

WELCOME TO THE NEW MAGAZINE FOR  
COMPUTING & DIGITAL MAKING EDUCATORS

# (Hello World)

Issue 2 | Summer Term 2017 | [helloworld.cc](http://helloworld.cc)

## BAREFOOT REACHES ONE MILLION KIDS

THE TECH LITERACY PROGRAMME HOPES  
TO REACH FIVE MILLION BY 2020

# 10 YEARS OF SCRATCH

We speak to Mitch Resnick about the visual programming  
tool that revolutionised learning for an entire generation

## DRAW WITH CODE

Get creative and get visual feedback fast

## HOST A HACK JAM

Fun and informative learning

## GUIDES AND LESSON PLANS

Fill knowledge gaps and inspire your students

## DIGITAL MAKING EXPLAINED

Everyone's talking about it, but  
what does it really mean?

## BEYOND THE CLASSROOM

Practical advice and ideas  
for education out of hours

**PLUS** GOOGLE GRANTS COME TO EUROPE • HOW DOES CODE CLUB BENEFIT CHILDREN? • EMBRACING CHAOS • CYBERSECURITY ON  
THE CURRICULUM • LEARN LOGIC GATES WITH MINECRAFT • COMPUTING & GENDER: LOST IN TRANSLATION? • INTRODUCING  
MICROPYTHON • EXPLORE PROGRAMMING WITH PHYSICAL COMPUTING • HOW ROBOTICS IS INSPIRING A GENERATION



/ UNITED KINGDOM



# Start a Code Club in Your School!

Code Club is a nationwide network of volunteers and educators who run free coding clubs for children aged 9-11.

Our aim is to inspire the next generation to get excited about computer science and digital making.



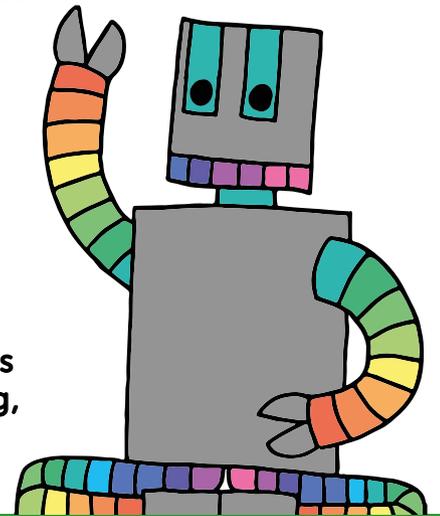
**Code Club is free**



**Code Club is flexible**



**Code Club develops skills including logical thinking, creativity, and resilience**



“Code Club has helped tremendously with the children’s confidence and engagement in coding and computing”

Caroline Harding, Year 4 teacher

We have over 5,000 clubs across the UK, teaching more than 70,000 children to code - join us!



Find out more at [codeclub.org.uk](https://codeclub.org.uk)



Code Club is part of the Raspberry Pi Foundation, Registered Charity Number 1129409

## HELLO AGAIN!

**T**hanks so much for the hugely positive feedback on our launch issue of Hello World. January's edition was a great success, with well over 12,000 downloads and so much demand for the free print edition that we had to reprint. So, on then to that tricky second album...

May sees the tenth birthday celebrations for Scratch, MIT's phenomenally successful open-source, block-based programming language. We're delighted to lead this edition with an exclusive interview with Scratch's founder, Mitch Resnick, Director of the Lifelong Kindergarten Group at MIT Media Lab.

Scratch has played a pivotal role in the CS for All movement: building programs, rather than writing them, means learners can focus on expressing ideas as code rather than



**SUBSCRIBE IN PRINT FOR FREE**  
TURN TO PAGE 28

mastering formal syntax. While the building blocks matter, it's Scratch's share button, and the vibrant online worldwide community of Scratchers, that make it such a success. Young people who take Scratch beyond the simple animations or games we might teach in school do so through connecting to, and learning from, their fellow Scratchers.

Scratch encourages a culture of sharing, and learning from one another. The Raspberry Pi Foundation and Computing At School want to do that with Hello World, too: we've some great new authors writing for this edition, and we'd love to have more! Please get in touch ([miles@helloworld.cc](mailto:miles@helloworld.cc)) if you've an idea for a story, a lesson plan, or just some news!

Miles Berry  
Contributing Editor



## FEATURED THIS ISSUE



**NICHOLAS PROVENZANO**  
RASPBERRY PI CERTIFIED EDUCATOR

Nicholas Provenzano is a high school language arts teacher, author, consultant, and a nerdy maker. He loves sharing his nerdy projects on Twitter and Instagram @TheNerdyTeacher.



**NICHOLAS TOLLERVEY**  
FREELANCE SOFTWARE DEVELOPER

Nicholas Tollervey is a classically trained musician, philosophy graduate, teacher, writer and software developer. He's just like this biography: concise, honest and full of useful information.



**LAURA SACH**  
CONTENT AND CURRICULUM MANAGER, THE RASPBERRY PI FOUNDATION

Laura Sach is Content and Curriculum Manager at the Raspberry Pi Foundation. She is an experienced ex-teacher, author, and speaker.

## EDITORIAL

**Managing Editor**  
Russell Barnes  
[russell@helloworld.cc](mailto:russell@helloworld.cc)

**Contributing Editor**  
Miles Berry  
[miles@helloworld.cc](mailto:miles@helloworld.cc)

**Sub Editor**  
Rachel Churcher

## DESIGN

**Critical Media**  
[criticalmedia.co.uk](http://criticalmedia.co.uk)

**Head of Design**  
Dougal Matthews

**Designers**  
Lee Allen, Mike Kay, and  
Daiva Bumelyte

**Illustrator:**  
Sam Alder

**Cover photography:**  
Tyler Olson/Shutterstock.com

## CONTRIBUTORS

Lucy Hattersley, Phil King, John Stout,  
and Rob Zwetsloot

Sponsored by



Hello World is a joint collaboration:



Raspberry Pi



**COMPUTING AT SCHOOL**  
EDUCATE • ENGAGE • ENCOURAGE  
Part of BCS, The Chartered Institute for IT

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 Raspberry Pi (Trading) Ltd., 30 Station Road, Cambridge, CB1 2JH. 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

COVER  
FEATURE

22



## SCRATCH: THE FIRST TEN YEARS

Hello World takes a look back at ten years of the educational programming language and asks 'What next?'

16



## DIGITAL MAKING

Everything you need to know about mixing technical and creative skills

09



## CODE CLUB GROWS UP

A new age range for Code Club!

## NEWS AND FEATURES

06 NEWS

Computing education news

14 ROBOTS IN THE CLASSROOM

Teaching and learning with robots

16 #INSIGHTS

Digital making in education

19 COMPUTING AND GENDER

Making computer science relevant for everyone

21 PHYSICAL COMPUTING

Bee-Bots and robot arms

22 HAPPY BIRTHDAY TO SCRATCH!

In conversation with Mitchel Resnick from the MIT Media Lab

30 DIGITAL SAFEGUARDING

Keeping children safe in the digital world

32 MICROPYTHON

A language designed for the Internet of Things

34 BREAKING THE CAESAR CIPHER

Simple cryptographical analysis

58 SNAP!

Block-based functional programming

60 PATTERN RECOGNITION

Using Haskell and Cuisenaire rods to teach algebra before arithmetic

62 COMMON MISTAKES

Understanding novice coders

64 BLOCKS TO TEXT

An innovative way to transition from block-based to text-based languages

68 DIGITAL LEADERS

Empowering students to lead

69 STUDYING WITH HARVARD

Online computer science CPD

70 CREATIVE BINARY

Teaching binary with music and pictures

77 SCRATCHMATHS

Cross-curricular teaching with Scratch

78 BROWSER-BASED IDE REVIEW

Integrated development environments for education

82 BOOK REVIEWS

Computer science books for teachers and children

86 INFORMATION STRUCTURES

Arrays, stacks, records, and trees

98 COURSEWORK OR CLASSWORK

Alternative approaches to practical assessment in CS



## YOUNG CODERS

From high school student to digital entrepreneur with Apps for Good

94

14

## HANDS-ON ROBOTICS

Robots in the classroom

32

## MICROPYTHON

Python for the Internet of Things

## OUTSIDE THE CLASSROOM

17

### BUILDING CONFIDENCE

Self-efficacy and the Hour of Code

20

### HACKATHON INSIDER

How to run your hackathon

35

### CHAOS AND CODING

Balancing Code Club activities

36

### CODE CLUB EVALUATION

An independent look at Code Club

38

### GIRLS WHO CODE

Promoting inclusion in CS

71

### SCRATCH AT HOME

Learning through play

72

### AFTER-SCHOOL CODING

Running a programming club

74

### CLUBBING TOGETHER

Code Club progress report

76

### CODING IN MINECRAFT

Exploring Code Kingdoms

84

### HACK JAMS

Learning together

88

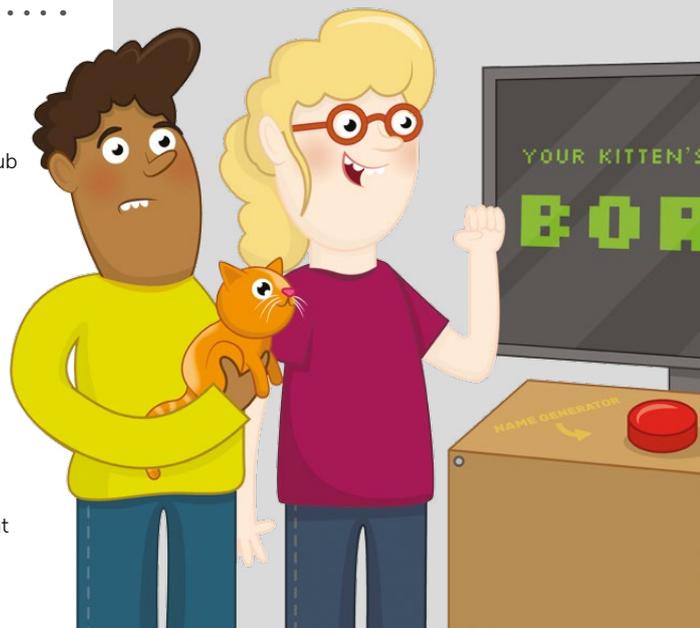
### EVENT GUIDEBOOK

Running a community event

94

### YOUNG CODERS

Digital entrepreneurs



# LEARNING

## TUTORIALS & LESSON PLANS

42

### PYTHON GRAPHICAL INTERFACE

Add a GUI to your Python program using guizero - and kittens!

44

### LOGIC GATES EXPLAINED

NAND gates, truth tables, and simulating logic gates in Python

46

### MINECRAFT LOGIC

Exploring logic gates with programmable Minecraft blocks

48

### UNPLUGGED AI

Devise a written set of rules for a noughts and crosses-playing computer

50

### ONCE UPON A TIME

Use variables in Scratch to tell a story that changes for every user

52

### DRAWN TO PYTHON

Create simple on-screen shapes by coding with PyGame

# BAREFOOT COMPUTING REACHES A MILLION KIDS

More than one million UK schoolchildren reached by tech literacy programme



**M**ore than a million UK primary school children have been reached by Barefoot Computing, a new programme to improve tech literacy skills for the young.

Barefoot Computing provides training and classroom resources for primary school teachers. It hopes to reach more than five million UK children by 2020.

Run by BT and BCS, the Barefoot Computing Project produces free teaching resources and volunteer-led training workshops for primary school teachers.

Liz Williams, BT's Director of Tech Literacy, says: "BT is delighted Barefoot has reached this major milestone, because we know that the next generation's life chances will be improved by being able to thrive in the digital world."

There is a pressing need for tech training at the primary school level.

■ Liz Williams from BT and Bill Mitchell from BCS celebrate reaching a million kids



## BAREFOOT RESOURCES

Barefoot gives primary school teachers free resources and training workshops on using the key concepts that underpin tech literacy. The resources cover computational thinking concepts, such as algorithms and abstraction.

Teachers can book workshops run by a Barefoot expert volunteer. They guide teachers through using the Barefoot resources to teach pupils computer science.

To find out more, access the resources, or book a workshop, visit [barefootcas.org.uk](http://barefootcas.org.uk).

You can also schedule a workshop at [enquiries@barefootcas.org.uk](mailto:enquiries@barefootcas.org.uk).

English Cymraeg  
Search... →  
Login Register

Home About Resources Workshops Contact The Barefoot Team

### Supporting teachers with computer science

- Build your confidence in teaching computer science
- Access free resources, created and recommended by teachers
- Receive free in-school CPD workshops from Barefoot Volunteers
- Empower young people with the skills needed for success in the digital world

Download free resources Book a free workshop Register as a volunteer

### Welcome to The Barefoot Programme

Today's teachers are key to the next generation's success. Barefoot supports primary educators with the confidence, knowledge, skills and resources to teach computer science. Resources aligned to the curricular for all UK nations. This includes FREE high-quality lesson plans and local

Research from Ipsos MORI and BT found that, while 75 percent of teachers think tech literacy is as important as reading and writing, only 25 percent strongly agreed that they felt able to 'prepare pupils for a digital world'.

Barefoot was created in 2014 to address a need for computer science training for teachers.

Since its launch, more than 130,000 resources have been downloaded, and more than 35,000 teachers have used them.

"A lot of people don't realise that computational thinking concepts, like logic, sequencing, abstraction, and debugging, now underpin much of what we do in our

daily lives," says Dr Bill Mitchell, Director of Education at the British Computer Society.

"Giving teachers the confidence to teach these concepts sets young people up to

computing as a skill. That it's not just about the hardware and how we incorporate that, but that it is also about gaining computational skills, using a simple pencil

## OUR BAREFOOT VOLUNTEER HELPED US TO VIEW COMPUTING AS A SKILL

successfully navigate a whole host of real-world challenges."

Mitchell, a teacher at Deansbrook Junior School in north-west London, says: "Our Barefoot volunteer helped us to view

and paper and the 'unplugged' thinking associated with that."

### Barefoot Wales

The Barefoot Computing project in Wales was recently launched by Cabinet Secretary for Education, Kirsty Williams.

Every primary school teacher in Wales will be contacted by the Barefoot team to highlight the new resources.

Williams says she was "delighted" to launch the free-to-use, creative resources, which "perfectly illustrate how digital skills can be integrated into the Welsh curriculum."

She adds, "Since becoming Education Secretary, one of my key priorities has been to raise the aspirations for all our children and young people, broadening horizons and developing ambition so that all can achieve."

Pupils and practitioners in Wales can access the bilingual materials at [hwb.wales.gov.uk](http://hwb.wales.gov.uk). [\(HW\)](#)



# GOOGLE EDUCATION GRANTS COME TO EUROPE & CANADA

Free access to artificial intelligence for professors and students

**G**oogle is expanding its Education Grants for Computer Science to Europe, Canada, and several other countries.

The grant enables computer science departments to access artificial intelligence technology. This technology can be used by students and professors in their projects.

“We’re excited to extend this program to 30 new countries across continental Europe, the UK, Israel, Switzerland, and Canada,” writes Elizabeth Sproat, Head of Education, EMEA.

The grants are available to university professors who teach computer science.

Google Cloud Platform is the same infrastructure that Google uses in-house to provide intelligent services, such as the Google Assistant and image recognition in Google Photos.

Using the Google Cloud Platform, students have access to machine learning technologies such as the Cloud Natural Language API or Cloud Vision API.

their projects into the Google service using an API (application programming interface). They then pay a small amount each time their program accesses the service.

## GRANTS ARE AVAILABLE TO UNIVERSITY PROFESSORS WHO TEACH COMPUTER SCIENCE

Students use these to build projects that use the latest advances in artificial intelligence.

This software requires incredibly fast computers to run, so it runs on Google’s server system. Developers typically plug

The Google Cloud Platform website ([cloud.google.com](https://cloud.google.com)) contains plenty of information on the kinds of services that students can access. They cover a broad range of video and image intelligence, speech recognition, and machine learning tools. Users can also obtain access to more traditional cloud services such as virtual networking, big data tools, and cloud storage.

According to Google, professors will receive a coupon code which covers approximately \$50 (US dollars) per student and \$100 per staff member. The coupon provides roughly two months of active use.

“We look forward to seeing the new ways professors and students will use their GCP Education Grants,” writes Google’s Elizabeth Sproat. “We’ll share stories about cool projects on our blog and our social channels.”

A full list of countries (and more application details) can be found on the Google Cloud Platform Education Grant application form: [helloworld.cc/2pOjBYK](https://helloworld.cc/2pOjBYK). [\(Hw\)](#)

■ The new grant enables professors and students to access to Google’s artificial intelligence software



# CODE CLUB EXPANDS

The popular computing club opens its doors to older children

**C**ode Club is extending its age range. From September, the popular club for computing and digital making will provide resources for students aged from 9 to 13 years old.

"In the last 18 months we have more than doubled the number of Code Clubs," said CEO of the Raspberry Pi Foundation, Philip Colligan. "There are now more than 10,000 active clubs all over the world."

That means that Code Club is inspiring more than 85,000 young people in the UK each week. While the focus of Code Club has been 9-11 year olds, it turns out that schools often use the free Code Club resources for older children too.

"We know that there are already plenty of senior schools that use our current materials to run Code Clubs," says Maria Quevedo, Director of Code Club UK. "We also know there is a demand from young people to gain more experience of digital making in schools, but there aren't enough opportunities for them to do so. We want to try and fill that gap"

Code Club is now running a pilot programme with 50 UK secondary schools. "Over the next few months we will be working closely with them to find out the best ways to make Code Club work for older children," said Quevedo.



■ All Code Club educational resources are developed by qualified and experienced educators

Colligan adds: "As part of the change, from September 2017 we will be expanding our curriculum and free educational resources to cater for older children and more experienced coders. As with all of our educational

resources, they will be created by qualified and experienced educators, and focused on helping young people build a wide range of skills and competencies, including team-work, problem-solving, and creativity. "

**“ THERE IS A DEMAND FROM YOUNG PEOPLE TO GAIN MORE EXPERIENCE OF DIGITAL MAKING IN SCHOOLS**



■ From September, free Code Club resources will be available for 9- to 13-year-olds

Code Club is a global movement and is also working with regional partners to make sure that Code Clubs are available to 9- to 13-year-olds in every community in the world. "We will also be accelerating the work to translate Code Club materials into even more languages," says Colligan.

If you know a school, youth club, library, or other venue that could host a Code Club for young people aged 9-13, then encourage them to get involved. Visit [codeclub.org.uk](http://codeclub.org.uk) for more information. **(IHW)**



■ Young students taking part in the CyberFirst Girls competition

# CYBERSECURITY ON THE CURRICULUM

More money for cybersecurity education in the UK and USA

**C**ybersecurity is making news headlines across political and private spheres. In response, teachers across the UK and USA are being asked to step up cybersecurity teaching.

In the UK, the Department for Culture, Media, and Sport recently

announced £20m to fund the teaching of cybersecurity to students in the UK.

Digital and Culture Minister Matt Hancock says: "This forward-thinking programme will see thousands of the best and brightest young minds given the opportunity to learn cutting-edge cybersecurity skills alongside their secondary school studies."

In the USA, Education Week reports that the National Governors Association, eight federal agencies, and a national commission are supporting an initiative to improve cybersecurity education across the country ([helloworld.cc/2qIDXjP](http://helloworld.cc/2qIDXjP)).

President Donald Trump's administration is set to release a delayed cybersecurity Executive Order. A draft of the order will compel government agencies to implement a cybersecurity framework developed by the National Institute of Standards and Technology.

"Cybersecurity is at the centre of a lot of discussions right now," Stephen Parker, the Legislative Director for Education and Workforce Issues at the National Governors Association, told Education Week ([edweek.org](http://edweek.org)).

Managing to create a curriculum that keeps up with the rapidly changing world of security technology isn't easy. In addition, there is some debate about whether schools should prioritise



■ Cryptoy, developed by GCHQ, is an app designed to help young students learn cybersecurity



national security over work-related computing concerns.

“There’s an absolute hunger to get this kind of content into the classroom,” says Bert Steele, a consultant with the non-profit Cyber Innovation Center ([cyberinnovationcenter.org](http://cyberinnovationcenter.org)). “Everybody realises how relevant it is in today’s society.”

messages that they can share with their friends.” An iPad version of the app is said to be in the pipeline for release this summer.

More recently, GCHQ has been supporting STEM efforts at public events, including the Edinburgh International Science Festival 2017.

“ THERE’S AN ABSOLUTE HUNGER TO GET THIS KIND OF CONTENT INTO THE CLASSROOM

British intelligence and security organisation GCHQ has produced a cybersecurity Guide for Schools ([helloworld.cc/2qMsQWo](http://helloworld.cc/2qMsQWo)).

GCHQ also provides an Android app for students and teachers called Cryptoy ([helloworld.cc/2pCkQHK](http://helloworld.cc/2pCkQHK)). The government agency explains: “The app teaches secondary school students about the mysterious world of cryptography, helping them to understand ciphers and keys, and enabling them to create encrypted

Says GCHQ: “Our ‘Commander’ game, designed to test recall of a light sequence, brought out the competitive spirit in many, and our special award goes to young Lily from Edinburgh who managed to score an amazing 43, the highest score ever recorded on this game.”

With renewed interest in cybersecurity, and a new Executive Order compelling agencies to action, it seems that cybersecurity in the classroom is very much back in style. (HW)

## CYBERFIRST GIRLS

More than 8,000 young students took part in the CyberFirst Girls competition. The team from Lancaster Girls Grammar School were judged the winners.

Organised by the National Cyber Security Centre, a part of GCHQ, CyberFirst Girls aimed to raise awareness of careers in cybersecurity amongst girls.

Only ten percent of the global cybersecurity workforce is female.

Alison Whitney, deputy director for digital services at the National Cyber Security Centre, comments: “All of the girls were very worthy finalists: the standard of work was incredibly high, and we were very impressed with their work. Having worked in cybersecurity for over a decade, I would recommend working in cybersecurity to any young woman hoping to make a positive impact on the world. Cybersecurity is increasingly important to help people live and work online, and we hope CyberFirst Girls will help young women develop skills that could lead to a dynamic and rewarding career.”

■ Dr Hannah Dee (left) with the BCSWomen Lovelace Colloquium



# BCSWOMEN LOVELACE COLLOQUIUM

UK event encouraging more female students in tech has its best year ever

**T**he BCSWomen Lovelace Colloquium is the UK's leading computing conference dedicated to encouraging more female students to enter the world of computing.

"This was the tenth BCSWomen Lovelace Colloquium," says Dr Hannah Dee, Senior Lecturer in Computer Science at Aberystwyth University.

About 200 attendees came from as far as Brighton and Aberdeen to present their work in four contests.

"I gave my first ever research paper when I was a research student," explains Dr Dee.

"About halfway through, I looked up and realised I was the only woman in the room. It was just a little bit disconcerting. It's not unpleasant; it's just a little bit weird."

She continues: "There is plenty of evidence now that having diverse opinions feeding into the decision-making process leads to better decisions. The McKinsey report *Women Matter* shows that businesses with diverse boards perform better, and this does not just relate to gender.

"The problem with computing is it is stale, male, and pale – we are an industry

where at the technical end we are looking at maybe 15 percent of the workforce being female. The reason Google, Amazon, and many other top employers are joining us at the BCSWomen Lovelace Colloquium is that they want to recruit people: they want to have bright women in their teams."

To illustrate the point, Dr Dee points to a recent health-tracking app developed by a leading technology company which did not have the ability to track women's periods.

"It turned out there wasn't a woman on their development team. That is actually a pretty major omission, and these kinds of issues are happening all the time."

The BCSWomen Lovelace Colloquium moves around the country so that different universities and groups of students get to benefit.

"Thanks to sponsorship, we pay the travel and accommodation, if necessary, for all of our poster contest finalists," says Dr Dee.

Next year's BCSWomen Lovelace Colloquium is planned for 28 March 2018 at the University of Sheffield, supported by Sheffield Hallam. **(HW)**



■ Female students were given advice and support on entering the technology industry



■ Vegard, aged 12, from Norway, built a plant monitor

# MICRO:BIT MOTHER AND CARERS WINNERS

Kids challenged to make Mother's Day special with technology

**T**he Micro:bit Educational Foundation has announced the winners of its 'Mothers and Carers' challenge.

The challenge was to use a micro:bit to help mothers and carers around the world have a better, easier day.

Fans of the micro:bit responded with an "amazing set of ideas and devices,"

says Howard Baker, Chief Editor at the Micro:bit Foundation.

Baker remarks, "It was an extremely hard job to choose the eventual prize winners. A mother or carer receiving any one of your inventions would be suitably pleased and happy."

The winning entry came from Joseph, 15, who built a package of useful micro:bit

utilities. His apps include an alarm clock, teeth-brushing alarm, door-opening alarm, and key finder.

