to overcome. As well as introducing the reader to fundamental programming concepts such as the need for precise instructions and sequencing, many of the illustrations are in the form of Scratch blocks, gently introducing

the reader to the appearance of the blocks they will use when they come to interact with Scratch.

What really makes this book stand out is the excellently produced website that supports it and provides guided activities that the reader can complete once they have finished reading.

This is perfect for parents who want to get their little ones to take their first steps into the world of coding, but would also be a great resource in a reading corner at school. Let's hope this is the first of many Willow adventures to come! **(H/W)**



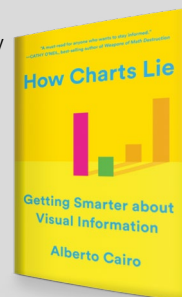
ESSENTIAL READING

Three books to help you explore data further

HOW CHARTS LIE: GETTING SMARTER ABOUT VISUAL INFORMATION

BY Alberto Cairo
PUBLISHER W. W. Norton & Company
PRICE \$16.95
ISBN 9780393358421
URL helloworld.cc/chartslie

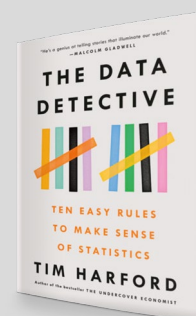
Information designer Alberto Cairo examines how charts, diagrams, and infographics can be used to mislead. He helps readers understand how to navigate our data-driven world, with examples including election result infographics and Covid-19 statistics.



THE DATA DETECTIVE: TEN EASY RULES TO MAKE SENSE OF STATISTICS

BY Tim Harford
PUBLISHER Riverhead Books
PRICE \$28
ISBN 9780593084595
URL helloworld.cc/detective

Journalist and economist Tim Harford encourages readers not to fear statistics, but instead to learn what they mean and how they can be used to improve our lives and working habits.



OBSERVE, COLLECT, DRAW! A VISUAL JOURNAL

BY Giorgia Lupi and Stefanie Posavec
PUBLISHER Princeton Architectural Press
PRICE \$25.20
ISBN 9781616897147
URL helloworld.cc/observe

Following on from the authors' 'Dear Data' project, this visual journal serves as a crash course in information design. It includes 160 pages to fill with observations about your life in data and drawings — perfect for engaging creatives with the world of numbers.



“HELLO, WORLD!”

Everything you need to know about our computing and digital making magazine for educators

Q WHAT IS HELLO WORLD?

A Hello World is a magazine for computing and digital making educators. Written by educators, for educators, the magazine is designed as a platform to help you find inspiration, share experiences, and learn from each other.

Q WHO MAKES HELLO WORLD?

A The magazine is a joint collaboration between its publisher, Raspberry Pi, and Computing at School (part of BCS, the Chartered Institute for IT).

Q WHY DID WE MAKE IT?

A There's growing momentum behind the idea of putting computing and digital making at the heart of modern education, and we feel there's a need to do more to connect with and support educators, both inside and outside the classroom.

Q WHEN IS IT AVAILABLE?

A Your 100-page magazine is available three times per year. Check out our new podcast too, to get more great Hello World content between issues. See page 8 for more details.



IT'S FREE!

Hello World is free now and forever as a Creative Commons PDF download. You can download every issue from helloworld.cc. Visit the site to see if you're entitled to a free print edition, too.

WANT TO GET INVOLVED?

There are numerous ways for you to get involved with the magazine.
Here are just a handful of ideas to get you started.

- **Give us feedback**

Help us make your magazine better –
your feedback is greatly appreciated.

- **Ask us a question**

Do you have a question you'd
like to share? We'll feature your thoughts
and ideas.

- **Tell us your story**

Have you had a success (or failure) you think the
community would benefit from hearing about?

- **Write for the magazine**

Do you have an interesting article idea? Visit
helloworld.cc/writeforus to submit your idea.

GET IN TOUCH

Want to talk? You can reach us at:
contact@helloworld.cc

FIND US ONLINE

www.helloworld.cc

 [@HelloWorld_Edu](https://twitter.com>HelloWorld_Edu)

 fb.com/HelloWorldEduMag

**NEXT
ISSUE
OUT**
**OCTOBER
2021**



helloworld.cc