Lumina, a runner-up, coded a recipe device displaying the ingredients for pancakes, which included a stopwatch and step indicator. Another runner-up, Zak, built a Mother's Day card that displayed random messages on a button push. **(HW)**

## MICRO:BIT GIVE SHOP

The Micro:bit Educational Foundation has also launched a Give Shop ([helloworld.cc/2p51mlb](http://helloworld.cc/2p51mlb)). The store is "a way you can help change the lives of disadvantaged children around the world by helping them learn to code" says the organisation.

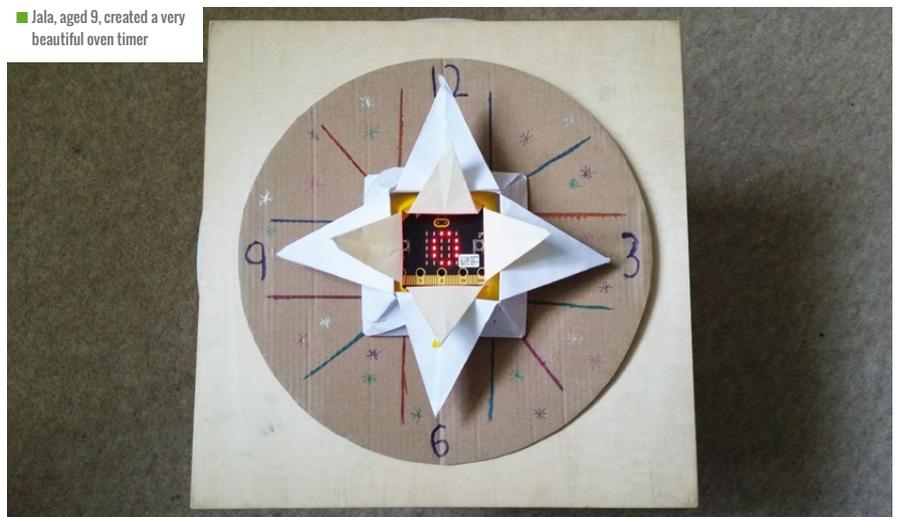
For every micro:bit you buy from the Give Shop, the Foundation will donate and ship a second micro:bit free to a school or youth program.

A single micro:bit Go! pack costs £19.95, while a pack of ten (with ten donated) costs £199.



## KIDS CHALLENGED TO MAKE MOTHER'S DAY SPECIAL WITH TECHNOLOGY

■ Jala, aged 9, created a very beautiful oven timer





**CARRIE ANNE PHILBIN** DIRECTOR OF EDUCATION AT THE RASPBERRY PI FOUNDATION

# THESE ARE THE DROIDS YOU'RE LOOKING FOR

From artificial intelligence to the Mars Rover, robotics is inspiring a generation to solve problems and manufacture a bright future

**“O**ne of my students has asked if they can use an old roller skate for a robot chassis!” Primary school teacher Bill Harvey is showing me his biscuit tin Arduino bot. “The answer? A resounding ‘yes!’”. Bill trained to be a teacher through the Troops to Teachers program, designed to train anyone who has served in the British Armed Services to become qualified teacher. During his training, Bill started a Robotics after-school club with his 9- to 11-year-old students. Each week they would build a new part of their design, recycling old materials where possible. At the end of term, the students paraded their work in the school assembly. Bill tells me “I was always only one week ahead of the children. I’m learning, and so are they.”

Until I met Bill, I had a very uneasy relationship with robotics. I wasn’t convinced that it was a good way to capture students’ imaginations. The robot arm accessory in particular caused me to wince when anyone suggested it. “Get a robot

arm to play chess”, they would say. “Why?” I’d reply. Blank expressions would follow.

For at least half a century, robots have excited and tormented our imaginations. Some of my favourite fictional characters are robots: R2D2, KITT, WALL-E, Data, Marvin the Paranoid Android, and K-2SO. Literature and film are full of fun and happy examples, as well as much darker visions of what could happen if the machines become a bit too clever for our liking: we’ve seen robots thrown out of airlocks, retired, or terminated. As robotics develops as a discipline and an industry, will it also inspire a generation to unlock robots’ potential for good?

## First steps

When robots are used in education, they are often more sophisticated versions of the Bee-Bots or turtles that children encounter in their early school years. Used to teach sequencing, algorithms, and debugging, they are a tangible way to encourage computational thinking. With the addition of physical computing, students can take their interaction to another level by building their own robots using devices like Crumble, Arduino, or Raspberry Pi.

Raspberry Pi Certified Educator Nic Hughes introduced me to Crumble bots at a conference. Redfern’s Crumble board is inexpensive and easily programmed using a block-based interface. It has been designed with young children in mind, connecting easily to motors using crocodile clip cables. Students typically start with simple forwards and backwards motion, before facing the challenge of figuring out how to make their robots turn left or right. Design skills are used to



■ Carrie Anne’s Crumble bot

build a stable and useful chassis. Of course, anything can be a chassis: my Pringles tube Crumble bot is just one example.

One of the great benefits of robotics is that it naturally lends itself to group projects, and to iterative development. Start with a simple robot that follows remotely issued commands, then add support for line following, or a display for greater interaction; support for solving mazes, for interpreting camera input, for co-ordinating with other similar robots: the list is endless. Think a little creatively, and it's clear that there's a whole world of robotics beyond simple movement.

There are plenty of robotics problems to solve. When NASA engineers were designing the Curiosity Mars Rover, they had to make one robot that could move on unfamiliar terrain, charge itself, beam images across space, be lightweight but super-strong, and not explode before it left our atmosphere! In war zones, robots are used to defuse bombs, particularly land mines. In very extreme environments they are used for search and rescue operations. I've worked with groups of learners at hackathons to build earthquake- or flood-detecting machines, motivated by the idea of using their skills to help others.

We're also part of the automation era, where machines are learning to do our jobs. They can drive cars, check out our groceries, and even perform surgical procedures. One of my favourite robotics projects made by school children was an automated waste robot that could detect whether a piece of waste was recycling or food, and then sort it into the correct



## NATIONAL EVENTS

National events such as the UK Robotics Week ([roboticsweek.uk](http://roboticsweek.uk)) and the US National Robotics Week ([nationalroboticsweek.org](http://nationalroboticsweek.org)) (RoboWeek for short) celebrate robotic innovation, educate the public and inspire the next generation of scientists and engineers. They often provide lots of educational resources to support teachers. They are a great opportunity to showcase student work, or to compete in challenges to create robots that improve social care, or help to understand a remote environment.

“ I've learned that robotics can overlap with a huge amount of computer science, and can be lots of fun.

bin in the school canteen. There are lots of opportunities in education to build machines to make our lives better, but what about the risks of mass unemployment? How will this type of development affect our future economy? This is a great discussion to have with students.

### Artificial intelligence

For many students, a robot that can show some degree of personality and demonstrate an emotion might be substantially more compelling. Who wouldn't want to have a robot pet? No mess, no vet bills, and it will never tire of playing fetch. What about a classroom full of robots interacting in interesting ways? By thinking about personality, students will be venturing into the world of artificial intelligence, or AI.

The Turing test is designed to detect programs that attempt to trick us into thinking they are really human, but what about AI that specialises in outwitting human opponents at a particular skill or game? In 2016, an artificial intelligence developed by Google, called AlphaGo, was victorious in beating human intelligence at Go, one of the

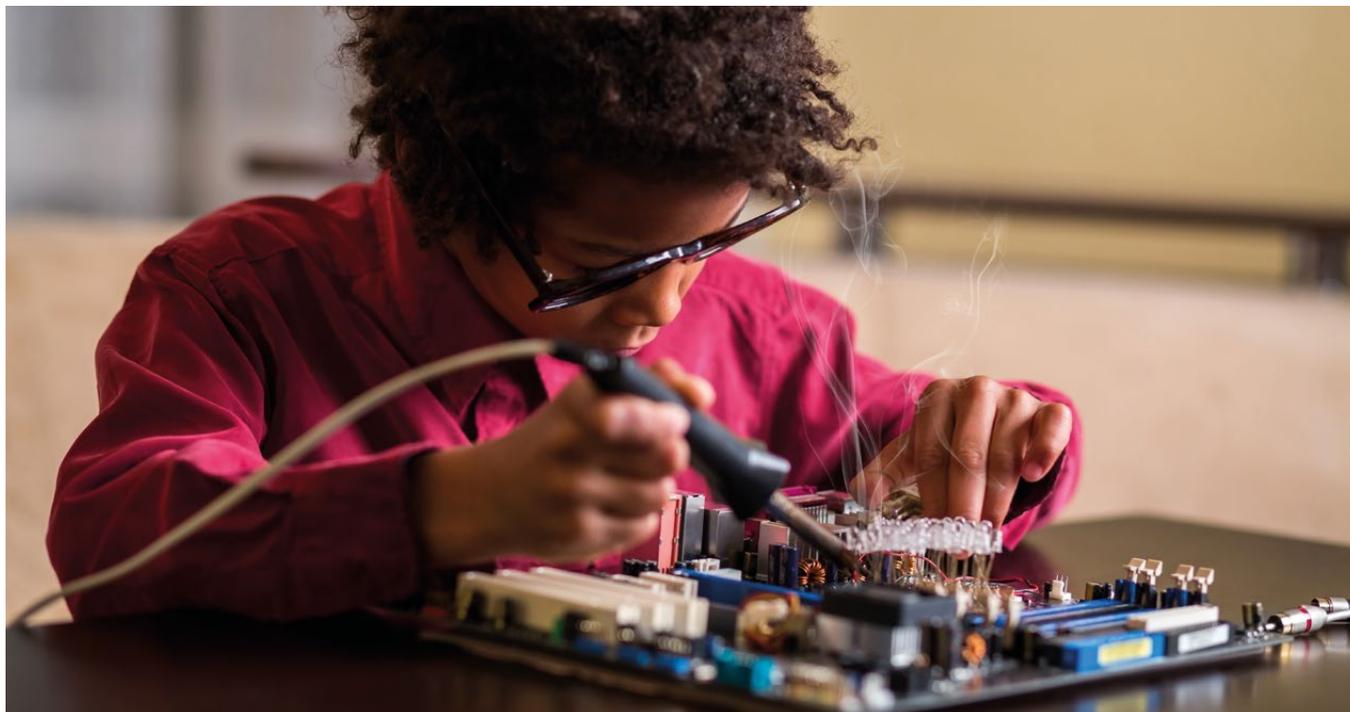
most challenging board games ever invented. It reminded me of my teenage years, following Garry Kasparov's battle against IBM's Deep Blue, which he ultimately lost. It blew my mind that a computer running a program could outwit a chess grandmaster! With a rematch planned for AlphaGo, this is a great opportunity to explore computation in the classroom. AlphaGo's intelligence is based on a combination of algorithms, machine learning and training. It would be fun to build games where users

playing against the computer program can also see the computation that it takes for the program to play.

Robots in the real world are so varied, but in the world of education we seem overly fixated on robot arms or simple wheeled designs. I've learned that robotics can overlap with a huge amount of computer science, and can be lots of fun. By taking a project-based learning approach with robotics as the theme, students can apply their understanding of design technology, engineering, electronics, programming and problem-solving. Real-world challenges for students to solve include: how does a robot understand its environment? How is that data processed and represented? What algorithms does it use to make decisions and reach its goals? How can it be programmed and given new tasks? How can we make robots that are safe and secure? And ultimately, how can we create technology that makes the world a better place? 

**Carrie Anne Philbin** is Director of Education at the Raspberry Pi Foundation, a Python Software Foundation and Computing At School board member, author, and YouTuber.

## #INSIGHTS



# WHAT IS DIGITAL MAKING?

More and more people are talking about the power of digital making, but what is this new area for educators?

STORY BY Oliver Quinlan

**A**cross the world, educators and policy makers are getting excited about the potential of digital technology for young people. As a school subject, information technology is being transformed in many countries to include in-depth technical understanding of programming and computer science. Understanding how computers work is really important, as is getting some experience of applying this understanding using programming. However, there is more to the power of digital technology than programming on screen, or considering the concepts that make

computers work. There is also creating things that solve a problem or explore a passion. There is making things that fit into the physical world and delight people who use them. There is integrating the skills from programming and computer science with those from other areas. This is where the concept of digital making comes in.

## Digital differences

It's a term used by different people for different skills. Digital making can refer to the making of any product using digital technology. This can include programming games and applications, but also producing

music, manipulating photos or creating visual art. For some people, making anything digital constitutes digital making. Others use this term to focus on the more technical end of digital skills, involving programming or electronics. However, it is important to remember that people can start off using relatively few technical skills and move towards learning them as their ambitions grow.

## Abiguity

People are often keen to find a concrete definition of what the term means. However, there is a danger that this

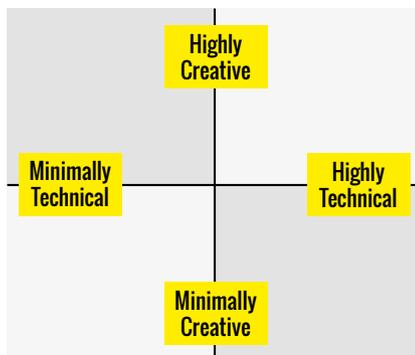
can shut down the creativity we see in the field of digital making education. Ambiguity can be problematic, but, in Essential Elements of Digital Literacies, Doug Belshaw has explored the idea that ambiguity of terms that are new and emerging can be positive.

‘Creative ambiguity is brought about when an intangible idea, process or way of thinking is defined in an imprecise way. It is a delicately balanced conceptual space in which the very nature of the ambiguity leads to creative outputs.’

He explains how ideas and terms move through types of ambiguity from their initial suggestion until they become a ‘dead metaphor’, meaningless through overuse. Although we tend to see concepts that are very specifically defined as being most useful, those which are developing allow people to interact with them in ways that create new meanings, new activities, and new possibilities. Once the definition becomes solidified, we are likely to see certain activities, quite possibly the more creative and less technical ones that get people started, excluded from the area of digital making. This potentially reduces the opportunities for some people.

### Mixing of skills

Digital making often includes a blend of technical and creative skills, which interact in complex ways. Technical skills can relate to programming, to electronics, and to physical fabrication. Creative skills can relate to problem-solving approaches in these areas, but also to the aesthetics of the end product and the experience a user will have with it. Both these types of skills have their own levels of challenge. I explored this in the Young Digital Makers report for Nesta, creating a diagram of these skills on two axes. ▶



# SELF-EFFICACY

STORY BY Samantha Baloro

**E**ducators have long been aware of the effect a child’s belief about their ability to achieve a task has on subsequent achievement. Self-efficacy is a term coined by Albert Bandura, who described it as the belief an individual has about their ability to complete a task. It influences the activities people participate in, the amount of effort they put in, and how long they persist. There are believed to be four sources of self-efficacy,

- Past performance occurs when an individual previously experiences success in a task. Bandura believed that strong self-efficacy is developed through success early during a task. Once success has been experienced, people are resilient to subsequent failures.
- Vicarious experience. Seeing other people successfully performing a task that is seen as threatening or difficult provides children with the idea that they can be successful too.
- Verbal persuasion. This involves encouraging and suggesting to an individual that they are able to complete a task which they previously found challenging.
- Emotional arousal. Stressful situations often elicit physical arousal which can harm performance, such as sweating or feeling tense and anxious. An individual is more likely to expect success if they are not met with such adverse symptoms.

Research has shown that students high in self-efficacy often work harder and longer, and are more resilient while experiencing a setback in their academic work.

Strategies have been suggested to develop self-efficacy:

### Goal setting

Encouraging children to set challenging but achievable goals ensures that they have previous experience of success, and of overcoming challenges and failure through persistence.

### Peer models

Allowing students to observe how peers have successfully completed tasks and overcome failure .

### Feedback

Providing learners with immediate, task-specific feedback from someone who they believe is trustworthy.

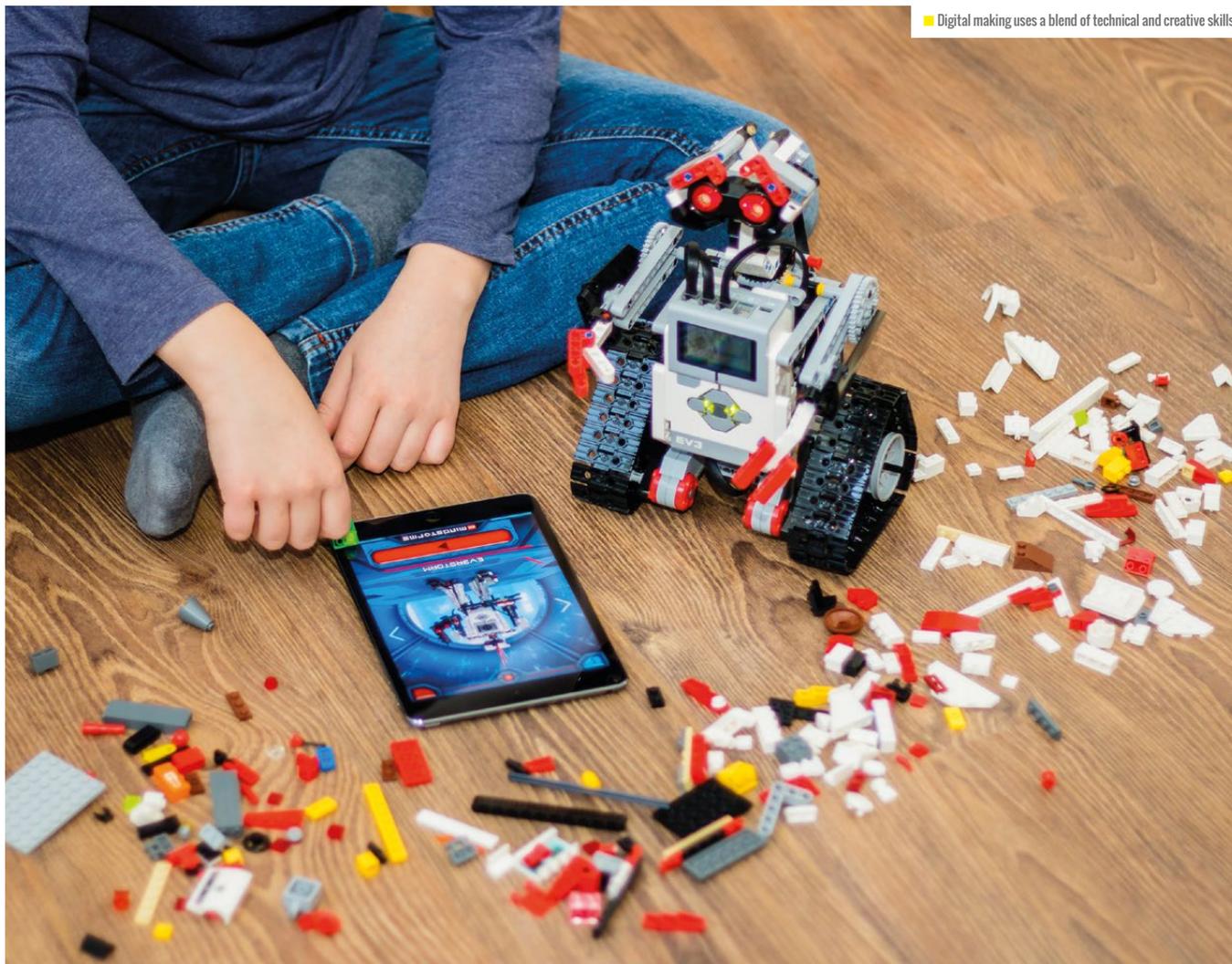
A recent study investigated the impact on self-efficacy of participating in the Hour of Code activity. It found children who took part, and especially girls, were more likely to report positive attitudes, as well as higher levels of self-efficacy, towards computing. These results might be explained by the fact that participating in the Hour of Code activity involved three elements that Bandura believed to be important in the development of self-efficacy:

- Past performance: students experienced instances of failure and success
- Vicarious experience: students observed their peers achieving success and handling failure
- Verbal persuasion: received from teachers

However, these results should be viewed with caution as the learners were not measured in the long term. Only self-report measures were used, so possible response bias needs to be taken into account.

More information on Bandura’s work can be found at [helloworld.cc/2qem0cE](http://helloworld.cc/2qem0cE).

The study looking at the effects of an hour of coding can be found at [helloworld.cc/2pblsoB](http://helloworld.cc/2pblsoB). (HW)



➤ Different digital making projects will sit in different places on this matrix. It may be that many projects that beginners undertake are higher on the creative axis than the technical. Often, technical skills

be in a highly creative mode, considering the outcomes they want to reach and designing a solution that could reach them. Then they might move into a more technical mode as they begin making,

don't yet have and need to develop, or to see the less obvious connections between their aims and the techniques they know and understand.

### Teaching digital makers

For educators, what is often important is understanding the kinds of skills makers are using at any one time, and what the level of challenge is given their current skills and understanding. The role of educators is, after all, to help people to learn. Identifying where learners are and what they could do next to ensure they are learning as well as making is an important approach, especially when open-ended and passion-driven projects are involved.

You can read more on these ideas in *Young Digital Makers* published by Nesta at [helloworld.cc/2oEBwli](http://helloworld.cc/2oEBwli). (HW)

## “ IDEAS AND TERMS MOVE THROUGH TYPES OF AMBIGUITY FROM THEIR INITIAL SUGGESTION

will increase as people learn new skills, and their ambitions for their projects increase alongside the skills they have to realise them.

A digital maker might themselves move around this matrix in terms of the skills they are using and developing at different stages of the project. At first they may

and have to problem-solve the practical tasks that need to be implemented to bring the project into being. Sometimes these technical elements will make use of approaches they understand well. At other times, they will have to bring a creative lens to look at how they might achieve certain outcomes using skills they

Translating...

# COMPUTING & GENDER: LOST IN TRANSLATION?

STORY BY Oliver Quinlan

**I**n industry and in exam classrooms, it is clear that computing is currently dominated by men. Girls can excel just as much as boys, so why is it so rare that they do?

Researchers Goode, Estrella, and Margolis tried to find out by running a three-year study in three Los Angeles public schools, including more than 200 interviews and lesson observations.

Their first finding was a lack of opportunities. The obvious counter to this is to run classes and create opportunities, which many readers will already be working on.

Girls often do not perceive computing as being relevant to them. Presentations of computer science and those who do it in the media show arcane work driven by an interest in computers for their own sake. Educators could reframe computing as something that is a powerful tool in the wider world. Creating a context for lessons and tasks that goes beyond an interest in

technology from the start will help students to see the relevance of the subject.

Many girls also have negative experiences when they do engage with computing classes. Boys often have lots of previous experience and see classes as a fun way of practicing familiar content. In observations in the study, boys regularly took over practical tasks and dominated discussions. Boys also reported how most of their learning and confidence came from peer networks that girls said they weren't part of.

Creating an inclusive classroom culture is paramount. Using techniques to ensure that all students are actively contributing, such as assigning roles and carefully structuring discussions, can help. Choosing widely relevant contexts beyond the computer can also put students on a more even level in terms of prior experiences.

Focusing on getting started with the mechanics of programming and syntax can miss higher-order thinking opportunities and

make students underestimate the academic value of computing. Girls often won't choose to engage with a subject unless they see it as a worthwhile qualification. Computing is often thought of as a niche subject for those with a special interest, rather than part of a palette of academic subjects that students with broader interests would choose. This can be avoided by making sure to weave demands for higher-order thinking in with the necessary early stages of learning the mechanics of programming. Highlighting the potential of computing as an academic subject for aspiring generalists as well as computer science specialists from the start can also help.

The forces that have made computing into a male-dominated area are complex. This research shines a light on how some of them work, and how teachers can begin to bring more gender diversity to the subject in schools.

The original article is available at [helloworld.cc/2oPJhzK](http://helloworld.cc/2oPJhzK). 

# HOSTING A HACKATHON IN A HIGH SCHOOL SETTING

Hackathons offer students the chance to collaborate and work on projects. Here, David Malan and Erin Carvalho from Harvard's CS50 offer tips for hosting your own hackathon for high school students

STORY BY David Malan, Erin Carvalho

**F**or us at CS50, the hackathon is an epic all-nighter during which students dive into a problem set or final project alongside classmates and staff (and food and photos and fun). It is by far the longest event we hold at Harvard, and it requires a lot of planning and preparation. Our event is 12 hours long, but in a high school setting for teachers using our Advanced Placement (AP) curriculum, teachers typically opt for a five- or six-hour event, starting at midday on a Saturday. It is a time for students to work on projects in a collaborative setting.

At the high school level, a hackathon can be a chance for students to dive deeper into a project and jump-start their excitement. It is

also an opportunity to promote campus-wide excitement for your students' final projects, which, once completed, can be showcased at a final project fair and used to spark future enrollment!

## Your strongest asset

After securing a venue (a school gymnasium is great and inexpensive), it's time to organise logistics. Parents are perhaps the strongest asset that teachers can use when it comes to hosting a hackathon. Aside from volunteering their time, parents can bring snacks and beverages, and work shifts to make sure things are running smoothly. They may own, or know people in, local businesses who can provide donations in kind. Solving some tech problems from potential sponsors could be a great way to get some food donated!

A cohort of dedicated parents can also help with finding experts in the students' fields of interest to assist them in building their projects. Never underestimate the connections students and their parents have in the community. Developing a strong

support staff for students is key to creating a memorable experience. Recruiting other tech staff from the school, and community members with coding experience, can help students to access the expertise needed to get them through the bugs in their code.

## Structuring your hackathon

A strong support system is key to a successful event. To ensure that students have the support they need, we ask them to submit project proposals before the event. These documents should include an outline: what they aim to achieve in their project; what they will accomplish, think they can accomplish, and hope to accomplish, respectively. We also ask students to consider what tools or skills they will need to learn to complete their project. Having this high-level overview can help you to better organise the groups students are working in, and the types of resources or support staff you will need at your event.

For more comprehensive details on how to plan your own hackathon, you can view our Medium post at [cs50.ly/hackathons](https://cs50.ly/hackathons). (HW)

## ORGANISING PARENT SUPPORT

Parents can be a great asset. Here are some ways they can contribute to your classroom and events.

- Help fundraise:
  - Bake sale
  - Car wash
  - Craft sale
  - Ice cream social
- Volunteer their time:
  - Setup
  - Clean up
  - Check students in
  - Supervise students
  - Take photos
- Bring snacks:
  - Baked goods
  - Pizza
  - Beverages
  - Candy
- Provide real-world problems for final projects from their companies or businesses
- If parents have jobs in the IT industry, they could give short seminars on specific software tools or programming languages



■ Students collaborate at a CS50 AP hackathon at NEST-m in New York

# EXPLORING PROGRAMMING WITH PHYSICAL COMPUTING

Primary school teacher Lorna Elkes discusses how physical equipment can be used to explore programming at different key stages

STORY BY Lorna Elkes

**W**e all know the benefits of exposing children to the concepts of algorithms and programming early, and most schools already have resources that can be used for this. For example, we retrieved our Bee-Bots from the early years team, much to the excitement of the Year 2 (seven-year-old) class, who instantly recognised them.

Building on familiar skills, the children were asked to program the Bee-Bots to follow a pre-agreed digitised shape. Limitations were imposed: before they could use the Bee-Bots, they had to write down their program. This

provided an additional challenge, requiring higher-level skills of logical reasoning. Several groups walked out the route for orientation. Instructions were passed to another group to check and then try out on the Bee-Bot, with debugging if required.

With the added competition of 'who could touch the Bee-Bot the least', the children were fully engaged in trying to understand the Bee-Bot's simple programming language.

The idea for this activity comes from the marvellous people at Barefoot Computing ([barefootcas.org.uk](http://barefootcas.org.uk)).

## THE NEXT GENERATION

Children of this generation grow up with computers and technology, meaning that coding skills come quite naturally. It's an exciting time for the next generation of computing technologists, as technology becomes increasingly important across all aspects of our lives. Involvement in this process is something that we are actively encouraging at Brookmead School.



### Using Scratch and Python to control a robotic arm

We were very lucky last summer to win a robotic arm education kit, thanks to filling in a form at a suppliers' conference ([helloworld.cc/2qbr4kX](http://helloworld.cc/2qbr4kX)).

Here's what Year 6's Mrs Nice had to say about it:

**Q** How much prep did you feel you needed?

**A** I had no previous experience of using a Raspberry Pi, but I found it straightforward. I familiarised myself with everything first.

**Q** What other skills did it help the children to develop?

**A** They were using an iterative approach to develop their programs, using creativity and logic as well as collaboration.

**Q** Which programming language did you use with it?

**A** Scratch when manipulating the virtual arm, and Python with the robotic arm.

**Q** How difficult was it for the pupils to master the virtual arm?

**A** To get started we watched clips of robotic arms in action for real-life examples. Then we used the virtual arm, which they manipulated easily: they were soon more expert than I was!

**Q** How did they use problem-solving to get the robotic arm moving?

**A** To start with, there was a lot of trial and error as they attempted to program the arm to move in a specific way, but they soon adapted their commands based on the movements they were seeing.

**Q** What will you do differently next time, and why?

**A** I'm aiming to become more familiar with Scratch and Python so that I can offer more support to groups where needed. I will also have higher expectations at the cardboard stage for understanding the importance of trial and improvement when it comes to the design process.

**Q** What ages would you say it was suitable for?

**A** Year 6 really loved it, and some could put the equipment together themselves. Next we're going to try it with Year 3. With the robotic arm, they can begin by trying out simple commands to move specific parts, and develop their ideas from there. With further exploration, the children can make up their own games using the virtual arm, and develop more complex movements. **(HW)**

# SCRATCH: THE FIRST TEN YEARS

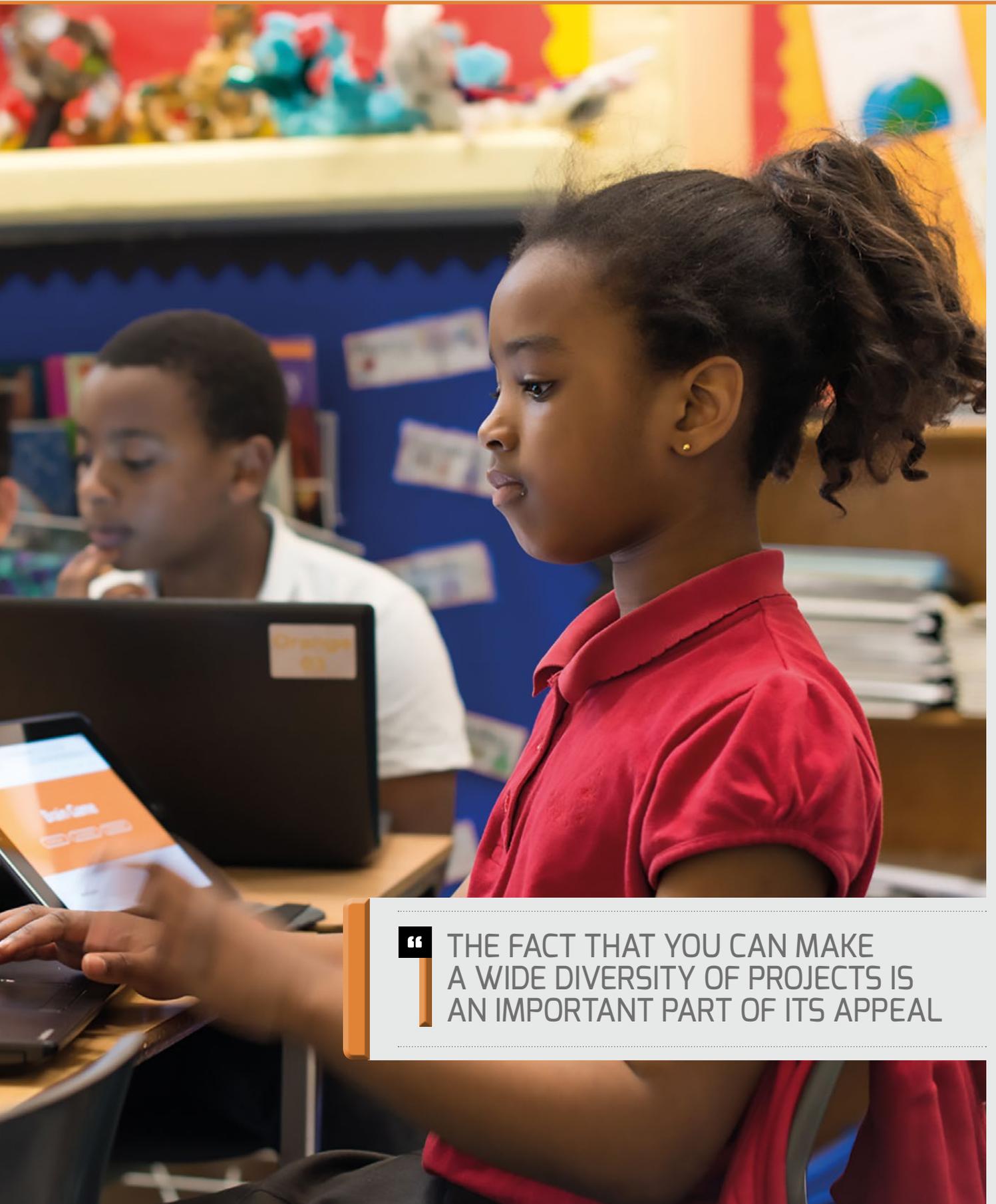
The educational programming language celebrates its tenth anniversary

**O**n 15 May 2007, a revolutionary programming tool was made freely available to the public. Developed by MIT's Media Lab, Scratch ([scratch.mit.edu](http://scratch.mit.edu)) featured a unique graphical interface that allowed programs to be assembled like building blocks, making it accessible to new coders of all ages. Since then, Scratch has gone through several iterations and experienced exponential growth in the number of users, becoming a mainstay in classrooms and after-school code clubs around the world. To celebrate its tenth anniversary, we chatted to Scratch Team leader Mitchel Resnick, Papert Professor of Learning Research at the MIT Media Lab.

Mitchel tells us that he and his team were inspired to start work on Scratch back in 2003 when working with youngsters in Computer Clubhouse after-school learning centres. "We saw that many of them wanted to create their own interactive stories, games, and animations, but there weren't good tools. So we produced Scratch to enable them to do it." He adds, "The fact that you can make a wide diversity of projects is an important part of its appeal."

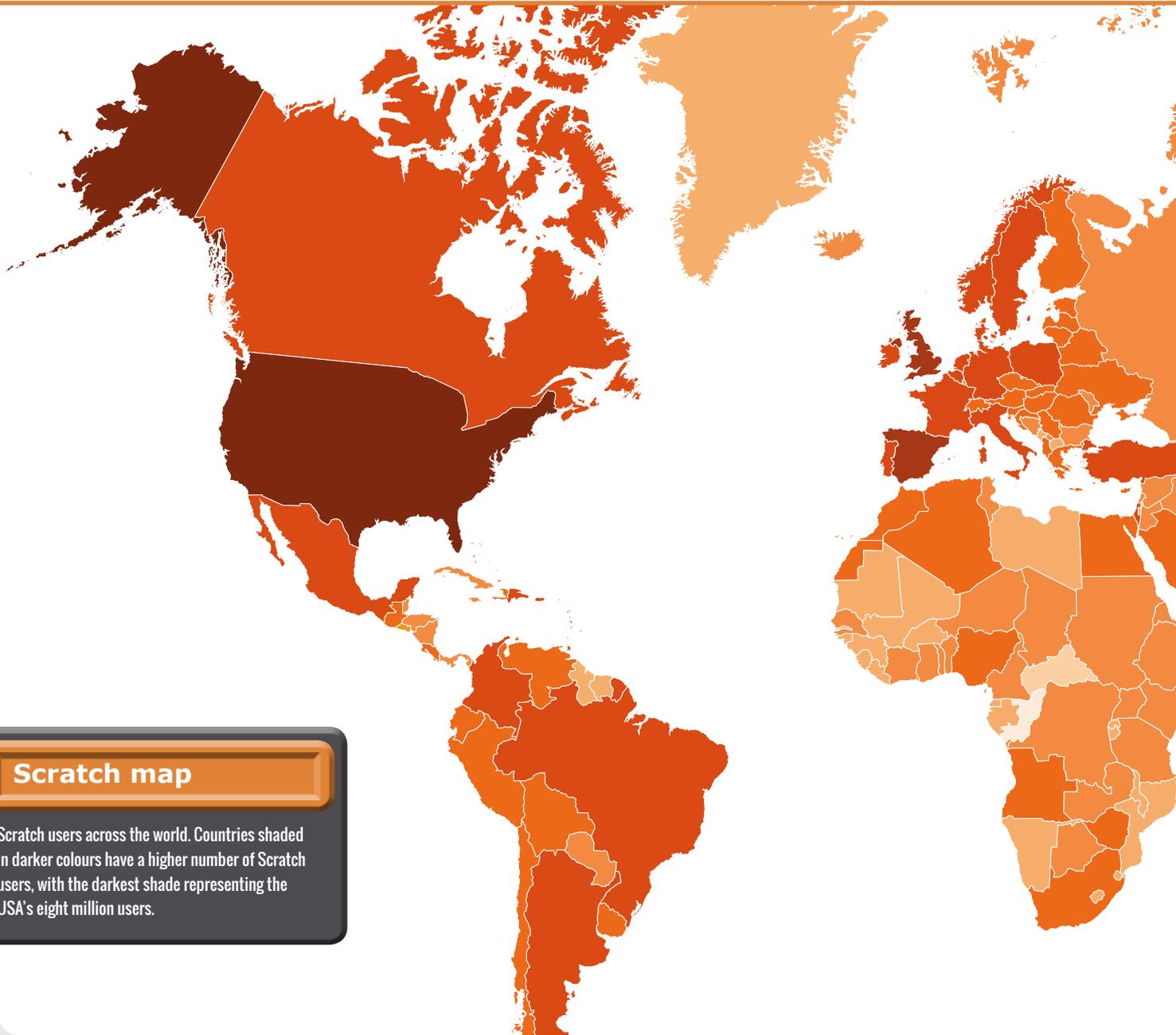
Mitchel says that Scratch was built upon a lot of lessons learnt over the years. "I worked closely with Seymour Papert, the developer of Logo, the first programming language for kids. So the work was very inspired by Seymour's ideas and his work on Logo, as well as other things that had been done in the field."





“

THE FACT THAT YOU CAN MAKE  
A WIDE DIVERSITY OF PROJECTS IS  
AN IMPORTANT PART OF ITS APPEAL



### Scratch map

Scratch users across the world. Countries shaded in darker colours have a higher number of Scratch users, with the darkest shade representing the USA's eight million users.

### Collaborative coding

While Mitchel says that the team always had high hopes that Scratch would eventually become something commonly used by young people around the world, "we continue to be surprised, delighted, and amazed at the wide range of ways the kids make use of it, particularly in relation to the online community: kids aren't just creating things of their own, but are finding all different ways of collaborating and sharing. You'll often hear kids say that they came to make a project but they stayed for the community."

This collaborative aspect was given a major boost by the arrival of Scratch 2.0 in May 2013, with the addition of an online code editor to the Scratch website. This also made it possible to instantly see the code inside other users' projects and 'remix' them. Previously, users would need to do all their programming in an offline code editor before uploading projects to the website. "I do think our second version of Scratch made it much more seamless to go between authoring and the community."

Following the launch of Scratch 2.0, there was a huge jump in the monthly number of

new projects and users, an upward trend that has continued ever since, resulting in a total of more than 18 million users and 22 million projects shared at the time of writing. Mitchel attributes some of its success to the general culture becoming much more interested in coding over the last few years, although he notes that Scratch also probably contributed to that: "One of the reasons the culture became interested in coding was because of Scratch."

Mitchel reveals that he is amazed by some of the Scratch projects created by users. "One of the things that makes it so

# SCRATCH

## IN NUMBERS

**22 MILLION**  
PROJECTS SHARED

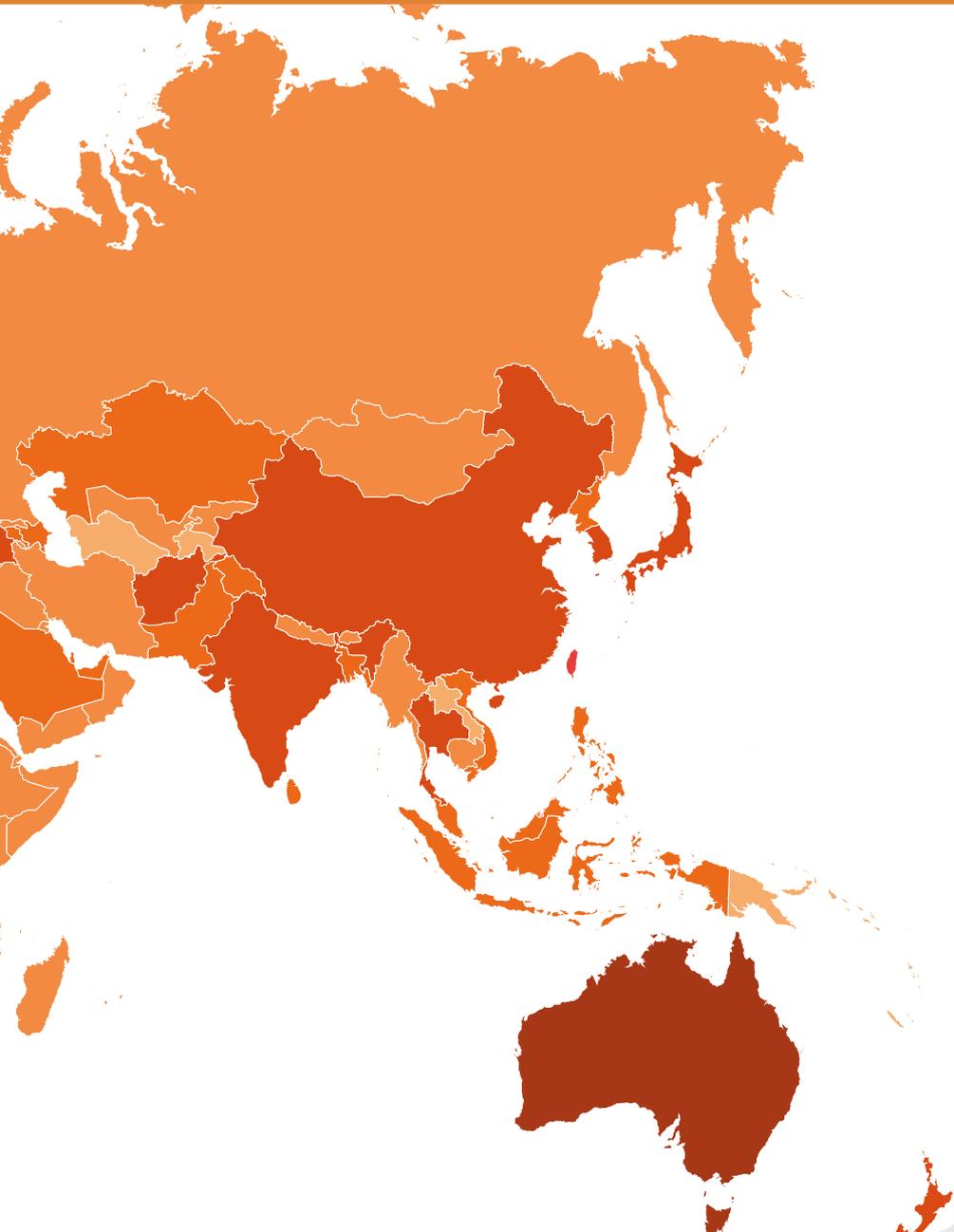
**18 MILLION**  
REGISTERED USERS

**3.4 MILLION**  
STUDIOS FORMED

**150+** COUNTRIES  
WITH USERS

**40+** LANGUAGES  
USED

**12 MODE AVERAGE**  
AGE OF USERS



exciting and rewarding to work on is that we are constantly surprised by what they do. A few weeks ago, there was a ten-year-old [autistic child] who did a project talking about the challenges of being autistic and trying to explain why it's OK to be different. So the fact [Scratch] gave kids the opportunity to express themselves that way is really, really important."

Mitchel is also impressed by a lot of the collaboration that goes on between Scratch users. "There is a young woman in South Africa who did this project called 'The Colour Divide' (v), influenced by her

upbringing and the scars of apartheid.

It's a story of a fantasy world where people are assigned one colour or another. But it's not that she had this idea which was deep with meaning – she then led a whole team of people working on it together. Some are doing the voices, some doing the illustrations, some developing the music ... a whole international team of kids."

### Code to learn

Scratch is used in many schools to introduce children to coding, and Mitchel believes that learning to code is an

essential part of education. "I do like to make the analogy of learning to write. We don't have people learn to write because everyone's going to grow up to become a professional journalist or novelist... It's a way of communicating with other people and a way of expressing your ideas. Also, as you write, you organise your ideas. I think we see the same thing with coding."

While Mitchel notes that lots of good jobs will be available for those able to program, learning to code has other benefits. "The act of coding helps you reason systematically: you have to think in a logical way in order

## Simple but powerful

The design team has always strived to make Scratch simple for beginners to learn, while offering powerful functionality for advanced users. Mitchel says, "My mentor Seymour Papert used to talk about a 'low floor and high ceiling': it should be easier to get started, but do more advanced things over time. We also talk about 'wide walls': the many pathways of getting started."

Most importantly, Mitchel insists that Scratch is for everyone: "All different ages, all different backgrounds, all different interests, all different cultures."

to create the programs. For us, it's just as important, if not more important, for kids to develop their voice, to express their ideas."

While he appreciates that the teaching of computing may have improved in recent years, "my critique is that a lot of the computer science classes are too focused on learning a set of computer



science concepts." While he notes that this is useful, "They generally don't focus on the ability to express yourself with the computer. And for me, that's an important element that's often left out."

He asserts that one of the most important aspects of learning to code is going through the creative design process.

■ The current release, Scratch 2.0, features online code editing and project sharing

"For me, it's very important to do things in a project-based way. We learn how to start an idea, and carry it through on a project. That's important for everybody – it's not just about computer science. Computers open up the possibility for kids to work on a much wider array of projects and learn the creative design process in a better way."

Asked whether computing lessons can help engage pupils who are otherwise not that interested in school, Mitchel replies, "It all depends on how it's introduced and used, because there's a risk that computers and coding can be introduced in a way that turns kids off: if you're just giving them problems to solve, they're made to feel inadequate because they get the wrong answer. On the other hand, if it's introduced in a way where kids work on projects that they care about, in collaboration with others, then I think it can be really motivating."

"We want to develop technologies and activities based on what I call the



■ Alongside Python, Scratch is one of the main programming languages used in Code Clubs

## SCRATCH TIMELINE

KEY EVENTS IN THE HISTORY OF SCRATCH

2003

Mitchel Resnick's Lifelong Kindergarten Group at the MIT Media Lab starts work on developing a new programming language

5 MAY 2007

The oldest existing Scratch project, Weekend ([helloworld.cc/2pdZ0um](http://helloworld.cc/2pdZ0um)) is uploaded

15 MAY 2007

Following various beta versions, Scratch 1.0 and the Scratch website are made publicly available

26 MAY 2007

Scratch 1.1 is launched, featuring updated Control blocks and other tweaks

four Ps of creative learning: projects, passion, peers, and play. If we introduce it in the right way, then it can help kids develop as creative thinkers, and I think there's nothing more important in today's society."

## The future of Scratch

The next stage in the Scratch story is version 3.0, currently in development, although a public release date is still some way off. Mitchel reveals that one key area of improvement will be with regard to physical computing. "We're designing it to be much easier for people to connect external hardware. We talk about trying to meet kids where they are, to be part of [their] lives. So to connect to the physical world is part of that."

He notes that kids also live in the online world, "so we want to make it easier to connect Scratch to online services and data sources. So if I make a Scratch project with a weather map, I should be able to pull in the latest weather data.

improve the experience for newcomers. To this end, Scratch 3.0 will feature both vertical and horizontal block placement. "We're planning to have a couple of different 'grammars'. In Scratch Jr [a mobile

■ The Scratch website made its public debut on 15 May 2007; projects could be played but not edited directly

## “ THE ACT OF CODING HELPS YOU REASON SYSTEMATICALLY: YOU HAVE TO THINK IN A LOGICAL WAY

Or, if I wanted to get a song for a Scratch project, from within Scratch I should be able to go to Spotify and pull in a music clip." Users will also be able to connect to external online services for functionality such as speech recognition.

Mitchel admits, "Scratch has not done such a great job of supporting use on mobile devices, so that's something we're paying a lot more attention to now." He adds that the team is always looking to

version of Scratch aimed at younger kids], we used horizontal blocks, partly because instead of words we use icons."

As for what the next ten years may hold for Scratch, Mitchel says, "My hopes are that it does become more of a type of literacy, an extended form of writing that lets you create all sorts of things. So it'll be part of the creative toolkit that kids grow up with, for when they want to make something dynamic and interactive." **(HW)**



■ Scratch Team leader Mitchel Resnick is the Papert Professor of Learning Research at the MIT Media Lab

**2 DEC 2007**

Scratch 1.2 features several new blocks and improvements

**2 SEP 2008**

Scratch 1.3 includes new features such as lists

**6 DEC 2008**

The first annual Scratch Day event takes place at MIT

**2 JUL 2009**

The last 1.x release, Scratch 1.4 has a revamped UI and many new features

**9 MAY 2013**

Scratch 2.0 is released to the public: it features a redesigned editor and website, and allows you to edit projects directly from your web browser

(HW)

# SUBSCRIBE

Sign up today for a year - prices start at FREE!



Subscribe Today

- Get three term-time issues
- Have them delivered directly to your door
- Hello World is not available in stores!

## HOW TO SUBSCRIBE

- Visit us online: [helloworld.cc/sub1year](http://helloworld.cc/sub1year)
- Call our subscriptions hotline: +44(0)1202 586848



### Not a UK-based educator?

- Buy any issue for £6  
Visit: [helloworld.cc/buyissue](http://helloworld.cc/buyissue)
- Subscribe from £15 for 3 issues  
Visit: [helloworld.cc/sub1year](http://helloworld.cc/sub1year)

# SUBSCRIPTION FORM

YES, I'd like to subscribe to **Hello World** magazine!

This subscription is:  For me  A gift for someone

HW#2

**YOUR DETAILS**  Mr  Mrs  Miss  Ms

First name ..... Surname .....

Address .....

Postcode ..... Email .....

Daytime phone ..... Mobile .....

Are you a UK-based educator?  Yes  No

**GIFT RECIPIENT'S DETAILS ONLY**  Mr  Mrs  Miss  Ms

First name ..... Surname .....

Address .....

Postcode ..... Email .....

Daytime phone ..... Mobile .....

Is the recipient a UK-based educator?  Yes  No

*If you're giving Hello World as a gift, please complete both your own details (left) and the recipient's (right).*

Your Hello World subscription will start from issue #2, unless you specify the issue number here: .....

**RETURN THIS FORM TO:** Hello World Subscriptions, Select Publisher Services Ltd, PO Box 6337, Bournemouth BH1 9EH. We'll be in touch!

Please tick this box if you DO NOT want to subscribe to the Raspberry Pi Foundation Education newsletter (no spam).

ANNE LONGFIELD OBE CHILDREN'S COMMISSIONER FOR ENGLAND

# HELPING CHILDREN FIND THEIR WAY IN THE DIGITAL WORLD

We need to make sure children have the information, power, and resilience they need to thrive online

**G**rowing up in a world without Google, Instagram, Facebook, and Twitter, I could never have imagined the opportunities the internet would offer children to learn, to explore and to engage with others. Now it's an irreversible fact of life. My five-year-old nephew has been deftly using his parents' tablet for years: to him it's an everyday object.

But parents and teachers worry that we are out of our depth, unsure about the impact the internet is having on children's lives. While adults should be the first line of protection against potential online dangers, there is much more that can be done to support children.

## Growing up digital

Earlier this year I published *Growing up digital*, a year-long study by experts with backgrounds in media law, online bullying, safety, and children's digital use. We proposed ways of giving children the resilience, information, and power they need.

As teachers know, children are often most worried about online bullying and inappropriate pictures of themselves appearing online. Many don't feel they have control over their social media lives, agreeing to terms and conditions for apps that are so long and complex that they could never be expected understand them fully. They have no idea how to report unacceptable content or behaviour they

experience online and even when they do, they are usually unhappy with the action taken.

Social media companies urgently need to rewrite their terms and conditions, and we need a mediator between children and the internet giants – an independent Digital Ombudsman – who can help to have content removed from sites if children are unhappy with a picture or a post.

## Digital citizenship

Children also need to be taught from an early age about their online rights and responsibilities. It makes sense for all four- to fourteen-year-olds to learn how to get the best out of the internet, while also avoiding its dangers. This obligatory part of the curriculum would include what it means to be a responsible citizen online, how to protect their rights online, how to respect others' rights online, and how to both disengage and engage with the online world. Younger children want to learn these skills, but sometimes feel uncomfortable talking to teachers and parents about the internet. They would rather talk to people of their own age, so digital citizenship should involve older children who younger kids are more likely to trust.

There is already a range of creative, fun, and stimulating digital curricula available or under development. Such a programme might be based on the 5Rights initiative: the right to remove what you yourself have put on the internet, the right to know who has access to your data, the right to safety and support when something online is distressing, the right to have the power to switch off the internet, and to understand the purposes of the technology that you are using.

We wouldn't let a child roam around the 'offline' world without teaching them the skills to do so first. It's time we did more to prepare children for a digital world so it can be a safe place for them to thrive. **(H&W)**

## GROWING UP DIGITAL

The *Growing up digital* study makes it clear that we need a powerful mediator for children within the industry. Although much of the behaviour children complain about online – bullying, sexting, harassment – is illegal, a number of studies have found that children often don't know how to report their concerns. A Children's Digital Ombudsman, based on a model that has worked well in Australia, would mediate between under 18s and social media companies over the removal of content. It would operate in a similar way to the Financial Ombudsman Service, funded by social media companies, but independent of them.

### Anne Longfield OBE

Children's Commissioner for England Anne Longfield is calling for children to have the information, power, and resilience they need for the digital world.



**PENNY PATTERSON** QUALITY ASSURANCE INSPECTOR

# ONLINE SAFETY IS SAFEGUARDING

Keeping Children safe in Education 2016 (KCSIE) saw the bringing together of safeguarding and online safety. Good schools have always done this, but it is now a requirement for all

**T**he danger with online safety is that we spend our time worrying about nearly all of our pupils, instead of looking for signs and symptoms that may indicate that a particular pupil is more at risk.

As well as teaching pupils how to be safe online, schools must also provide safe online spaces for learning. Appropriate filtering and monitoring should be in place, both age and role specific. The UK Safer Internet Centre issued useful guidance, also referenced in KCSIE.

## Filtering

Filtering must implement the IWF (Internet Watch Foundation) list of child abuse sites and the police-assessed list of unlawful terrorist content; any sites in these lists cannot be unblocked.

Filtering should have further categories which are used to set a profile that is suitable for phase and role; this may include blocking adult content, hate speech, gambling, etc. Filtering must be customisable in school. The school must be able to block or unblock single sites as needed for individuals or groups in the school. It should not over block. Filtering must be able to block content on hand-held devices that use WiFi, and manage the languages in the school.

Filtering will never be perfect. Breaches are always possible, but it should provide a supportive and safe environment.

## Monitoring

Monitoring starts with the adult in the classroom, and appropriate supervision of all online activity. A software monitoring solution should identify content and activity which is illegal, bullying, grooming, raises concerns for the child's mental or physical health, violent, extremist, pornographic, or related to drug and substance abuse.

## TEACHING PUPILS TO KEEP THEMSELVES SAFE

Complementary to other safeguarding arrangements is the appropriate teaching of online safety in the curriculum, offering regular opportunities to develop resilience, responsibility, and an understanding of risk.

Reports from monitoring software should go to the safeguarding lead, to be considered alongside other knowledge of the pupil. If deliberate bullying behaviours are seen, the school behaviour policy should be followed.

## Support rather than blame

The school must have clear processes in place if a pupil accesses unsuitable material. Consideration should be given to supporting, not blaming the pupil, looking at the material accessed and making a safeguarding referral if needed. In school this gives the opportunity to look carefully at the risks that have become 24-hour-a-day issues for children and young people with smart phones, tablets, and other devices with internet access. It is important to talk about social networking, gaming, and messaging to support children in looking at behaviours, friendship, responsibility, and reporting. Quality discussions with trusted adults give children the vocabulary and the opportunity to talk about online life, when it is going well, but also when there are problems. [\(HWF\)](#)

### Penny Patterson

Penny is a Quality Assurance Inspector for Havering Education Services. She says a safe environment in school, online and offline, provides the security to enjoy learning.



■ The first Pycon UK teachers' track, from 2012. Readers may recognise some familiar faces...

# MICROPYTHON: GOOD THINGS COME IN SMALL PACKAGES

MicroPython developer (and former teacher) Nicholas Tollervey explains why Python is one of the world's most popular programming languages, and what's special about MicroPython

STORY BY Nicholas Tollervey

**P**ython is remarkably easy to learn, with a wide range of free online resources available, aimed at beginner programmers. It is widely used throughout all sectors of industry, academia, and the arts, and by many of the most famous organisations on the planet. Python is also backed by an open, diverse, and global community of hundreds of thousands of volunteers who organise conferences, write

documentation, and work on the Python language and its wider ecosystem.

## Python and education

Many teachers and students are active in the Python community, and many Python developers volunteer time to support, create software, and write documentation to help teaching colleagues. Many PyCons (community-organised Python

conferences) run specialist education tracks for both teachers and students, a long running and well-established example being the one held at PyCon UK.

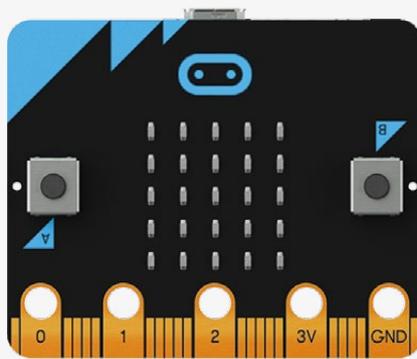
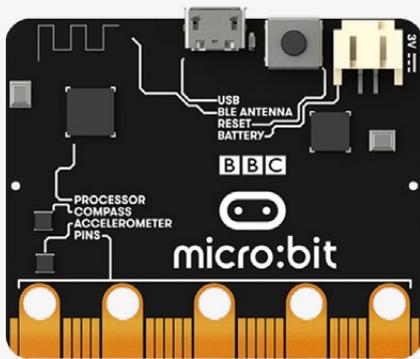
Python's inventor, the Dutch programmer Guido van Rossum, likes to point out that "Python is for everyone". He regularly supports the educational uses of Python, and in 2013 he wrote a short letter to the world's young programmers:

“

DEAR (INSERT NAME HERE),

I HEARD YOU ENJOY A CERTAIN PROGRAMMING LANGUAGE NAMED PYTHON. PROGRAMMING IS A WONDERFUL ACTIVITY. I AM A LITTLE JEALOUS THAT YOU HAVE ACCESS TO COMPUTERS AT YOUR AGE; WHEN I GREW UP I DIDN'T EVEN KNOW WHAT A COMPUTER WAS! I WAS AN ELECTRONICS HOBBYIST THOUGH, AND MY BIG DREAM WAS TO BUILD MY OWN ELECTRONIC CALCULATOR FROM DISCRETE COMPONENTS. I NEVER DID DO THAT, BUT I DID BUILD SEVERAL DIGITAL CLOCKS, AND IT WAS AMAZING TO BUILD SOMETHING THAT COMPLEX AND SEE IT WORK. I HOPE YOU DREAM BIG TOO – PROGRAMMERS CAN MAKE COMPUTERS (AND ROBOTS!) DO AMAZING THINGS, AND THIS IS A GREAT TIME TO BECOME A PROGRAMMER. JUST IMAGINE HOW MUCH FASTER COMPUTERS WILL BE IN FIVE OR TEN YEARS, AND WHAT YOU WILL BE ABLE TO DO WITH YOUR SKILLS THEN!

- GUIDO VAN ROSSUM (INVENTOR OF PYTHON)



## THE MICRO:BIT

The micro:bit is aimed at 11-year-olds. It has a small LED-based display, a couple of buttons, a compass, accelerometer, radio, and pins to which you can connect peripherals. A million of these devices have been delivered to children in the UK. Unfortunately, many teachers tasked with handing out the devices were unsure what to do with them, despite the provision of many lessons and tutorials, freely available on the web (hint: search for "MicroPython micro:bit tutorial").

■ The BBC micro:bit bare-board micro controller

### Introducing MicroPython

Recently a new version of Python has been grabbing lots of attention in the education world. It's called MicroPython and it's designed to run on the smallest of computing devices: the microcontrollers at the heart of the Internet of Things movement and the embedded development renaissance.

Rather than static, dead objects, they become instructable and independent, and demonstrate behaviours that bring them to life, much like the magical objects in fictional worlds such as that inhabited by Harry Potter.

### MicroPython on the micro:bit

One device that runs MicroPython is the micro:bit (see box). The micro:bit provides an opportunity to use a fun and engaging

that learning MicroPython on the micro:bit is an economically valuable skill, and could land you a job as a professional programmer! MicroPython is also not going away any time soon: diverse organisations including the European Space Agency, Adafruit Industries and several universities already make use of MicroPython in their work, products or courses.

### Taking things further

If you want to find out more about Python, there are many free online tutorials. Several good examples can be found at Python's own website, [python.org](http://python.org).

There are two options for using MicroPython on the micro:bit: a browser-based MicroPython editor at the micro:bit website ([python.microbit.org](http://python.microbit.org)) or a very simple code editor for beginner programmers called Mu ([codewith.mu](http://codewith.mu)). Mu is a native application, which gives it more features, whereas the browser-based editor is more convenient, as you don't need to install anything. There's a beginner-friendly tutorial for MicroPython on the micro:bit at [helloworld.cc/2oTYrJ9](http://helloworld.cc/2oTYrJ9), and my book, *Programming with MicroPython*, will be published in June by O'Reilly Media.

Finally, if you're a teacher and you will be in the UK during the Autumn half term, book a place at this year's PyCon UK in Cardiff ([pyconuk.org](http://pyconuk.org)). It's a very friendly conference, and the education track gives teachers and developers an opportunity to work together and learn from each other.

I hope to see you there! [@helloworldcc](https://twitter.com/helloworldcc)

## “ PYTHON LEARNED ON THE MICRO:BIT IS THE SAME AS THE PYTHON ON THE RASPBERRY PI

MicroPython is the work of the Australian physicist Damien George. He wondered whether it would be possible to create a full reimplementation of Python for highly constrained devices, and surprised and delighted everyone by succeeding in spectacular fashion.

Today, MicroPython runs on all sorts of devices and is used in all sorts of situations where size is a constraint, from satellites bound for space to boards the size of a fingernail. Because MicroPython is a full reimplementation of Python 3, all the friendly and powerful features of the language are available on devices that are small enough to be stuck, sewn, soldered and screwed in and on to other objects. This turns everyday things into programmable, 'enchanted' objects.

device in a way that is much more inspiring than printing 'Hello, World!'. In only a few lines of code it can display pictures, scroll text, play music, speak (with a simple speech synthesiser), and respond to user input via the buttons or via gestures detected by the accelerometer.

While there are several languages you could use with the micro:bit, MicroPython is the only one backed by a global, diverse, and proactive community of volunteers, many of whom are passionate about education and itching to support teachers. MicroPython is the only platform with continuity of learning: Python learned on the micro:bit is the same as the Python on the Raspberry Pi, on your laptop or in lambda functions running on Amazon's global cloud infrastructure. This means

# MATHEMATICAL MUSINGS

Mark Thornber uses Python to work out the frequency of characters in some text as a way to crack the Caesar Cipher, which he introduced in Hello World's launch issue

Last time we looked at how to create a coded message using the Caesar cipher. We shift all of the letters along in the alphabet by a fixed number of places. To decode such a message we just shift back by the same amount. If we don't know the shift we can look for the most common letter in the message and assume this corresponds to the most common letter in the English language: the letter e.

Finding the most common letter in a message poses a few nice challenges for a beginner programmer: how to count, how to store the results, and how to find the letter with the largest frequency. Let's look at these in turn.

## Counting letters

To count, we can go through the message letter by letter with a `for` loop and then store the results in a list. As we did last time, we can turn letters into numbers with `ord()`. We can then use those numbers to update a position in our list.

```
countList = [0 for i in range(128)]
for letter in message.lower():
    try:
        countList[ord(letter)] += 1
    except:
        pass
```

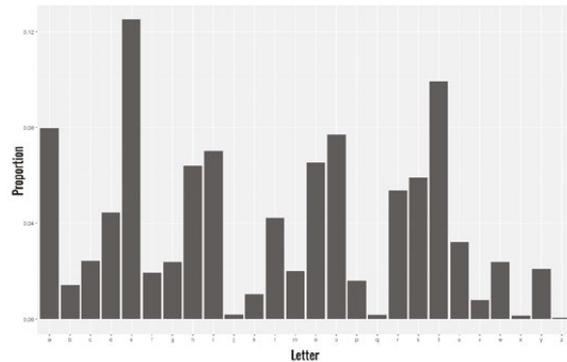
(See the box for how `try:` and `except:` work...)

## Finding the maximum

The frequencies of letters in the message are now stored in positions 97 to 122 (since we converted to lower case before counting). The other positions contain frequencies of punctuation etc. Next we need to find the position with the largest frequency. One way is simply to go through the list one item at a time.

### try and except

There's a useful trick here - the `try ... except ...` construct allows us to avoid problems with non-ASCII characters like foreign letters, symbols, or emojis, etc. All of these are stored in Unicode and either cause problems for `ord()`, or return numbers larger than 127. Normally this would lead the program to crash, but in this case the `try` statement does not work, so the program moves on to the `except` statement and does nothing.



The proportion of different letters in Alice in Wonderland

```
maxPos = 96
maxFreq = 0
for i in range(97,123):
    if countList[i] > maxFreq:
        maxPos = i
        maxFreq = countList[i]
```

## Shifting the message

At this point, the location of the most common letter is stored as `maxPos`. Since we hope this corresponds to e, normally in position 101, we can calculate the amount to shift by `shift = 101 - maxPos`.

As a check, suppose the original message was shifted three places forward. We would expect the most common letter in the secret version to be h, at position 104. The formula gives `shift = -3`. If we now use the program from last time with a shift of -3, we should recover the plain text of the original message.

With very short or unusual messages, this method may fail because the most common letter is not e. As a backup you could add a part that assumes the most common letter in the message corresponds to the second most common letter in the English language. Do you know what this is? Why not count a few books from Project Gutenberg to find out?

Next time we will look at more modern and secure methods to encrypt messages with prime numbers. [\(HW\)](#)

**Mark Thornber** has been a maths teacher at Durham Johnston School for the last 25 years. Mark has been interested in the mathematical parts of computing since owning his first ZX81.



**RIK CROSS** SENIOR CONTENT AND CURRICULUM MANAGER

# EMBRACING CHAOS

The Senior Content and Curriculum Manager at the Raspberry Pi Foundation explains that 'chaos' is often a good thing, and gives advice on making the most of the high energy and enthusiasm that comes with running a Code Club

**A**t a recent Code Club meet-up, I was chatting to a volunteer who asked for tips on running a Code Club. They felt that some aspects of their club were, in their words, 'chaos'.

This got me thinking. In some ways, my club can be chaos too – and I think that's a good thing! Obviously there is a need for rules and structure within a club, but children also need an environment in which they feel free to experiment and share ideas.

Here are a few examples of chaos in my Code Club.

## Children work on different projects

They are personalising their learning, working on a project that interests them, at their own pace. I've known children to skip projects that don't interest them, or spend weeks on a project that captures their imagination. Some children may, after completing a handful of projects, decide that they have enough knowledge and skill to build something of their own.

## Children move around a lot

They look around at what others are making, getting ideas and inspiration. They often invite others to play (i.e. test) their finished projects, and then make improvements based on feedback they receive. Children get a lot of motivation from seeing others huddled around their computer, playing and enjoying a project they have made. This prompts children to make sure that their project is of high quality before allowing others to play with it.

## It can sometimes get loud

Children ask each other questions, and move around the room to help each other out. They test each other's projects, giving verbal feedback, sharing ideas or even just having fun with the things they've created. When children are motivated to create things that interest them, I think it's important that they have time to enjoy the things they've made.

## Children play games

My club uses online Scratch, and so as well as playing each other's games they do get time to play other Scratch projects online. Obviously it's important that this doesn't dominate a club, but I think this helps children to learn lots about what's possible with Scratch, especially when moving past the basics. Posting their own creations online is also a great opportunity for children to get real feedback from the wider Scratch community.

What some volunteers call 'chaos' is in fact part of the fun, and part of the learning experience. It is how children show the excitement and enthusiasm they feel when making things with computers. All this differentiates a Code Club from regular computing classes, so I always advise volunteers to embrace it! **(HWD)**

## Rik Cross

Rik Cross is a former secondary teacher, responsible for developing the Raspberry Pi curriculum. Rik enjoys learning new skills and is currently learning to brew beer.



■ Using Code Club resources

# HOW DOES CODE CLUB BENEFIT CHILDREN?

An independent evaluation found that Code Club has a measurable effect on children's programming skills, but we need to dig deeper to understand how we develop their computational thinking

STORY BY Oliver Quinlan

**I**n the last five years, Code Club has reached hundreds of thousands of children. There are more than five and a half thousand clubs in the UK, and thousands more across the world. In that time we have learned so much about creating

opportunities for primary school children to learn to code. We are always trying to develop and learn from feedback. Last year a grant from Nesta and the Cabinet Office allowed us to commission the National Foundation for Educational Research to run an independent evaluation of Code Clubs.

It was chosen at random which children would go to Code Clubs and which would carry on as normal. Researchers measured the programming and computational thinking skills of all these children. They found that attending Code Club significantly improves pupils' skills in Scratch, HTML/CSS, and Python, but the trial did not find an impact on pupils' computational thinking from attending a Code Club. Both groups of children increased their scores on the computational thinking test at the end of the trial. Teachers and children also reported many positive experiences of Code Club. They reported children growing

in confidence, in their ability to work with others and teach their peers, and in their use of programming as a tool.

*"It's a fantastic resource and it enables pupils to succeed when they haven't elsewhere. It's a very good skill for pupils to develop and it opens up lots of opportunities for them in the future. It's a great thing to be exposing children to and in the future they will be glad that they did it."*

Organisations undertake these kinds of trials to find out what is working and what they can learn. At Code Club we have been really pleased with this trial as it has met both of these aims. It is heartening to know that children are measurably improving their programming skills, and that our work helps schools to set up and run Code Clubs well. It's also important to know what we can

## HOW TO START A CLUB

We're always looking for people who can help to start new Code Clubs. You might be a volunteer who wants to teach children to code; or perhaps you represent a venue such as a school, museum, or youth club, that wants to find someone to help run a club. You don't need to be an expert in programming, just have an enthusiasm for helping children to learn. Get started at: [codeclub.org.uk/start-a-club](https://codeclub.org.uk/start-a-club)



learn from and work to improve, and the trial has given us some strong pointers to act on.

The strongest of these is to review what computational thinking means for Code Club, and how clubs develop it in children. Computational thinking skills are hugely important. This is not just for programming, but also for understanding the increasingly digital world and solving problems. There is a lot of discussion about exactly how these skills work and how people develop them. We are engaging with the research in this area to better understand how it works and what we can aim to achieve for children in Code Clubs.

### **Growing and evolving**

The trial has given us an even better understanding of what and how children learn. We are looking at how we can collect more information and measure the learning of children in Code Clubs. Fun and inspiration are important to us. We don't want to start setting regular tests for children, but we do want to help them understand what they have learned. If they can share this with us then we know how well we are doing.

Much of the power of Code Clubs comes from the volunteers and educators who run them. We are using what we have learned to continue to develop our support for these fantastic people. This involves finding new ways to identify and share the great practices

sure they are still learning new things. We have already put plans in place to create more Scratch resources and cover more programming concepts this way.

Getting feedback can be challenging, but a fresh set of eyes so often sees

## **“ ATTENDING CODE CLUB SIGNIFICANTLY IMPROVES PUPILS’ SKILLS IN SCRATCH, HTML/CSS, AND PYTHON**

being used across our clubs. We also want to offer more help with some of the challenges highlighted by the trial, such as moving from visual to text-based programming. We know that the tools for Python are less appealing than those for Scratch, and we are looking at how we can support the development of tools that make text-based languages more fun and engaging.

Children of this age love Scratch. Some teachers in the trial told us they felt it would suit their children to stick to this fun and powerful tool. If they do, we want to make

things that you would not. This rigorous evaluation has reinforced the great work that is happening in Code Clubs. It has also challenged us to continue to improve and think even more deeply about what we do. We now have an even clearer picture of how we want Code Club to grow and continue to evolve. [\(HW\)](#)

You can download and read the full version of the Code Club evaluation report from [rpf.io/research](http://rpf.io/research)



■ New ECS teachers undertaking collaborative activities during a Saturday professional development session

# EXPLORING COMPUTER SCIENCE AND GIRLS WHO CODE

What pedagogical approaches are needed to promote inclusion within CS? Neil Rickus examines two programmes in Los Angeles, successfully delivering CS to a diverse range of pupils

STORY BY Neil Rickus

In the last issue, I examined the differences between the Exploring Computer Science (ECS) curriculum and the English national curriculum, with a focus on how the ECS curriculum's content aims to broaden participation. The pedagogy implemented during sessions also aims to promote inclusion within the subject, so this article will discuss the approach taken not only within ECS, but also through the Girls Who Code (GWC) after school clubs. These have now reached more than 40,000 girls, in 6th-12th grade (11-18 years) in all 50 US states. GWC also runs summer programmes for 10th-11th grade (15-17 years) pupils. Further information can be found at [girlswhocode.com](http://girlswhocode.com).

The ECS curriculum can be downloaded free from [exploringcs.org](http://exploringcs.org), and is usually studied by 9th grade students (14-15 years) with no prior knowledge of computer science (CS). The curriculum has a specific topic focus for each half-term with individual lesson plans available, although teachers can adapt the sessions to the needs of their pupils.

## Community relationships

Both projects focus on the importance of creating a community within the classroom, and on ensuring the wider community benefits from the programs or content developed. Within ECS sessions, decisions relating to lesson content and the structure of sessions are discussed as a group. For example, during observations in South Central Los Angeles, the teacher said to the class, "we're going to agree as a community", and she ensured all pupils had a chance to contribute to discussions. The teacher took care to use non-confrontational language, which ensured communication in the class remained calm and pupils could express their opinions in a safe environment. Websites developed by pupils as part of the ECS curriculum were focused on having a community benefit, and children in the class were encouraged to share their work with a wider audience where appropriate. The ECS team outlined how this helps give pupils a 'tech identity', and it also led to discussions around the topics being developed within the sessions,

which ranged from Black Lives Matter to gun crime. This approach is related to Papert's theory of constructionism: based on the belief that pupils learn best through making things for others.

The importance of community was also evident during the GWC sessions I observed. Participants had previously created their 'rules of sisterhood', and the girls were encouraged to share their successes at the start of each weekly session, which included a girl applying for the UN's Youth Ambassador programme. The main project undertaken during each term should have an impact on the local community, and the girls had decided to create an app connecting people with similar interests in the local area. GWC members were also encouraged to assist with computing lessons for older members of the community where possible, which were held at the same location as their meeting.

## How could I use this in the classroom?

Identify creative, practical projects, which are likely to have a community benefit. This could be linked to the school community,



the local area or an area of interest to individuals. Links could be made with schools in other countries through Skype in the Classroom, and projects tailored to aid those less fortunate. Where possible, opportunities should be given to share successes both inside and outside the classroom.

### Collaboration

While closely linked to community relationships, specific interventions were put in place to facilitate effective communication among members of the group. Throughout the GWC projects, each girl is assigned a different role at each stage of their work, which is linked to a project management

work collaboratively on a range of problem-solving tasks, often away from the computer, in groups of various sizes. Opportunities are also provided for the children to communicate with people outside the classroom, such as family members, when examining the need to purchase a computing device. The need for pupils to work together was also

## CAS #INCLUDE

There are three initiatives outlined here: community relationships, collaboration, and role models and careers. Each area is recognised by the Computing At School (CAS) #include working group, which aims to give all students the opportunity to study computing; particularly pertinent when taking a 'gender-neutral approach to the teaching of computing' [casinclude.org.uk](http://casinclude.org.uk)



## ENCOURAGE PUPILS TO WORK TOGETHER THROUGH INITIATIVES SUCH AS PAIRED PROGRAMMING

methodology followed by the group. A virtual learning environment (VLE) is used as an area for pupils to store their work as the project progresses, and the technology includes a messaging platform to allow the girls to communicate in a safe environment, including with the course instructor.

Collaborative tools are used within several ECS sessions. In addition to this, the initial unit during the programme requires pupils to

emphasised during professional development (PD) activities for ECS teachers. A range of strategies was modelled during observed PD sessions, including paired programming, gallery walks, and think, ink, pair, share.

### How could I use this in the classroom?

Encourage pupils to work together through initiatives such as paired programming, where one pupil manages the input devices

▶ and the other advises on the development process, or 'C3B4ME', where pupils must ask three of their peers for assistance before being offered guidance by the teacher. Alternatively, Digital Leaders could be used to support pupils and enable them to collaborate. As projects become increasingly complex, pupils will need to

**How could I use this in the classroom?**

When introducing technology, pupils could be shown projects created by role models within the tech community, linked to the subject area and possible careers outlined. In England, this would be particularly beneficial before the age of 14, prior to pupils making examination choices. The

**“ WHEN INTRODUCING TECHNOLOGY, PUPILS COULD BE SHOWN PROJECTS CREATED BY ROLE MODELS**

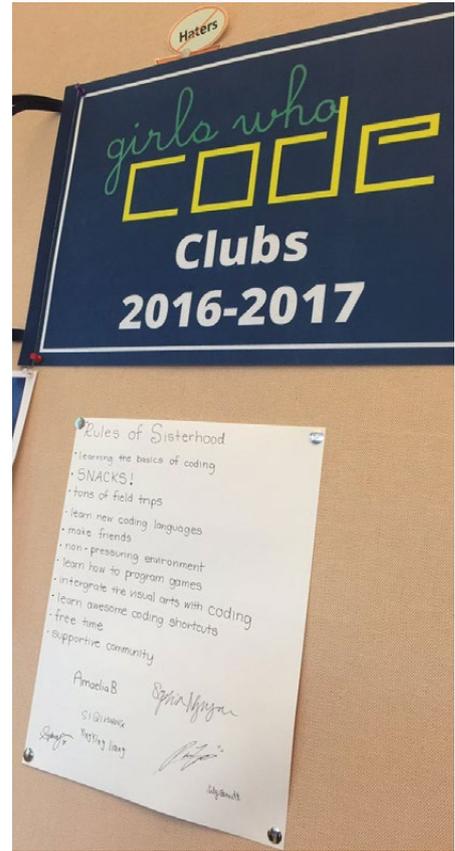
manage a number of deliverables, which could be undertaken as part of a product development methodology, such as Kanban.

**Role models and careers**

Both ECS and GWC regularly link the topics being studied to opportunities in further and higher education, and roles in technology industries. Within the units of work, pupils undertake research on influential members of the tech community, with an emphasis placed on those who make a positive contribution to society through their work. Female role models are regularly included within GWC sessions, and information on further women in technology is always accessible from the Canvas VLE.

focus of research tasks could be related to role models within the industry, with pupils subsequently able to present their findings using a technology of their choice.

Finally, despite the importance of collaborating and building a community within ECS and GWC, both programmes emphasise the need for pupils to work independently on occasions. This is achieved through the provision of session resources on a class website/VLE, which pupils can access without assistance from the instructor. It was evident during observations how motivated and engaged pupils were when working in this manner, which saw the teacher taking on more of a facilitator role during sessions, rather than being the source of knowledge.



■ GWC after-school club in San Francisco's 'Rules of sisterhood'

Have the approaches used in ECS or GWC been beneficial in your classroom? Do get in touch on Twitter @computingchamps and let me know how you get on. (HW)





# READ US ANYWHERE

Get the free app and keep up to date on your favourite Apple or Android device

## DOWNLOAD THE FREE APP **TODAY**



- Instant downloads, every issue
- Fast rendering performance
- Live links and interactivity
- Buy issues to help support the magazine's mission

### AVAILABLE AT:



[www.helloworld.cc](http://www.helloworld.cc)

 @HelloWorld\_Edu

 fb.com/HelloWorldEduMag

## AGE RANGE

9-11 years

## LESSON TYPE

Programming

## REQUIREMENTS

- Python 3
- guizero library  
(easy to install, see the link to the documentation)

# NAME MY KITTEN

Python is a hugely popular language in schools, and adding a user-friendly graphical interface to your Python programs is easier than you think, as Laura Sach demonstrates

**Y**ou just got a new kitten, but you're not sure what to name it. You've managed to narrow it down to... oh... about 36 different names. Why not put

the decision into the hands of the universe and let Python decide? Learn how to easily add buttons, colours, and pictures to make your Python programs more fun.

## Exploring the lesson plan in detail

A program with a GUI behaves differently to programs students may have written so far. In a procedural-style program, the program starts at the beginning and executes statements in order (perhaps with some detours for loops or selection statements)



## THE CHALLENGE

- 🐱 Create a GUI with coloured text, a picture, and a button
- 🐱 Randomly choose a kitten name from a list
- 🐱 Understand and use functions

until the end of the program is reached. However, a GUI program is **event-driven**, which means that the GUI itself runs an event loop which is listening and waiting for **events** (such as clicking a button) to happen. These events could happen multiple times, in any order.

Begin by creating the app, which is the basis for all GUI programs:

```
from guizero import App, Text, PushButton, Picture
import random
```

```
app = App("Name my kitten")
app.display()
```

Save and run this program and you should see a very basic GUI appear!

GUI programs have an **event loop**. The line `app.display()` starts the loop and then waits forever for events to happen, such as buttons being pressed. All of the controls, such as buttons and text boxes, should be placed before the event loop, and then **functions** are created to take care of what happens when an event occurs.

Add some code before `app.display()` to display some text. You can even jazz it up a bit with a different font and colour if you like!

```
display_name = Text(app,
text="Name your
kitten...", font="Arial",
color="red")
```

After this, add a button. When this button is pressed, we will generate a random kitten name and display it on the GUI.

```
name_it = PushButton(app,
command=name_my_kitten,
text="Name it!")
```

We need to write the function `name_my_kitten()`. Between the import statements and the start of the app, write a function:

## ALTERNATIVE ACTIVITY IDEAS

### Explore and make

Adapt this project theme for other age groups:

#### 14-16 years - Programming

- 🐱 Add a GUI to a program you have already written
- 🐱 Create a new program using a GUI (e.g. hangman)

#### 16-18 years - Physical Computing

- 🐱 Use a GUI to control physical objects: turn LEDs on and off, display readings from sensors, or even make a remotely controlled robot

## ASSESSMENT

- 🐱 How could you add more possible name choices?
- 🐱 What does the function `name_my_kitten` do?
- 🐱 How does the GUI program know when to execute this function?

```
def name_my_kitten():
    display_name.set(
        random.choice(
            ["Tiddles", "Bob",
            "Fishy", "Tabitha"] ) )
```

Test your program. When you press the button, the randomly chosen kitten name should be displayed on the GUI.

You can display a GIF image of your kitten, too. You just need to make sure the picture is in the same folder as your Python file, then add the following code before the `app.display()` command:

```
my_kitten = Picture(app,
"kitten.gif")
```

(HW)

## FURTHER READING

guizero documentation

[lawsie.github.io/guizero](https://lawsie.github.io/guizero)



## AGE RANGE

14-18 years

## LESSON TYPE

- Electronics and logic
- Text-based programming

## REQUIREMENTS

- A computer: Linux, Mac OS, or Windows
- Python 3 and IDLE or access to [trinket.io](http://trinket.io)

# INTRODUCING LOGIC GATES

While abstraction is a key concept in Computer Science, and can be very useful, sometimes it can be enlightening for students to understand concepts at a slightly deeper level. Marc Scott explains how logic gates really work...

**B**efore you can really explain to your students what logic gates are and how they work, they'll first have to understand what a transistor is, and the properties of a transistor. To really understand a transistor, they'd need to have quite a thorough understanding of electrical theory, so I've found that a water pipe analogy is the best way to explain how a transistor works. Once students have a firm understanding of how a transistor can act as a simple switch, then it's not too difficult to demonstrate how a NAND gate works.

Hopefully the diagram has helped your students to understand how a NAND gate functions. The key point for them to take away is that the output is only False when both inputs are True. You can then show them the truth table for a NAND gate.

The next step is to show them how to recreate a NAND gate in Python. They can create their own NAND gate in code with this simple function, that uses none of Python's built-in logical operators, although they may come up with their own method.

```
def NAND(A, B):
    if A == False:
        return True
    elif B == False:
        return True
    else:
        return False
```

If they run their program they can call their function and then use the shell to test whether it works. For instance:

```
>>> NAND(True, False)
True
>>> NAND(True, True)
False
```

Next, show them how a NOT gate can be formed by taking a single input and feeding it into both inputs of a NAND gate.

They can then create a NOT function, using the NAND function they've already created.

## ALTERNATIVE ACTIVITY IDEAS

There are other ways to use this style of activity with students, in a variety of contexts.

### 7-10 years - unplugged

- ▶ Have your students role play at being a NAND gate. They can shout 'True' or 'False' depending on the input they receive. By chaining students together, you can produce a variety of logic gates.

### 11-13 years - visual programming

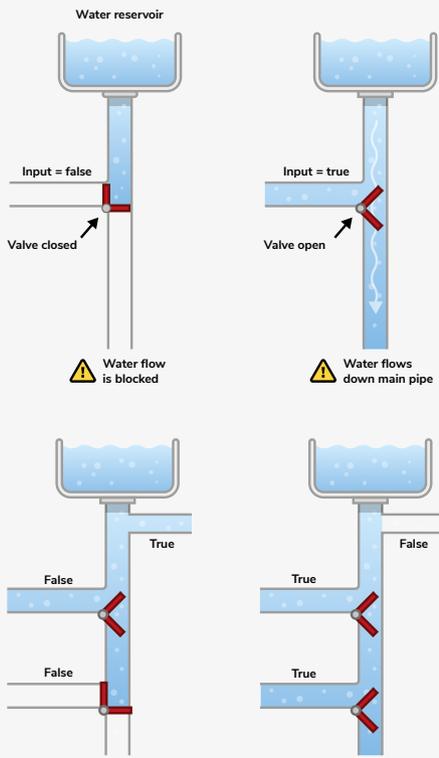
- ▶ The functions detailed in the lesson plan can easily be created in Scratch using the custom blocks feature, or broadcasts, which may be simpler for some students.

### 16 - 18 - physical computing

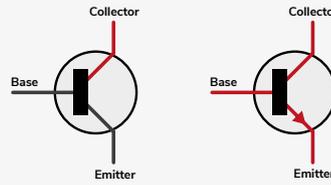
- ▶ If you have access to breadboards and components, you could ask your students to create their own simple logic gates.

## THE CHALLENGE

- Explain to the students how a transistor works, using a water pipe analogy.
- Explain to the students how a NAND gate works using a water pipe analogy.
- Show students the diagram for a NAND gate, explaining the two inputs and the outputs.
- Write a Python function to simulate a NAND gate, and then draw the truth table.
- Demonstrate how NAND gates can be joined together to form other types of logic gates.
- Code and draw truth tables for the NOT, AND, and the OR gate.



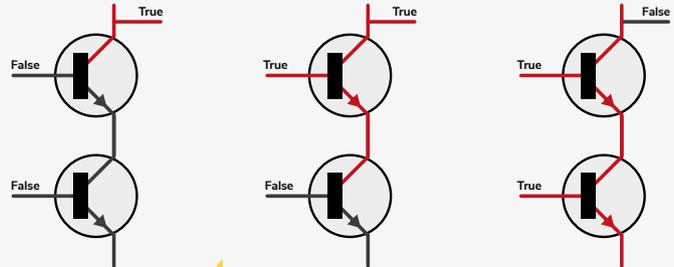
When either of the valves is closed, water cannot flow down the main pipe. Instead, it flows out of the narrow pipe at the top, and the output is therefore **True**. When both valves are open, water can flow down the main pipe, and does not need to flow out of the pipe at the top. The output is therefore **False**.



When a small electric current reaches the Base, current can flow from the Collector to the Emitter.

The truth table for a NAND gate, showing the two inputs and the output.

INPUT A	INPUT B	INPUT C
False	False	True
True	False	True
False	True	True
True	True	False



Transistors arranged to make NAND gates. Only when both inputs are **True**, does the output become **False**.



The symbol for a NAND gate showing the same combinations of Inputs and Output.

```
def NOT(A):
    return NAND(A, A)
```

Trying this one out in the shell will flip the input. For instance:

```
>>> NOT(False)
True
```

Now you can show them a diagram of how an AND gate can be formed from two NAND gates.

An AND gate takes the output from a NAND and feeds it into both the inputs of another NAND. More able students might be able to figure out how to code this themselves, or you can provide them with the code.

```
def AND(A, B):
    C = NAND(A, B)
    return NAND(C, C)
```

By feeding inputs into their defined AND gate, they can then work out the truth table.

```
>>> AND(True, True)
True
```

Here you can see the water pipe analogy for a single transistor and a NAND gate, compared with arrangements of actual transistors. Use this to explain to your students how a NAND gate operates.

Lastly you can show them the more complicated setup for an OR gate.

Again, students might be able to create this themselves, or you could provide them with the code.

```
def OR(A, B):
    C = NAND(A, A)
    D = NAND(B, B)
    return NAND(C, D)
```

Once again, truth tables can be constructed as the students experiment with different inputs and outputs.

```
>>> OR(True, False)
True
```

Once all this has been completed, and the students have worked on the assessment activities, they should have a fairly firm understanding of the key logic gates, and how they function. (HW)

## RESOURCES

High resolution download of the diagram:

[helloworld.cc/2qNGEnl](http://helloworld.cc/2qNGEnl)

Code for the completed logic gates and truth tables:

[helloworld.cc/2puU9V5](http://helloworld.cc/2puU9V5)

## FURTHER READING

NAND Gates: [helloworld.cc/2oPwf6c](http://helloworld.cc/2oPwf6c)

NAND Logic: [helloworld.cc/2plwuqL](http://helloworld.cc/2plwuqL)

## ASSESSMENT

Show students the arrangement of NAND gates to make a NOR gate, and see whether they can code it independently.

Have students draw the truth table for the NOR gate.

Show students the truth table for the XOR gate and see whether they can figure out how to code it.

# SIMULATE LOGIC GATES IN MINECRAFT

Beth McCue shows us how to simulate logic gates in Minecraft using Python

**A**s the excitement of getting a place at Picademy started to wear off, it was replaced with a feeling of fear. Would I be able to come up with anything to do for the project on day two? I'd kept up with the projects from the Cardiff Picademy in February on Twitter and they all looked so creative!

During day one of Picademy, I got my first ever experience of using Minecraft and enjoyed using Python to create items in the world. This helped make my decision for day two: I was going to work outside of my comfort zone and produce something using Minecraft that would challenge me and be beneficial to my students. The aim of my project was to simulate logic gates using the blocks in Minecraft.

The first simulation generates three blocks to represent an AND gate. Here, the first two blocks (looking from left to right) represent the inputs, and the third the output. Stone is used to signify off and gold for on; when the blocks are

generated, they are all stone (**Figure 1**). The two input blocks can be toggled between off and on by hitting them (**Figure 2**) and when both blocks are on, the output changes to on (**Figure 3**).

The second simulation adds a fourth block to represent the gate type, to allow for five different gates to be simulated. Let's call this the 'gate signifier'. Each gate type is represented by using a different material for the signifier block:

AND gate:		
BRICK_BLOCK	ID: 45	
OR gate:		
DIAMOND_BLOCK	ID: 57	
XOR gate:		
LAPIS_LAZULI_BLOCK	ID: 22	
NAND gate:		
WOOD_PLANKS	ID: 5	
NOR gate:		
SNOW_BLOCK	ID: 80	

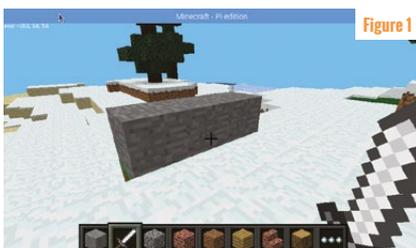


Figure 1

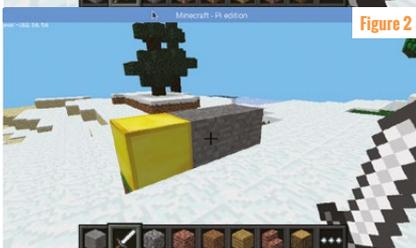


Figure 2

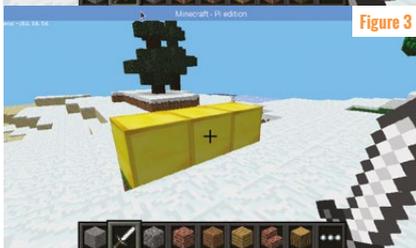


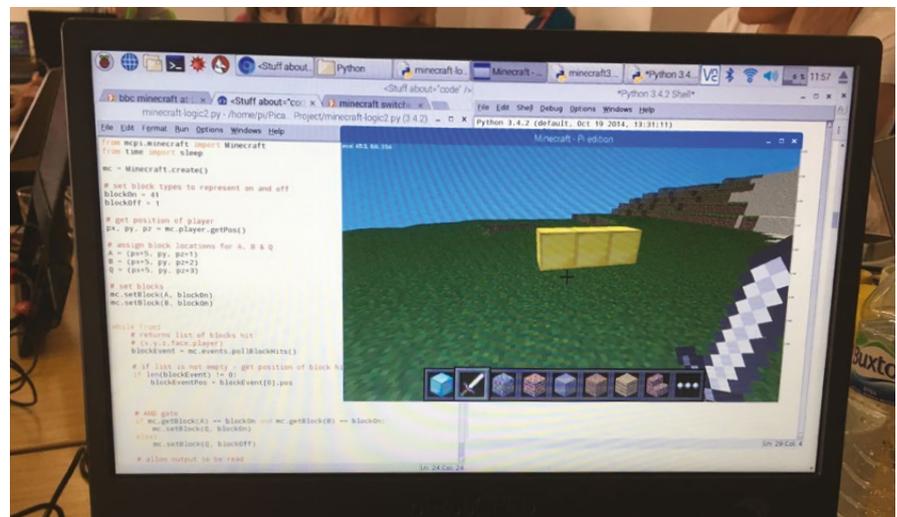
Figure 3



Figure 4



Figure 5





■ Beth learnt all about using Minecraft Pi during one of the Raspberry Pi Foundation's free Picademy courses

The two inputs are toggled between on and off, as before, and the gate signifier block rotates through the five gate types available when it is hit. When the programme starts, a message is displayed to tell the user that stone and gold represent on and off and that the AND gate is currently in use (**Figure 4**). Each time the gate signifier block is toggled, a message is displayed to tell the user which gate is now in use (**Figure 5**).

The final challenge I was given for the day was to see if I could use water and lava to represent wires: water for when there was no current, and lava for when there was.



The first difficulty I had to overcome with this scenario was, as I'm sure my students know, if you place a block of water above ground with nothing around it, it will flood the world. I did this three times before working out what was wrong! The solution was to set the y coordinate for all

## “ THE AIM OF MY PROJECT WAS TO SIMULATE LOGIC GATES USING THE BLOCKS IN MINECRAFT

of the blocks to -1 so that they sat nicely in the ground.

The same blocks were used to represent the inputs, output, and gate signifier, but they were spaced out to allow the 'wires' to be placed in between. This effect really helps to highlight what is happening with the different gates (**Figures 6 and 7**).

There are a number of ways this project could be used in the classroom to engage students and aid their understanding of logic gates. With my A level students, I intend to introduce them to programming in Python to manipulate the world in Minecraft, and then show them just the code for the version with the three blocks. From there, I will challenge them

to improve on it by asking a series of questions such as:

- Can you add a block to represent the gate type?
- Can you make it output messages so the user knows which gate is in use?
- Can you use water and lava to show when there is electricity flowing between the gates?

For working with classes lower down the school, it could be used to demonstrate the different gates. They could be given the final code to use so that they can toggle the blocks to see what happens with each of the different gates. The graphical representation would help them to understand how the gates work. To test students' understanding of logic gates, the messages could be removed and then the students would have to use the program to work out which material is being used to represent which gate. To structure it for them, they could have a worksheet with truth tables to complete for each material, and from there they should be able to match the material to the gate type.

The code for all three versions of the simulation is available to download from GitHub ([helloworld.cc/2oMDRph](https://github.com/helloworldcc/2oMDRph)). If you give it a go, I would love to hear how you get on with it and if you develop it in any way.

So, where do I go from here? The next challenge I have given myself is to represent the gate by using multiple blocks across the end of the wires. Once that's working, I may add more inputs to make the circuit bigger, but I guess that'll have to wait until after the exams in the summer! Keep an eye on Twitter (@SouthSeftonICT) and Instagram (@southseftoncs), or check back on GitHub to see how I get on. (HW)

## AGE RANGE

16-18 years

## LESSON TYPE

Unplugged

## REQUIREMENTS

Pen and paper

# NOUGHTS AND CROSSES AI

How does a computer play noughts and crosses? Investigate basic AI and test your computational thinking skills using only pencil and paper. Laura Sach shows us how...

**M**any people enjoy playing video games. In recent years, technology has developed to allow us to play online against our fellow humans, but most computer games still pit the player's wits against a computer AI. In this

lesson we explore a basic method for developing a program to make decisions, and to play a game. We investigate the familiar game of noughts and crosses, where it is possible to make a computer play so well that it is unbeatable.

Play a critical game of noughts and crosses. Ask two students to play each other, but to explain their strategy at each move out loud as they play.

The students then write down a list of the logical thinking behind each move. This is likely to involve a lot of language along the lines of "if x, then I will do y". The students should aim to devise a written set of rules that a computer should follow in order to play noughts and crosses.

Your students may come up with some of the following strategies:

- Win
- Block the opponent from winning
- Fork (create two ways of potentially winning)
- Block opponent's fork

Is there a priority order in which these rules should be applied? What should the AI do if none of these strategies applies?

## THE CHALLENGE

You are trying to program an AI to play noughts and crosses.

- ✗ What rules should your AI follow to win (or draw) the game?
- ✗ Do any of the rules have priority over the others?
- ✗ How would you represent the rules you have devised to a computer?
- ✗ Is it possible to make your AI too good?



“ AIM TO DEVISE A WRITTEN SET OF RULES THAT A COMPUTER SHOULD FOLLOW

Once the students have devised their rules, ask them to play a game with one player exactly following the rules they have come up with, and the other person playing as they normally would. Are the rules clear and lacking ambiguity? Did the rule-following player win?

**Extension**

Some students may realise that it would be beneficial to look several

moves ahead to see what would happen if you made a particular move, and if the other player made certain moves.

This provides a good opportunity to discuss trees. Each possible move can be represented as a branch of the tree, which itself then branches further depending on the move the opponent makes. The minimax algorithm takes a board scenario and recursively traverses

a tree of potential ways for the game to play out, allocating -1 to board scenarios which result in a loss, 0 to a draw and 1 to a win. Each node of the tree is assigned the lowest value of its child nodes, and the recursion allows the AI to determine which next move will be most likely to result in a win. (HW)

**ALTERNATIVE ACTIVITY IDEAS**

**Explore and make**  
Adapt this project theme for other age groups:

**14-18 years - Programming**

- Build a noughts and crosses game for two human players

**ASSESSMENT**

- Did your AI strategy win? If not, why not?
- How might you represent the game in a real program?
- Is there such thing as an AI that is too good?

**FURTHER READING**

**Flexible Strategy Use in**

**Young Children's Tic-Tac-Toe:** [helloworld.cc/2qtY9Gd](http://helloworld.cc/2qtY9Gd)

**Interactive minimax example:** [helloworld.cc/2oTOW6C](http://helloworld.cc/2oTOW6C)

AGE RANGE

7-10 years

LESSON TYPE

Block-based programming

REQUIREMENTS

Scratch 2.0

# STORY TIME

Rik Cross introduces variables by using them to tell a personalised story

**C**reate a unique story, that is personalised to the person reading it! In this project, variables are used to store answers to questions, which are then used as part of the story.

This project provides a fun first introduction to variables in Scratch, by showing learners how to input

and store numbers and text in their projects. Furthermore, sharing their story with others provides a real audience for their creations, and the feedback will help them to spot problems and improve their story.

### Choosing a theme

Learners should start by deciding on a theme for their story, and add a suitable backdrop and sprite to a new Scratch project.

They can also decide on some questions to ask before telling their story, for example:

- What is your name?
- What is your favourite food?
- What are you most scared of?
- What is the adjective that most describes you?

### Asking a question

Once the questions have been decided, learners can use the Ask block (found in the light blue Sensing section) to ask their questions.

### Storing the answer

Next, learners should create variables to store the answers to the questions. They will need to

## ALTERNATIVE ACTIVITY IDEAS

### Explore and make

This project can be adapted and extended:

#### 5-6 years: unplugged

- 👤 Learners create a story template, which they give to their friends to fill in the blanks

#### 7-10 years: programming

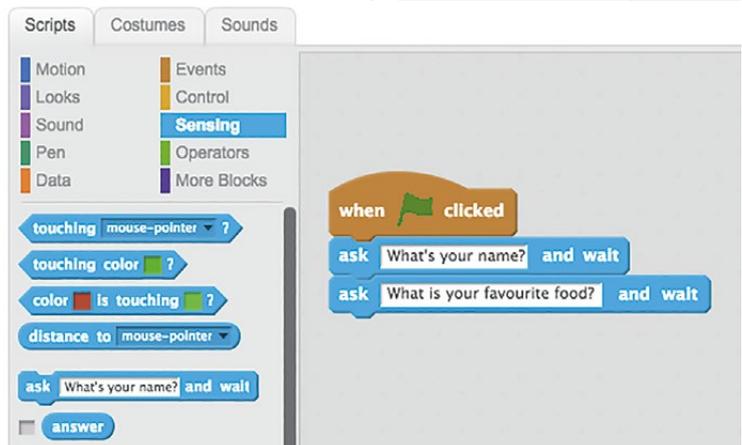
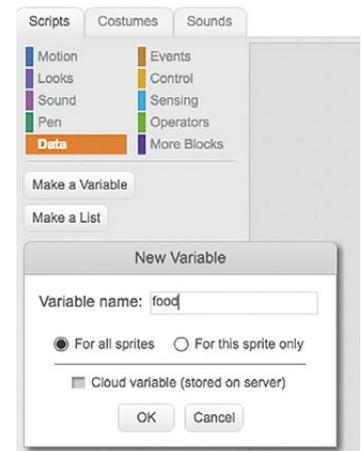
- 👤 Learners can use random list items to add even more personalisation to their story
- 👤 Learners can use the data stored in variables to take different paths through their story
- 👤 Learners can add sounds and special effects to add suspense to their story

#### 7-10 years: collaborative programming

- 👤 Learners can create a story together, with each learner being responsible their own character or part of the story

#### 11-13 years: text-based programming

- 👤 Learners can use a text-based programming language such as Python to create their story
- 👤 Learners could make use of lists and dictionaries to create a more complex storyline

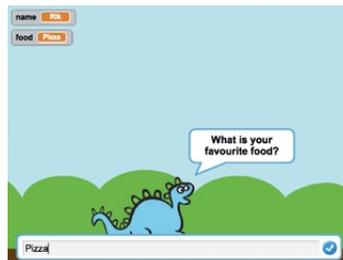


## THE CHALLENGE

- Plan a story, with spaces for the user's input
- Use the Scratch Ask block to ask questions
- Store the answers to your questions in variables
- Use your variables, along with Say blocks, to tell your story

create one variable for each of the questions they have asked. Variables can be created by clicking the red Data section, and then clicking Make a Variable. Remind learners that they should choose sensible variable names, so that they can remember what is being stored where!

Learners should see their variables appear at the top right of the stage. Now that learners have places to



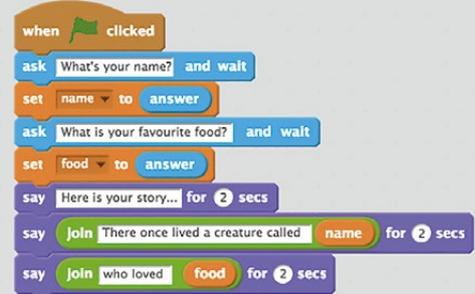
store the answers to their questions, they can add code to store the answers given.

At this point, learners can test their code by clicking the green flag and answering the questions when prompted. They should see that the answers given to each question are stored in the variables.

- Ask the user questions, and store their answers in variables. This data can then be used to create a personalised story.



Join There once lived a creature called name



## Telling your Story

Now learners will have everything they need to tell their story. To add their variable contents to their story, learners will first need to use a Join block, which they can find in the green Operators section.

They can place their Join block inside a 'say... for 2 secs' block, so that their sprite tells each part of their story.

Encourage learners to ask others to test out their story, and to use their feedback to make improvements. If learners want to hide their variables on the stage, they can deselect them in the red Data section. (HW)

## FURTHER READING

See the completed project: [helloworld.cc/2pBuTNN](http://helloworld.cc/2pBuTNN)

'Teaching Programming in Primary

Schools' online course: [helloworld.cc/2pE8XDw](http://helloworld.cc/2pE8XDw)

'Ghostbusters' Code Club

project introducing variables: [helloworld.cc/2oTAIJ5](http://helloworld.cc/2oTAIJ5)

## ASSESSMENT

- What other variables did you need to tell your story?
- How do variables allow you to personalise your story?
- Did you have any problems with telling your story?



# DRAWING WITH PYTHON

Get creative with code and help your students reinforce what they've learnt with immediate visual feedback

In this feature, we're going to look at drawing and colouring various shapes in a window. This isn't quite modern art, admittedly, but drawing shapes is the first step in building just about anything. We're going to assume that you and your students have written

some simple Python (or similar) programs in the past, and are able to create files and navigate your computer's file system without too much difficulty. If you haven't set up your computer with Python and PyGame, there are lots of easy-to-follow guides on the web which will help bring

you up to speed. See [pygame.org](http://pygame.org) for more information.

To start off, open your preferred text editor. Create a new file, insert the following code into it, and save it as `hello.py`:

```
import pygame

pygame.init()
window = pygame.display.set_mode((500, 400))

while True:
    pygame.draw.rect(window,
                     (255,0,0), (0, 0, 50, 30))
    pygame.display.update()
```

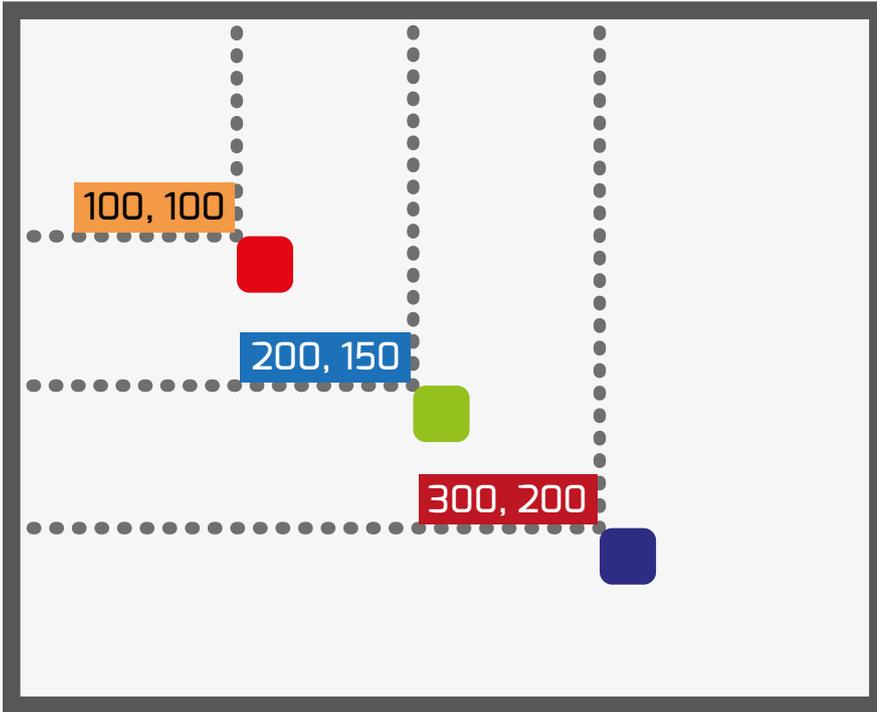
Let's run that code and see what it does. In your terminal window, enter `python hello.py`. If all has gone well, a new window will open, showing you a red square on a black background in the top-left corner of the window. We've just created our first Pygame program: let's walk through it.

## Understanding hello.py

The first two lines of our first program are very simple: all we've done is told Python that we want to use Pygame. `import pygame` loads all of the Pygame code into our script, so we don't have to write all of that code ourselves. Pygame is designed to make the creation of games and interactive

```
window = pygame.display.set_mode((500,400))
```





```
pygame.display.rect(window, (255,0,0), (100,100,50,50))
pygame.display.rect(window, (0,255,0), (200,150,50,50))
pygame.display.rect(window, (0,0,255), (300,200,50,50))
```

software easy. `pygame.init()` tells Pygame that we're ready to start using it. Let's look at the third line of code:

```
window = pygame.display.set_mode((500, 400))
```

`window` is the parameter we're going to use to tell our Pygame program about how it should look when it runs; each parameter affects the application window's shape and size. Note that here, width always comes before height. `window` is also the parameter that we'll use to tell other lines of code the surface on which they should draw shapes and set colours. With `window`, we're calling the `set_mode` function of Pygame's `display` module: the latter is responsible for how the game window and surface (an informal term for the pixels we'll be manipulating) behaves. We're passing a tuple (which we can think of as a special list of things – in this case, a list of numbers) to `set_mode()` to tell it how big we want our game window to be. In this case, the application window is 500 pixels wide by

400 pixels tall. If we pass numbers that are bigger, the game window will be bigger; if we pass numbers that are smaller, the game window will be smaller.

The next few lines are where we make our program draw shapes on that window. When programs run, they execute their code, and when they're finished, they close themselves. That's fine unless, of course,

**“ THE WHILE LOOP WILL NEVER FINISH BECAUSE TRUE IS ALWAYS TRUE, SO WE CAN KEEP RUNNING OUR PROGRAM ”**

you want your program to be interactive, or to draw or animate shapes over time, which is exactly what we need from a game. So, in order to keep our program from exiting, we make a `while` loop and put all our code inside. The `while` loop will never finish because `True` is always `True`, so we can keep running our program and drawing our shapes for as long as we like.

The first thing we do in our `while` loop is draw a rectangle. A rectangle is the simplest shape that we can draw in Pygame:

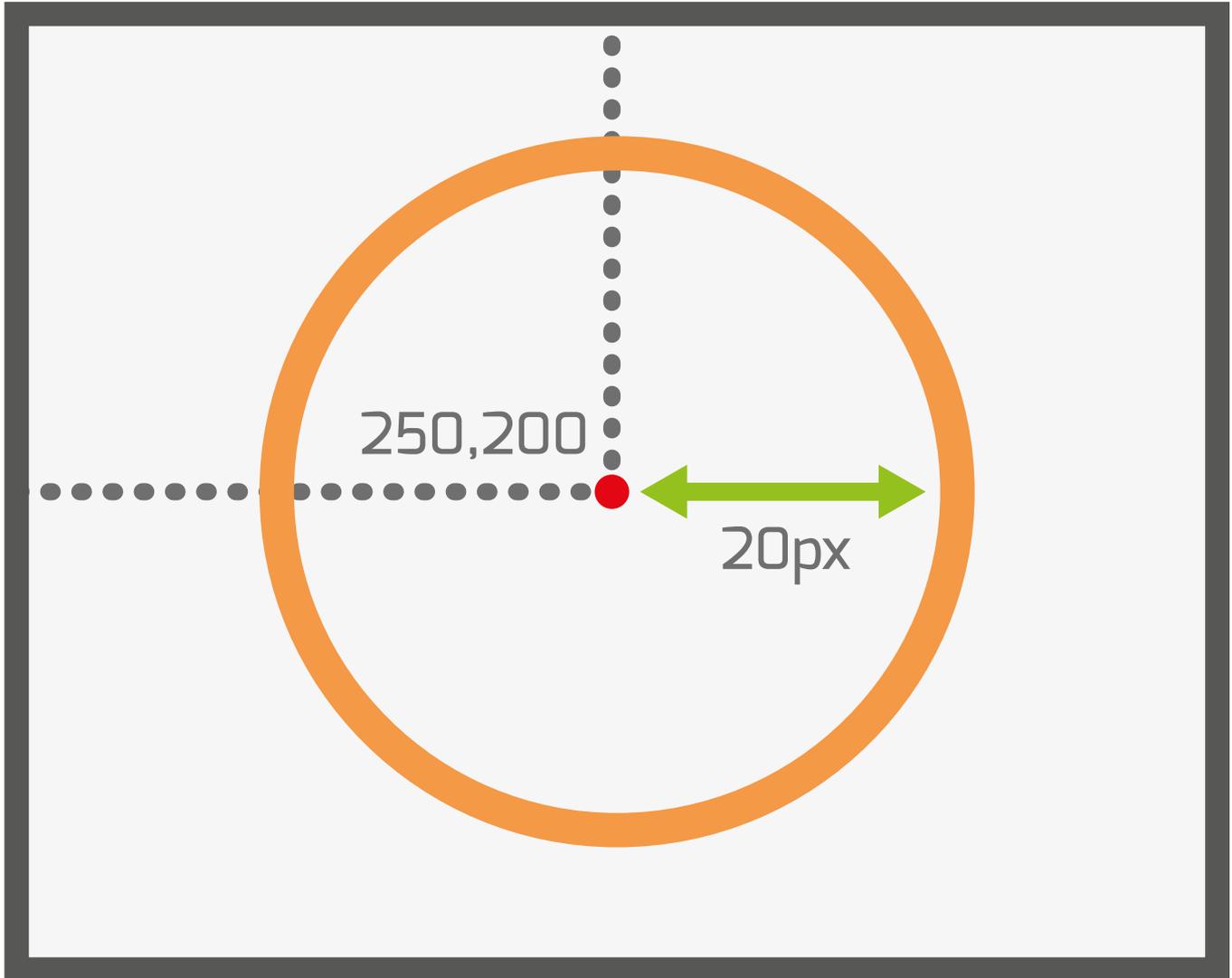
```
pygame.draw.rect(window, (255,0,0), (0,0,50,30))
```

The parameters at the end are telling Pygame where we want to draw our rectangle, the colour we want our rectangle to be, how we want to draw it, and how big we want it to be.

In our `hello.py` program, we've told Pygame to draw a rectangle in our window – or, at least, on the surface we create with our `window` parameter. Next, we told Pygame what colour we wanted our rectangle to be by passing it through a tuple (a special list of numbers) representing how much red, green, and blue the final colour should have in it. We use red, green, and blue as these are the three colours your screen combines to create every shade you can see on it. A value of 0 means that none of that colour should be used in the shape; 255 means that the maximum amount of colour should be in that shape. We told our rectangle that it should be the colour (255, 0, 0), which is pure red. If we had told it to be (255, 0, 255), it would have been a bright purple, because it's being drawn with the maximum amount of red and the maximum amount of blue. If we had told our rectangle to be coloured (100, 100, 100), it would be a dark grey, because all the colours would be equal.

After we've passed through a colour for our rectangle to be, we have to tell it where it should go and how big it should be. We do this by passing a tuple of four numbers. The first number is an X coordinate, which sets out how far from the left side of the window the left edge of our rectangle should be. The second number is a Y coordinate; this tells the rectangle ▶

```
pygame.draw.circle(window, (255,255,0), (250,200), 20, 1)
```



- ▶ how far from the top of our window the top edge it should sit. The third number gives the width of our rectangle, and the fourth number defines its height. So, for example, if we wanted our rectangle to be 50 pixels from the left side of the window, 100 pixels from the top of our window, 20 pixels wide and 80 pixels tall, we would pass `(50, 100, 20, 80)` to `pygame.draw.rect()`.

Please note that the order never changes. If you tell Pygame how big you want the rectangle to be when it's expecting a colour or vice versa, the program may crash, so take your time.

Our last line in `hello.py` is nice and simple: it tells Pygame that we're done drawing shapes for the moment, and that it can now refresh the window. This saves our computer from having to draw and redraw the screen for every shape that we've created. Instead, it can get them all drawn in one go.

### Adding more shapes

We've successfully drawn one shape, so let's draw a few more. We'll draw some squares around the screen and mess around with their properties a bit. There's

no need to create a new file: we'll stick with `hello.py` for now. Edit the `while` loop so it matches the following:

```
while True:
```

```
    pygame.draw.rect(window,
(255,0,0), (100, 100, 50, 50))
    pygame.draw.rect(window,
(0,255,0), (150, 100, 50, 50))
    pygame.draw.rect(window,
(0,0,255), (200, 100, 50, 50))
```

```
    pygame.display.update()
```

Now we should have three squares: red, blue, and green. So far, this is nice and simple, but those squares are placed right next to each other. What would happen if they were to overlap? Let's find out. Change your code once more to the following:

```
while True:
```

```
    pygame.draw.rect(window,
(255,0,0), (0, 0, 50, 50))
    pygame.draw.rect(window,
(0,255,0), (40, 0, 50, 50))
    pygame.draw.rect(window,
(0,0,255), (80, 0, 50, 50))
```

```
    pygame.display.update()
```

second square so that it becomes the third square drawn, like so:

```
while True:
```

```
    pygame.draw.rect(window,
(255,0,0), (0, 0, 50, 50))
    #pygame.draw.rect(window,
(0,255,0), #(40, 0, 50, 50))
FROM HERE
    pygame.draw.rect(window,
(0,0,255), (80, 0, 50, 50))
    pygame.draw.rect(window,
(0,255,0), (40, 0, 50, 50))
#TO HERE
```

```
    pygame.display.update()
```

“

## JUST LIKE DRAWING A RECTANGLE, WE TELL PYGAME ON WHICH SURFACE WE WANT TO DRAW OUR CIRCLE

This time we get two rectangles and a square, but that is not what we asked for. So, what has gone wrong? When we execute our code, it works through what it has to draw, and where it has to put it, line by line. If one item is drawn and then another is drawn over it or on top of part of it, then we can no longer see what is beneath that second shape. The pixels of the shape drawn first are lost when we overlap it with another shape. If we change the order of our code, we can see this effect in action. Cut and paste the code for the

Now the code apparently produces rectangle, square, rectangle. This is because the red and blue squares were drawn first, and then the green square was drawn over the top of them. The red and blue squares are still there in their entirety, but we can't see all of them, so they look like rectangles.

Pygame allows us to do a great deal more than simply drawing rectangles. We can make all kinds of other shapes too, including circles, ellipses, and paths (which are made up of many lines between multiple points).

## Drawing circles

The process of drawing a circle is much like drawing a square, except that instead of passing a width and a height, we pass a radius and a point, around which we draw our circle. For example, if we wanted to draw a yellow circle in the middle of our window with a diameter of 40 pixels, we would use the following code to replace the code in the original `while` loop in `hello.py`:

```
while True:
```

```
    #Just like before to help
us remember
    #pygame.draw.circle(WHERE
TO DRAW, (RED, GREEN, BLUE),
(X COORDINATE, Y COORDINATE),
RADIUS, HEIGHT, WIDTH)
```

```
    pygame.draw.circle(window,
(255,255,0), (250, 200), 20, 0)
```

```
    pygame.display.update()
```

Just like drawing a rectangle, we tell Pygame on which surface we want to draw our circle, what colour we want it to be, and where it should go. The radius is specific to drawing this particular shape. You might have noticed that we put a 0 after our radius; this is a value used to determine the width of the line that draws our circle. If we pass 0, the circle is filled; but if we pass 2, for instance, we get a 2-pixel-wide line with an empty centre:

```
while True:
```

```
    #Filled
    pygame.draw.
circle(window,(255,255,0),
(200, 200), 20, 0)
```

```
    #Not filled
    pygame.draw.
circle(window,(255,255,0),
(300, 200), 20, 2)
```

```
    pygame.display.update()
```

What about ellipses? They are a slightly strange cross between drawing rectangles >





▶ and drawing circles. As we did when we drew a rectangle, we pass an X coordinate, a Y coordinate, a width, and a height, but we end up with an elliptical shape. Let's draw an ellipse or two.

```
while True:
```

```
    pygame.draw.
    ellipse(window, (255, 0, 0),
            (100, 100, 100, 50))
    pygame.draw.
    ellipse(window, (0, 255, 0),
            (100, 150, 80, 40))
    pygame.draw.
    ellipse(window, (0, 0, 255),
            (100, 190, 60, 30))
```

```
    pygame.display.update()
```

Just as before, run your code. You should now see three ellipses: one red, one green, and one blue. Each should be a different size. If you want to visualise how these shapes were generated, you could draw a rectangle using the same coordinates you used to draw an ellipse and it would fit perfectly inside that box. As you may have guessed, this means you can also make circles by using `pygame.draw.ellipse`, if you keep the width and height parameters the same.

```
while True:
```

```
    pygame.draw.rect(window,
                    (255, 0, 0), (100, 100, 100,
                    50), 2)
    pygame.draw.
    ellipse(window, (255, 0, 0),
            (100, 100, 100, 50))
```

```
    pygame.draw.rect(window,
                    (0, 255, 0), (100, 150, 80,
                    40), 2)
    pygame.draw.
    ellipse(window, (0, 255, 0),
            (100, 150, 80, 40))
```

```
    pygame.draw.rect(window,
                    (0, 0, 255), (100, 190, 60,
                    30), 2)
    pygame.draw.
    ellipse(window, (0, 0, 255),
            (100, 190, 60, 30))
```

```
#Circle
pygame.draw.
ellipse(window, (0, 0, 255),
        (100, 250, 40, 40))
```

```
    pygame.display.update()
```

## A new path

We have covered rectangles, squares and circles, but what if we want to draw a triangle, a pentagon, a hexagon, or an octagon? Unfortunately, there aren't functions for every kind of shape, but we can use paths. Paths allow us to draw irregular shapes by defining points in space, joining them up with lines, and filling in the space we've created. This is a little more complex, so it's time to move on from our original `hello.py` program. Create a new file, call it `paths.py`, and save it with the following text inside:

```
import pygame
```

```
pygame.init()
window = pygame.display.set_
mode((500, 400))
```

```
while True:
    pygame.display.update()
```

This is simply our bare-bones Pygame app again. If you want to make a copy of this for experimenting without breaking anything, now would be a good time to do so.

Every path is made of connected lines, but before we start joining things up, let's draw a couple of standalone lines to familiarise ourselves with them. We can do this with `pygame.draw.line()`. Edit `paths.py` so your `while` loop reads as follows:

```
while True:
```

```
    pygame.draw.line(window,
                    (255,255,255), (0, 0), (500,
                    400), 1)
```

```
    pygame.display.update()
```

If you run this code now, you'll see a one pixel wide white line going from the top left to the bottom right of our Pygame window. The parameters we pass to `pygame.draw.line` start off the same way rectangles and ellipses do. We first tell Pygame where we want to draw the shape and then we choose a colour. Now, things change a little. The next argument is a tuple with the X and Y coordinates for where we want our line to start, and the third argument is a tuple with the X and Y coordinates for where we want our line to end. These are the two points between which our line will be drawn. The final argument is the width of the line is being drawn in pixels.

With this, we can now create shapes by defining points in our window. Let's draw that triangle we talked about earlier:

```
while True:
```

```
    pygame.draw.line(window,
                    (255,255,255), (50, 50), (75,
                    75), True)
    pygame.draw.line(window,
                    (255,255,255), (75, 75), (25,
                    75), True)
    pygame.draw.line(window,
                    (255,255,255), (25, 75), (50,
                    50), True)
```

```
    pygame.display.update()
```

You should have an image of a white triangle with a one-pixel edge. However, this code is rather lengthy. So many things, like the colour or the width of the line, are written multiple times. There is a more concise way to achieve the result we want. All we need is `pygame.draw.lines()`. Whereas `pygame.draw.line()` lets us draw a line between two points, `pygame.draw.lines()` enables us to draw a sequence of lines between numerous points. Each XY-coordinate point will be joined up to the next XY-coordinate point, which will be joined up to the next XY-coordinate point, and so on.

After running the code, you'll see that the resulting triangle is exactly the same, except that we produced it from one line of code instead of three. You might have noticed that we didn't actually close the triangle: Pygame did it for us. Just before we pass the points for our shape to be drawn from, we can pass either a `True` or a `False` value that will let Pygame know that we want it to close our shapes for us. Change it to `False` and we get the first two lines of our shape, but not the third.

If we want to make a more complex shape, we simply add more points, like so:

```
while True:
```

```
    #pygame.draw.lines(WHERE  
    TO DRAW, COLOUR, CLOSE THE  
    SHAPE FOR US?, THE POINTS TO  
    DRAW, LINE WIDTH)
```

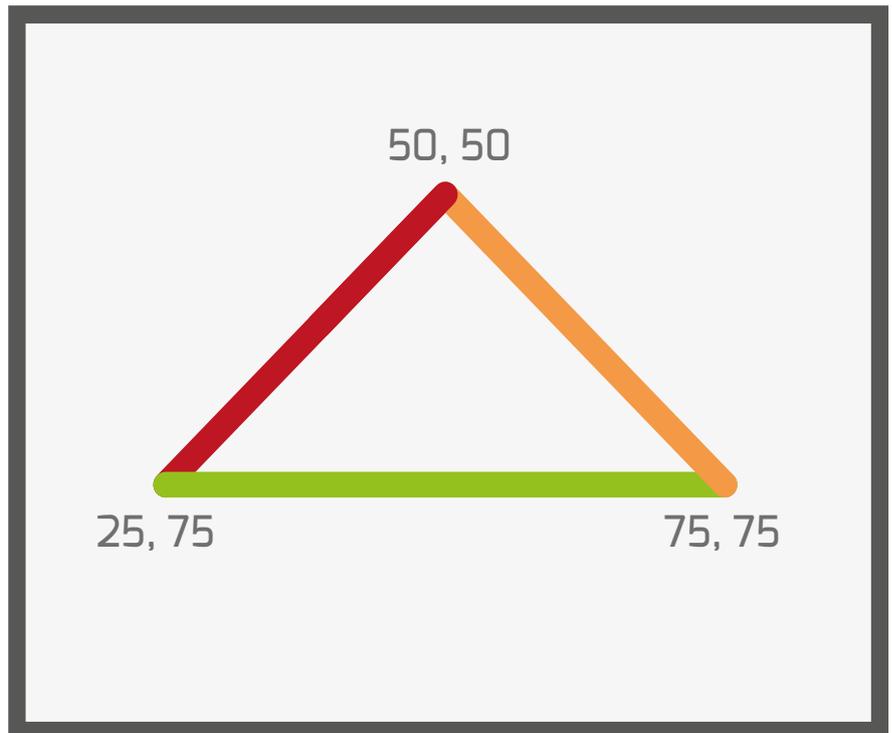
```
    pygame.draw.  
    lines(window,(255,255,255),  
    True, ((50, 50), (75, 75),  
    (63, 100), (38, 100), (25,  
    75)), 1)
```

```
    pygame.display.update()
```

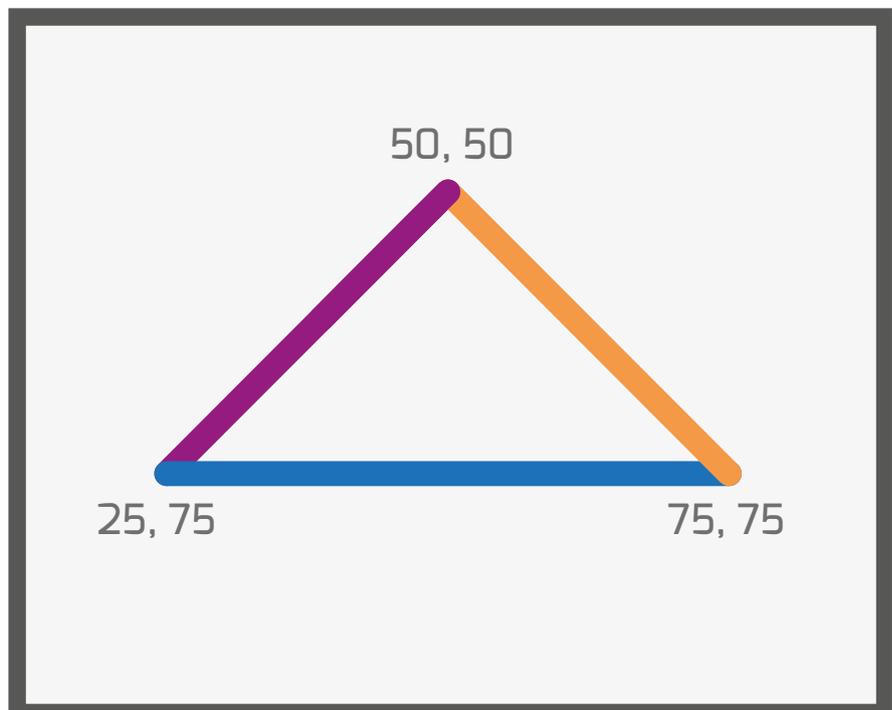
There you have it: your very own pentagon. If you want to make a hexagon, an octagon, or even a triacontagon, just add more points – it's that easy!

Why not try experimenting with Python with your students to produce some interesting pixel art? [\(HW\)](#)

```
pygame.draw.line(window,(255,255,255),(50,50),(75,75),1)  
pygame.draw.line(window,(255,255,255),(75,75),(25,75),1)  
pygame.draw.line(window,(255,255,255),(25,75),(50,50),1)
```



```
pygame.draw.lines(window,(255,255,255),True,((50,50),(75,75),(25,75)),1)
```



# FUNCTIONAL PROGRAMMING IN SNAP!

Functional programming is creeping into a lot of languages: here's an easy way to get to grips with FP without complicated syntax

STORY BY John Stout

The article on Snap! in Hello World's first Issue introduced the basic concept of functional programming (FP):

allowing procedures (either built-in or user-defined ones) to be used in the same way as other data, as first-class items

- map applies a function to every element of a list and returns a list of the results
- filter uses a function to decide whether to include each item of a list in the result and
- fold uses a function to fold up or reduce the list to a single value, e.g. its sum, product, largest, smallest

Looking at the three definitions you can see the same basic structure:

```
if list is empty then
  return a value:
  an empty list for map/filter,
  result so far for fold (the
  base case)
```

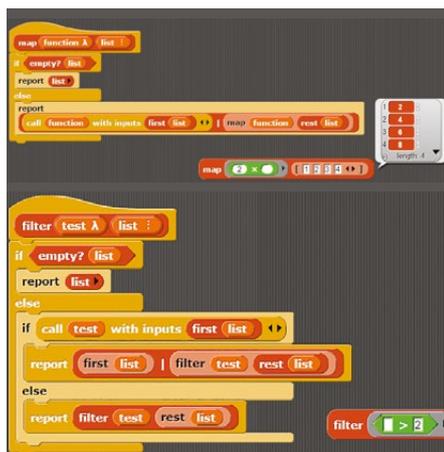
```
else
  return 'something involving
  the first item in list and
  map/filter/fold of the
  function on the rest of list'
  (the recursion)
```

These examples of recursion are relatively straightforward, and students can be introduced to them without going into how they are implemented or how 'scary' they look (I prefer to think of it as 'magical'). In my experience students

The examples in this article (and more) are included in a shared Snap! project at [helloworld.cc/2r0yQLb](http://helloworld.cc/2r0yQLb).

often use recursion for themselves once you introduce procedures to get the effect of repetition.

It's worthwhile now thinking about all the bits of programming you've written and seeing how many of them are examples of map/filter/fold, with the common framework (repeat over every item in the list doing something to or with each one) being abstracted away into the higher level function. More of your programs could be written using functional compositions of these three functions, e.g. to produce the sums 1, 1+2, 1+2+3, ..., and find the first one that's even and over 1000 is a filter of a map of a fold of a map (see the 'Example' sprite's scripts).



Three core FP functions: map, filter, fold, from the example project

## More important FP concepts

In a non-FP programming language, omitting one or more parameters to a function will cause an error. Snap! will just use a default value of 0 or an empty string.

However, FP languages call this *partial application* and instead of giving an error, or using a default value for the missing parameters, will instead return a partially applied function, i.e. one that has only some of its parameters filled in.

To do this with Snap! we *ringify* the function. Take any block in Snap! (this example uses the `+` block) and fill in one of

■ Partial application of the `+` function (operator)



its empty slots (it doesn't matter which) with a number, e.g. 4. Double clicking this gives 4 as a result since Snap! uses 0 for the missing slot. Now right-click on the block and select *ringify* from the menu to surround the block with a 'protective' grey ring. Double-clicking now shows that you've got a ringed function with one argument filled in.

The only problem now is actually filling in the slot, and this is where the `call` block in the `Control` category comes in, taking



■ Find the smallest even number that's a sum of the positive integers, and over 1000: you can test each part on its own from the inside out

specialised functions from more general ones, e.g. in the example shown we've produced a function that adds 4 to its input from the general `+` operator.

In the FP project you'll see there's a function called `+` (not to be confused with

Some languages are guaranteed to optimise tail calls, and use tail calls as a way to implement iteration: Scheme (remember that Snap! is Scheme disguised as Scratch) and Snap! both do this.

## So why use FP?

Inevitably this is a personal thing but I find:

- It's more interesting. I don't have to keep on writing the same boring bits of support code again and again, just concentrate on the important bits and how I combine the functions I write.
- I can test each function independently and easily (see the `filter/fold/map/map` example).
- I write less code to accomplish the same task (although convincing students that less is better can be hard!).
- There's a whole class of errors I just don't write any more (off-by-one errors mainly).

The syntax of FP languages can be a bit daunting, but with Snap! syntax has a concrete form, and there are visible clues as to what's going on. For example, in partial application an input that doesn't have a value shows as an empty slot, rather than just not being there. (HW)

## THE SYNTAX OF FP LANGUAGES CAN BE A BIT DAUNTING, BUT WITH SNAP! SYNTAX HAS A CONCRETE FORM

a partially applied function and supplying the missing input(s). Here there's only one empty slot and one input, so there's no possible confusion.

The *Snap! Reference Manual* has the rules about how missing slots are filled with inputs, but these can be overridden by giving names to the inputs using the black arrowhead on the ringed function. By default the names are `#1`, `#2` etc. but you can change these: for example a quadratic expression function will need the `x` input twice (see the 'Ringify' sprite in the sample project).

One reason you might want to use partial evaluation is to be able to define

Snap! doesn't have lazy evaluation as standard but is powerful enough to implement something very similar called 'Streams' (see the 'Primes' sprite in the example project).

Most of the functions shown are defined using recursion, and it's worth mentioning that most of them use what is called 'tail recursion'. Tail recursion, where the recursive call is the last thing before returning, can be optimised so that it is (almost) as fast as simple iteration; doesn't have the overhead of creating new stack frames to hold parameters and return addresses; and will never cause a stack overflow.

## RECENT SNAP! ENHANCEMENTS

After a drag and drop to a pane, an 'undo' and a 'redo' control (once something's been undone) appear in that pane and can be used to 'time travel' through your edits.

Ctrl-F (or shift click) can now translate a typed arithmetical expression into blocks, using parentheses to override operator precedence.

■ The undo/redo controls and typing an expression (variables will still need defining)

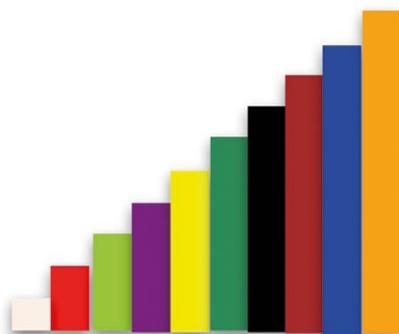


# USING HASKELL WITH 5- TO 7-YEAR-OLDS

Ian Benson and Jenny Cane show how Haskell helps learners to reason symbolically with Cuisenaire-Gattegno Mathematics: a proven approach to teaching algebra before arithmetic

STORY BY Ian Benson and Jenny Cane

**A**t Bursted Wood Primary School, children in key stage 1 had already embarked on a secure mathematical journey, putting algebra at the heart of the curriculum. The demands of the 2014 National Curriculum meant that in Year 1 children were expected to be taught to use all four arithmetic operations, and fractions, as operators. As a result, Bursted Wood adopted a series of textbooks written by Caleb Gattegno called *Numbers in Colours*. These textbooks use Cuisenaire rods to encourage the learner to explore patterns, and then to give letter code names to the patterns, and record the relationships between them with equations. It is only after the letter code exploration that they finally use a rod as a unit of measure (e.g. white representing one), and transform their equations into arithmetic. Very early on in Year 1, children are able to read and write a variety of complex algebraic



■ The ten Cuisenaire rods arranged in order as a staircase with step size White.

equations using plus, minus, multiply, and fractions as operators, and translate these into arithmetic. The results had been impressive throughout key stage 1. In Year 2 we wanted to take it a step further by running computing lessons using the same use of natural language and level of sophistication.

## Why Haskell?

Haskell is closely aligned to conceptual mathematics and number theory, the inspiration for Gattegno Mathematics. It has four attributes that make it particularly appropriate for mathematics education:

- Its syntax is very close to traditional mathematical notation (but beware that it can be too terse).
- Variables are statically typed, which means that programmers are protected by the compiler from passing invalid information as an argument (input) or output value.
- The Haskell interpreter evaluates functions 'lazily', only when required. This makes it possible to program infinite data structures in finite memory.
- Information is passed from the arguments to functions by pattern matching. This is an intuitive way for programmers to envisage and reason about the behaviour of their programs.

## 19. Are the following true

$$o = 10w = 5r = 2y$$

$$B = 9w = 3g$$

$$t = 4r = 2p = 8w$$

$$b = 7w$$

$$d = 6w = 3r = 2g$$

$$y = 5w$$

$$p = 4w = 2r$$

$$g = 3w$$

$$r = 2w$$

■ Equivalent expressions using Gattegno's rod codes to represent the relationship between a rod and equivalent trains of one colour

## Starting with a staircase

To start, we went back to a familiar pattern that Gattegno introduces in the textbooks, called a staircase. This is built from one rod of each colour arranged in size order (figure 1). The children were able to physically build the staircase. However, because they were so familiar with the pattern, they could also access it virtually, which would help with debugging and logical reasoning later.

A role-play game to establish computational thinking was introduced before any actual programming began. This allowed the children to familiarise themselves with the function calls `successor` and `predecessor` (`succ`, `pred`) while traversing the staircase. The children would take turns to ask each other questions such as `succ White` (answer Red) or `pred Orange` (answer Blue). To help the children access the new programming vocabulary, one child

```
data Colour = NaC | Red | Purple | DarkGreen | Brown |
Orange deriving (Show, Ord, Eq, Enum)
```

```
w=2
g = 3*w
p = 2*r
r=2*w
y = 5*w
d = 6*w
k = 7*w
t = 8*w
n =9*w
o = 10*w
```

■ A rod arithmetic calculator built after studying the staircase step size White, and its conjugate with every other rod removed

would take the role of the 'engine' and the other 'interpreter'. This was extended to `succ $ succ` (successor of successor) which uses function composition `$`. Some children could creatively enjoy inventing functions such as `succ $ succ $ succ White` to test the technique. The children experimented by writing down their calls before building themselves a rod calculator in Haskell. The language soon became a natural part of lessons and meant that it was easy to progress to the next step.

run, and that this would require carefully debugging and retyping. The children could spot bugs in the engine because they had already worked out the answer. The staircase was a familiar pattern, so they could predict what the output should be.

Children were also able to edit the `Colour` data type to model odd or even staircases and test their functions to see what the effect would be on the derived `successor` function. A transcript (see insert) shows a pupil working with the

## “ SOME CHILDREN COULD CREATIVELY ENJOY INVENTING FUNCTIONS

### Programming

Very soon the children were able to construct their own rod calculator in Haskell, by first constructing the engine using the `Colour` data type. Then the children asked the same role-play questions, this time by typing the language into the interpreter window. It soon became apparent that errors in the engine meant that the interpreter would not

number names for the rods, measured with White rather than the colour names. He becomes aware as he is speaking that the version of the `succ` function that works for a staircase with a step size White is distinct from the version of `succ` in an even staircase, and he corrects himself. It is clear to see that he has constructed an image of the staircase and is virtually traversing it.

■ Transcript of a conversation about program behaviour

```
13:08 Q: So what are you doing? Are you doing |succ| or the other one?
19:08 A: I am doing |succ|
21:11 Q: What does that mean?
24:03 A: It means that if I do |succ|, you add one number on ... If I had six
and I did |succ| it equals (pause for reflection) 7
37:00 Q: You're right. So you do it ....
40:07 A: (speaks over Q) Actually 8 because there are no odd.
46:03 Q: Ah. You are just doing the evens.
48:06 Q: Right, Brilliant
```

## WHERE TO GO NEXT

Many Cuisenaire and Gattegno resources can be found online:

- New developments in arithmetic teaching in Britain: introducing the concept of 'set' [helloworld.cc/2pYFg0w](http://helloworld.cc/2pYFg0w)
- The common sense of teaching mathematics: [helloworld.cc/2p9dyKO](http://helloworld.cc/2p9dyKO)
- Gattegno Mathematics, Book 1: [helloworld.cc/2qz7xZe](http://helloworld.cc/2qz7xZe)
- Virtual Rods: [helloworld.cc/2pl8ctc](http://helloworld.cc/2pl8ctc)
- Sample code for these exercises: [helloworld.cc/2pFyx9j](http://helloworld.cc/2pFyx9j)

### Calculating

To build the rod arithmetic calculator, the pupil reuses the equations that they studied in Year 1. Gattegno's definitions lie in the algebraic domain. Programming illustrates the distinction between algebra and arithmetic. Making this clear is crucial to investigating the level of abstraction that the learner has reached. The figure shows a rod arithmetic calculator constructed after four terms of Gattegno Mathematics. The values associated with the rod code letters are modelled as constant functions in Haskell. Nine equations are derived from Gattegno's equivalent expressions. The tenth, which binds an integer value to the code for White, shows the choice of measuring unit, which fixes the values of the other codes.

By introducing Haskell to the children we have been able to take the mathematics that the children are familiar with and successfully and creatively introduce programming functions. We have been able to provide both the teacher and child with a rich situation, allowing them to think both mathematically and computationally. It is exciting to think where our investigations may lead us to next! (HW)

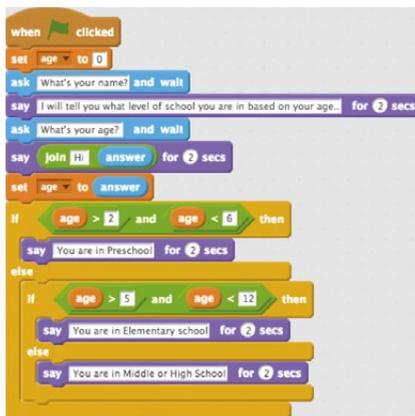
# TACKLING NOVICE LEARNERS' NAIVE CONCEPTIONS IN INTRODUCTORY PROGRAMMING

If you're a teacher of introductory programming at any level, chances are that you will notice several common patterns of problems exhibited by students during their early programming activities

STORY BY Shuchi Grover, PhD

**T**he naive conceptions of novice learners result in frequently observed sources of confusion across age levels and regions. While some aspects of novice difficulties studied over the decades are specific to text-based and object-oriented languages and programming environments (such as the confusion related to '=' operator that stands for assignment in programming and equality in mathematics), several others persist even in the context of commonly used block-based programming environments.

Recent investigations suggest that learner misconceptions endure even when students are using child-friendly block-based interfaces. These graphical programming environments, inspired by



■ Fig. 1. What does the programmer believe will be stored in Scratch's answer variable?

Seymour Papert's 'low floor' (and high/no ceiling) philosophy, are engaging. They afford a much more motivating programming experience than traditional text-based languages do, by allowing early programming experiences to focus on designing and creating, and avoiding issues of programming syntax.

This article describes several naive conceptions and difficulties novice programmers (regardless of age or grade level) exhibit in introductory programming settings.

## The superbug

First, we hark back to the 1980s work of Roy Pea (Doctoral Advisor at Stanford University, and my mentor) on language-independent conceptual bugs. Roy and colleagues found that many such bugs in students' code in environments such as LOGO could be traced to what he called a 'superbug': the belief or understanding that somehow a computer can interpret a students' intention due to some sort of intelligent interpretive process.

Another key source of difficulties lies in a weak understanding of the temporal logic of programs and program states: the mental model of a program as a sequence of instructions, where actions at any point are the result of what has gone before (including values of variables, and other attributes of objects), and determine what



■ Fig. 2. What are the roles of Counter and NumberOfTimes in this Scratch code?

happens next - unless code is placed in event handlers which are triggered by an action at any time.

These two shortcomings impact students' fundamental understanding of data and control flow in general, and in particular, concepts such as initialization, loops and how state changes with each iteration of the loop (including values of variables), as well as how to create logical expressions in order to manage repetition and the conditional logic of If commands.

## Variables

Learners need to understand that variables are named values or quantities, and that their values can change over the course of a program. Some difficulties with variables in block-based programming environments are:

- A hangover from maths that leads learners to think variables stand for a mystery, unknown value and have names like 'x' or 'y'. Research encourages teachers to speak of a variable as having a 'role' in a program, and having a meaningful name. Learners should be guided to make that leap.
- The metaphor of variable as a box or placeholder leads learners to assume that they can hold multiple values at a time. A teacher might target this misconception with a debugging exercise using buggy code as in Fig. 1 (That's not the only problem with the code!).
- Despite syntactic simplifications and scaffolds such as color and shape in environments like Scratch, learners need help understanding the semantics of variable use. What does it mean to control a loop with a variable, or an expression that contains a variable (Fig. 2)?
- When does the value of a variable change? Many students will incorrectly answer the question in Fig. 3, even after learning about Set and Change blocks in Scratch.
- Assigning a variable to another somehow links the two, causing a change in the value of one to result in a change of value to the other (even after the assignment).

## Loops

In my work, I have found that students often struggle to distinguish between what goes inside a loop and what precedes or follows a loop. This is often the case when a first or last iteration is different from the 'inner' part that repeats. In addition:

- When there are multiple blocks inside a loop, instead of thinking of the loop as executing the entire sequence of actions, some students believe that each action is

■ Fig. 3. What is the value of the variable 'steps' after the following code sequence is executed?



repeated separately before subsequent action(s) are similarly separately repeated.

- Some students struggle to understand that an expression involving the control variable of a loop can have different values in each cycle of the loop. They believe that a loop repeats the same set of actions, and expect loops to produce exactly the same output in every iteration.
- 'Off-by-1' and fencepost errors in loops have been well-documented in several studies involving various age groups. Most middle school students in Israel, and in two classrooms in California, incorrectly answered a question from the 2012 Israel National Exam shown in Fig. 4.
- Many students in my experience also struggle to formulate a loop terminating condition involving a Boolean expression.

## Boolean logic

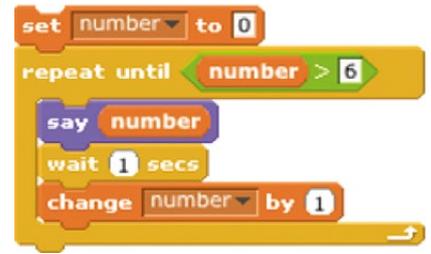
Students have been found to perform less well on logic-based questions on the Advanced Placement CS A Exam in the US than on any other type of question. The Boolean AND/OR operators are often mistakenly interpreted as they are in the English language. Specifically, students tend to misinterpret the OR operator as true when one of the operands is true, but not both. We found this to be the case with middle school students learning Scratch. In order to address this, we should encourage students to create Boolean expressions to model real-world phenomena.

For example, write an expression to model the situation: A car gives a warning beep if:

- Driver's seat-belt is not locked
- Passenger's seat-belt is not locked
- Car is being driven

## How can teachers help?

The first part of the solution is awareness. It is a rare teacher who will not attempt to address these student targets of difficulty if they are cognizant of them. Secondly, most misconceptions can be addressed if students are confronted with them explicitly. I have long argued for a balanced pedagogy of programming that combines active exploration approaches with methods of scaffolding that



■ Fig. 4. What will the sprite say when the script above is executed?

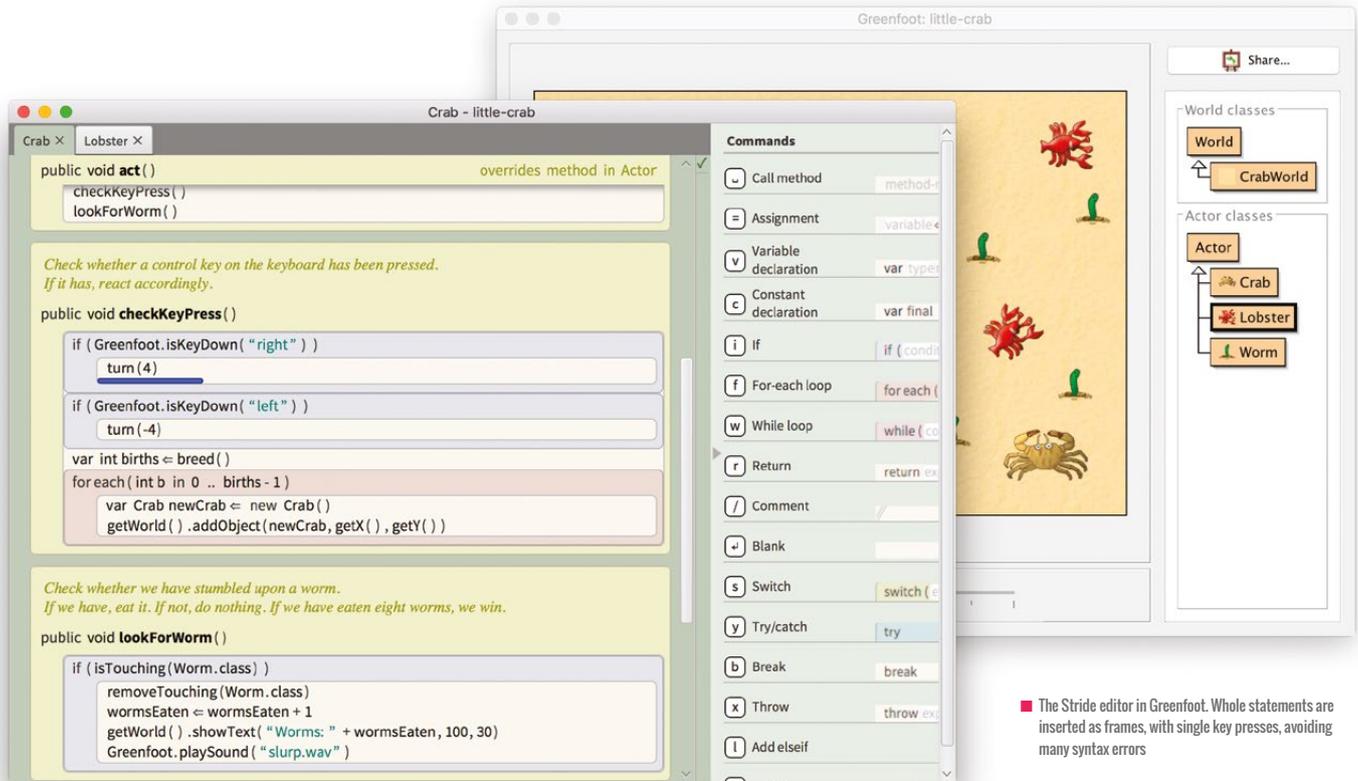
enable knowledge construction and deep understanding of targeted concepts.

Besides the ideas and examples shared, other useful strategies that have been known to work include demonstrating flow of control through visualizations and code-tracing; having students trace code, including but not limited to 'unfurling' loops that involve variables whose value changes in each iteration and involve a termination condition based on that variable; and targeting known difficulties and misconceptions head-on through code comprehension, worked examples, and debugging exercises.

Lastly, using conversational metaphors, though a powerful pedagogic technique to introduce concepts in new learning contexts, should be avoided, as mapping conventions for natural human language instructions onto programming often results in exacerbating the problem. (HW)

## FOR FURTHER READING

- Grover, S., Pea, R., Cooper, S. 2015. *Designing for Deeper Learning in a Blended Computer Science Course for Middle School Students*. *Computer Sc. Education*, 25(2), 199-237
- Grover, S., & Basu, S. 2017. *Measuring Student Learning in Introductory Block-Based Programming: Examining Misconceptions of Loops, Variables, and Boolean Logic*. In *Proceedings of the 2017 ACM SIGCSE Technical Symposium on Computer Science Education* (pp. 267-272). ACM.
- Pea, R. D. 1986. *Language-independent conceptual 'bugs' in novice programming*. *Journal of Educational Computing Research*, 2(1), 25-36.



■ The Stride editor in Greenfoot. Whole statements are inserted as frames, with single key presses, avoiding many syntax errors

# FROM BLOCKS TO TEXT: PROGRAMMING WITH STRIDE

Transitioning from block-based to text-based programming is one of the great challenges for educators and students. One of Stride's developers explains its unique features

STORY BY Michael Kölling

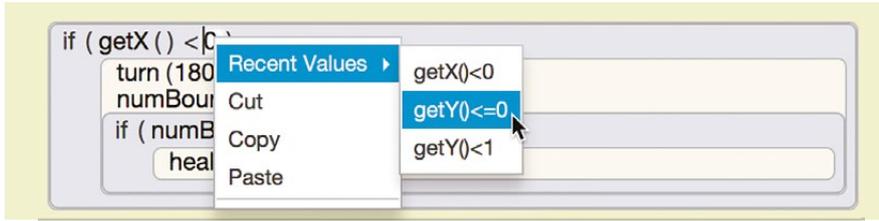
**W**hen teaching programming in our schools, it seems we have managed to agree on both ends of the curriculum. At the lower end, we start with block-based languages, available in fantastic and popular systems such as Snap! and Scratch. At the high end, we program in text-based languages: Java, Python, or something similar.

That leads to the question: How do we get from one to the other? Transitioning from block-based languages to textual

ones has become one of the difficult stepping stones in the progression of programming. It is also a hot topic: in the last edition of Hello World, this challenge was mentioned in three different articles. All of them discussed progressing from Scratch to Python, and where they offered advice, it was based on providing side-by-side comparisons of block- and text-based code segments to facilitate a transition. We can, however, do better. Much better.

## The problem of transition

Moving from blocks to text is not easy. Neither for the learner, nor for the teacher. As teachers, we need a much deeper understanding of a more complex system. Good object-oriented programming practice is not as easy to pick up as tinkering in Scratch. As learners, we are faced with numerous new challenges. Even though some fundamental programming concepts transfer from one system to the other, several distinct additions in complexity pose



Local undo: every frame supports an individual undo operation via a context menu

significant challenges: textual languages are more difficult to read, statements have to be known and memorised (they cannot be easily browsed and explored), the exact syntax has to be memorised and typed in (without making typing mistakes!), the

still persist, even after pupils have realised the equivalency of the two systems.

### Have your cake and eat it

This is where Stride comes in. Stride is a new language in a unique

side-by-side, but at the same time. To get a concrete impression of what this looks like, watch the Stride introduction videos (details in the 'Where to start' box).

Frame-based editing avoids many syntax errors. It will not avoid all errors, but a good number of the most common ones. There can never be any mismatched brackets, there will never be missing semicolons, users do not have to remember whether to use commas or semicolons in a For loop, and indentation will never be wrong. On the other hand, longer programs are arranged logically and are easy to navigate (no wild hopping between multiple random scripts laid out in random arrangements), a full professional library is available, and expressions can be typed quickly and conveniently. The functional scope and performance available is the same as in professional systems, such as Java or Python. (Stride runs on the Java Virtual Machine.)

## STRIDE AIMS TO UNIFY THE BEST CHARACTERISTICS OF BOTH BLOCK-BASED AND TEXT-BASED EDITING

number of methods available is much larger, indentation and layout has to be managed, parentheses and brackets may be out of balance, data types have to be understood, error messages have to be interpreted, and much more along the same lines. A more detailed discussion of these challenges is available in a paper: see the 'Further Reading' box on page 66.

In short, in addition to being faced with an entirely unfamiliar programming environment, learners are confronted with a substantial number of additional challenges. None of these is trivial.

frame-based editor, built into the Greenfoot environment.

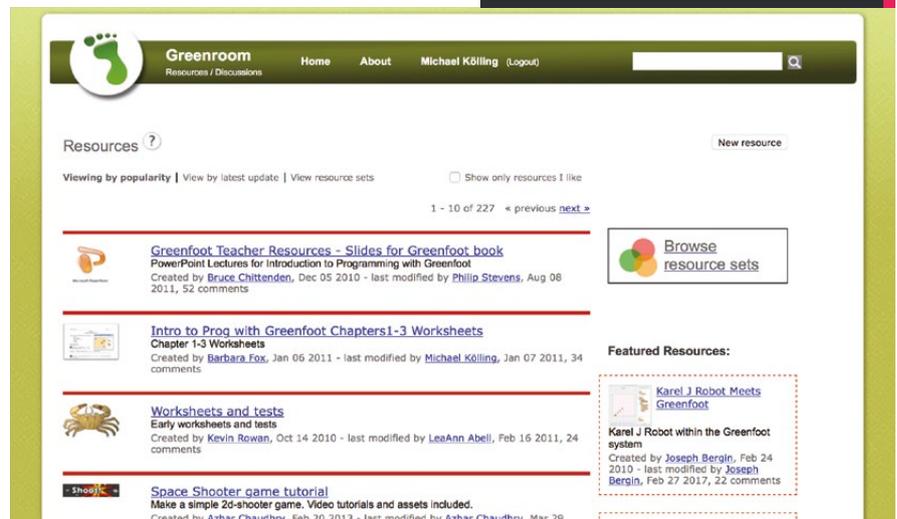
You can think of frame-based editing as the missing link between blocks and text: it is keyboard-driven, just like text, but statements are entered in their entirety with a single action, just like blocks. The layout of the code is in classes and methods, just like classic text-based languages, but statements can be browsed to discover them, just like in block-based systems. In short, Stride aims to unify the best characteristics of both block-based and text-based editing in a single system. Not

## WHERE TO START

- Greenfoot [greenfoot.org](http://greenfoot.org)
- Joy of Code videos [helloworld.cc/2qSnnll](http://helloworld.cc/2qSnnll)
- The Greenroom [greenroom.greenfoot.org](http://greenroom.greenfoot.org)
- Introduction to Stride videos [helloworld.cc/2qSusyM](http://helloworld.cc/2qSusyM)

### Side-by-side display

Most efforts to address this problem are based on the idea of side-by-side presentation of blocks and text. This may either be done on paper, as in the lesson plan presented in the last edition of Hello World, or automatically in a programming system, such as Pencil Code, Alice, or Droplet. However, there are two problems. Firstly, there is no evidence that side-by-side presentation has a real learning effect. Some researchers argue that presentation in two alternative modes may add to cognitive overload, rather than help. And secondly, most of the problems, such as having to memorise and then type out the statements,



## FURTHER READING

### FRAME-BASED EDITING: EASING THE TRANSITION FROM BLOCKS TO TEXT-BASED PROGRAMMING

BY Kölling, Brown, and Altdmri  
 URL [kar.kent.ac.uk/50406](http://kar.kent.ac.uk/50406)

As a result, learners can progress to more challenging and interesting projects when they outgrow Snap! or Scratch, but do not get stuck in the frustrations and unnecessary errors of syntax.

### Transitions

One obvious use of Stride is as a stepping stone between block systems and text-based systems. Stride can, however, also be used as a first system for learners with no prior experience. The target age group is from about 12 years old upwards. Since Stride scales well in functionality and performance, it can be used throughout key stages 3, 4, and sixth form.

When the time comes to move on to traditional text-based systems, Stride and Greenfoot help as well: the Stride editor offers a 'Preview as Java' option that displays the code in Java syntax. Greenfoot then offers a 'Convert to Java' function that lets users convert their classes and continue working in straight Java.

Java and Stride classes can be mixed in Greenfoot, and standard Java projects are also supported. This allows Stride to provide a gentle, easy progression from

### Methods

*Describe your method here...*

```
public void act()
```

```
if (condition)
```

■ A frame for an If statement, inserted into a method. Users fill in the condition and body. Other syntax is automatic

blocks, through frames, into a standard text-based language.

### Starting with Greenfoot

Greenfoot is a well-established system with large amounts of available teaching material and a very active user community. To start getting familiar with Greenfoot, download the system from the Greenfoot website (URLs are in the 'Where to start' box) and try it out. To get started with project ideas, read the Greenfoot book, or watch the free Joy of Code video series. (You can also get your pupils to work through these videos. They have been used in thousands of classrooms.) Then go to the Greenroom and join our teacher community, where you can chat with other teachers, find lesson plans, project ideas, slides, quizzes and more. Finally, watch the Stride overview videos to get a good feel for the frame-based editor.

Currently, much of the material is presented in terms of Java. However, you can write exactly the same code in Stride; after watching the Stride videos, it is easy to work out how to do this.

### To text or not to text

I want to leave you with one final thought to consider: do you actually need to teach a text-based language? I am not talking about curriculum requirements, but ideal pedagogical considerations.

In Stride, you can do everything you can do in Python or Java. Just without the pain of unbalanced parentheses, incorrect indentation, or hours spent over a misplaced comma.

These syntactical aspects have no fundamental pedagogical value; they are an accident of our time. I would argue that editors like Stride are the way every casual programmer should program. All computing concepts can be taught in this system, with less pain. For general computing education (for non-professionals), this is all that is needed. We can then transition those pupils who want to go forward into a career path in computer science to traditional text-based systems. For everyone else, the resulting syntax problems are arcane, unnecessary, and harmful. (HW)

### DID YOU KNOW?

Stride is also available in BlueJ ([bluej.org](http://bluej.org))

■ A frame's context menu offers different delete operations with a preview of the effect (red area)

The image shows two overlapping screenshots of the Stride editor's code editor. The top screenshot shows a code block for the `incrementAge()` method. A context menu is open over the `if (getAge() > MAX_AGE) setDead();` line. The menu options are: Copy, Cut, Delete (highlighted in blue), Disable, and Delete outer frame. A red shaded area indicates the effect of the 'Delete' operation. The bottom screenshot shows the same code block, but the context menu is open over the `setDead();` line. The menu options are: Copy, Cut, Delete, Disable, and Delete outer frame (highlighted in blue). A red shaded area indicates the effect of the 'Delete outer frame' operation.

Get your **CAS** **FREE Barefoot**

Supported by **BT**

# computing workshop!

**Our lives and work are increasingly shaped by technology. The government predicts that over 90 per cent of all jobs will require digital skills by the 2030s. To ensure everyone can succeed in that future, we must first set everyone up with the right tech know-how.**

The Barefoot Computing Project helps primary school teachers grow their confidence in computational thinking. These skills help pupils understand the building blocks of the digital universe.

A Barefoot teacher workshop is a great way to get started with Barefoot's resources. It's fun, FREE and very hands on. You don't need any technology and it's delivered by a volunteer at a time that suits you.

Get to grips with computational thinking concepts the easy way, and prepare your pupils for their tech-powered futures by booking a Barefoot workshop today!



They only take approximately 60 minutes



Hands-on and fun to take part in



Easy to grasp, with real world examples



They help you get the most from your Barefoot resources

## To get your **FREE Barefoot** workshop...



### 1. Register

Go to [www.barefootcas.org.uk](http://www.barefootcas.org.uk) to register – it's quick and it's also where you access hundreds of resources



### 2. Booking a Barefoot workshop

Fill out your details on the same website and choose a time and place to suit you



### 3. Enjoy!

The friendly, specially-trained volunteer will come to your school, show you how to get the best out of the Barefoot resources, and introduce you and your colleagues to the wonderful world of computational thinking

**Go Barefoot:**

[www.barefootcas.org.uk/book-a-free-workshop](http://www.barefootcas.org.uk/book-a-free-workshop)

SWAY GRANTHAM COMPUTING TEACHER

# CHILDREN TEACHING CHILDREN (AND TEACHERS!)

Trying to bring more technology into your lessons without losing teaching time? Digital Leaders are the answer!

**D**igital Leaders are children within your school who are trained to make it easier for teachers and other pupils to use technology in lessons. For example, this could involve charging equipment or getting all of the laptops or tablets out and ready during a lunchtime. When you've shown them what to do, any member of staff can book equipment out to them, so getting everything ready becomes a doddle! More than this: with just a few simple tips, Digital Leaders can update software, restart the printers, or change the batteries in the Bee-Bots, leaving you time to focus on other things. They can also use some of their free time to practise using an app or website, so that they can support other children in lessons, and even train teachers on how to use them.

## Who do I pick?

Some schools choose to have several children from each class, while others choose the older pupils to give them a position of responsibility – but really it's what works for you. When you've decided on who you want, launch the scheme with an assembly outlining the role and explaining why children are so much better than adults in this area. I like to show children the technology I was using when I was their age, and explain that there's no wonder teachers don't understand all the latest tech! After this I invite pupils

## WHAT COULD THEY DO?

- Check ICT resources
- Prepare ICT resources for lessons
- Compare different apps, websites and software
- Create model examples using apps, websites, or software
- Suggest popular apps, websites, or software that could be used for learning
- Run assemblies on online safety
- Hand out online safety information at parents' evenings
- Offer tech support in lessons
- Train staff on specific resources
- Run clubs for younger children

■ Sway's Digital Leaders in action



to apply for the post. I usually use a Google Form for the children to complete. I add a link on the school website as well as giving them a paper copy to take home. I also let them use the school computers one lunchtime.

Depending on the number of applications you receive, you may also want to run interviews. I find that with a year group I haven't taught, interviews can be a good way of getting to know them better and seeing how they interact. For this, I give them an unseen app or website to use for the first time, and tell them to find out how to use it. The idea behind this activity is to see who will click around and have a go, who will keep going, and who will enthusiastically share what they've done with the person sitting next to them. These are the traits I want in my Digital Leaders!

## Why do they make a difference?

Even the most technologically enthusiastic teacher doesn't have as much time for exploring new technology as they would like. Children have the free time to research, compare, and try out new technologies, but this is also an empowering experience. Those children are able to lead a subject, make recommendations that truly matter and educate their peers as well as the adults around them.

What are you waiting for? **(HAW)**

**Sway Grantham** is a computing teacher and digital technology consultant. She promotes the importance of using technology as a tool across the curriculum.



SHAUN WHORTON TEACHER

# SUPER CPD FOR COMPUTER SCIENCE TEACHERS

Shaun Whorton discusses how completing Harvard's Introduction to Computer Science MOOC (CS50) improved his teaching practice.

**R**ecently, I've found my classroom resources becoming a bit dated and stale; using them every term for the past few years has slowly reduced my eagerness to teach key concepts to students. I needed something fresh. To find inspiration, I enrolled on Harvard's CS50 course, 'Introduction to Computer Science'. I didn't actually attend Harvard: the world-class university offers a MOOC (Massive Open Online Course) version of their popular course via edX, for free and at a pace that suits you.

## Binary lamps

Having a Computer Science degree and programming knowledge behind me was encouraging and I was eager to learn from the world's best. But, seeing how some of the course tutors teach certain concepts was the most inspiring. For example, Prof David J Malan, the course leader, teaches his students binary using desk lamps alongside a lot of input from the lecture attendees; I'm now adapting my approach to match his. An advantage of being in the 'learning' seat rather than teaching has refreshed my understanding of what it means to learn a language or concept from the beginning, how it feels to be utterly confused and, upon learning the solution, what I could do in a classroom situation with my own students to circumvent this confusion.

## Students in mind

The course showed me new ways to teach abstraction, web development and the discipline of computer science in general, even though this was not the objective! Studying how the course tutors explained their concepts has impacted my own teaching. I went away to my own resource pool and edited them to within an inch of their lives, using what I have learnt from the course in a myriad of ways. Not only did the course force me to think and reflect upon my

## WHAT IS EDX?

edX is a collaboration between MIT, Harvard, Berkeley, and other major universities to put their courseware online. It is one of the primary vehicles through which top-tier institutions are surging into the online education space in an unprecedented expansion of academic provision.

own practice, it also improved my knowledge of several programming languages. The course itself is broken down into nine assessment tasks (psets) which are accompanied by formative assessment quizzes along the way. Each assessment is graded pass/fail, leading onto and informing the next task. The course ends with you, the student, drawing together your newly learnt skills in the form of a final project. I built a Google Maps/BBC News API mashup that gave the user local news depending on where they dropped the pin. I'd never built anything like this before, and it was incredibly satisfying personally, but in the back of my mind I was already brewing a way to teach this to my A level students.

## Only for teachers?

Teachers of GCSE and A level computer science could benefit from taking this course. It has really enhanced my teaching twofold, from both a subject knowledge and a pedagogical point of view. I would also signpost high-ability students to at least attempt the course alongside their formal academic studies; persevere and they will no doubt reap the rewards! [\(HW\)](#)

**Shaun Whorton** is Head of Computer Science and Business at a secondary school, and author of assessment material for a large exam board.

PAUL POWELL CURRICULUM LEAD FOR COMPUTING

# LINKING DATA REPRESENTATION TO CREATIVITY

Data representation using binary can be a little dry, so why not link it to a creative project?

**F**or a few years, I have taught a unit on binary, data representation, and logic gates. Although this has been successful enough, I felt that it was a little dry and abstract. I thought that instead of teaching it as a block, I could revisit binary several times over key stage 3 by tying it to graphics and sound projects as well as hardware theory.

We are currently running a sound unit in Year 8 (12- to 13-year-olds) where we start with some simple binary mathematics (binary having been introduced with images in Year 7). We are using Audacity (a free, open source-program) to start to look at sound. Audacity is fantastic for this because you can zoom into a sound, and this enables you to see the individual sample points. The theory is then linked to a practical project.

### Going deeper into binary

When pupils understand binary mathematics, we can prompt them to consider how manipulating sound is actually manipulating binary digits. Multiplying every sample point by a constant amount will make the sample louder whilst dividing will make it quieter. Fade in and fade out are a bit more interesting: to fade out, we multiply by a progressively smaller value until we get to zero. Speeding up and slowing down will involve interpolating: removing sample points or adding more in.

Simple operations such as cut and paste, deleting, adding silent sections, adding noise, and so on are all easily explainable in terms of binary mathematics. Talking about creative projects in this way will, I hope, help students to see binary as something that is relevant and observable rather than arcane and abstract.

### Output and compression

Having created their audio projects (in our case a simple radio advert) it is time to output. Here again there is an opportunity to output with different qualities and compression settings. Opening a high-quality and a low-quality version together in Audacity allows you to zoom in to see the difference in the precision of the sampling.

### Final thoughts

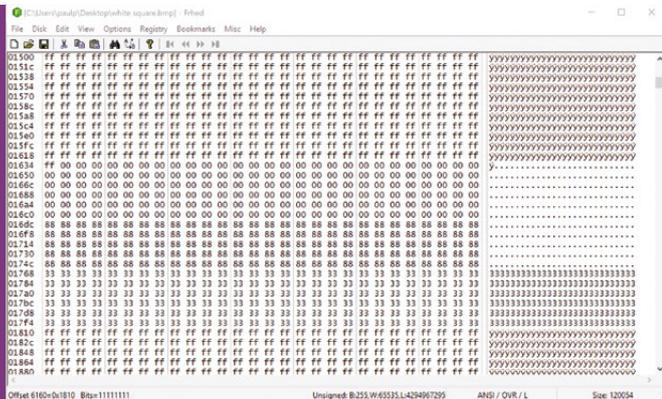
One of the strengths of this approach is that it enables the teaching of the curriculum in a more integrated way. By sourcing our sound clips from creative commons sources, we are also introducing some digital literacy. Integrating the curriculum in this way helps to break up the more academic sections and give them real meaning. (HW)

Paul Powell is the Curriculum Lead for Computing in George Mitchell School, East London.

### HEX IMAGE EDITING

One simple way to show the direct link between data representation and images is to edit the data directly. Create a 200x200 image, all white, and save as a .bmp file. Open the bitmap in a hex editor (There are lots of free ones available). At the start of the file you will see the metadata, including the height and width (200 is C8 in hex).

The image data is in RGB triples of FF FF FF. Edit these, then reopen the image in an image editor to see the result.





**NICHOLAS PROVENZANO** TEACHER

# DAD, CAN I PLAY THE DANCING CAT GAME?

The Nerdy Teacher tells us the effect Scratch has had on his son...

**D**ad, can I play the dancing cat game?" That's how the conversation always starts when my son wants to play on Scratch. Before Leo had a Kano Kit, I had never used or seen Scratch before. It was a whole new world for me, but my son was drawn to it. I figured that I had better start learning how to use it so I could help him develop his skills and do some fun things. Well, he became the master in no time!

Within the first week or so, I was working at my computer and he was on Kano playing Scratch when he asked me to take a look at what he had done.

"I made a submarine for the cat to drive in!" I watched as he hit the space bar and the cat bounced around the screen, surrounded by a crude drawing that resembled a submarine. He had figured out how to do this all on his own. No support from me or YouTube. He is five years old and has limited reading skills, but he was able to figure everything out through trial and error. It's truly amazing! As Scratch celebrates its tenth birthday, this story demonstrates why it is such a great tool for parents and educators to share with their children.



## Scary concept

Creating with code sounds like a very scary concept for the average person, but Scratch does a great job of making everyone feel comfortable using their interface and learning about the cause and effect relationship of commands and actions. The creators at MIT Media Lab have worked very hard to make sure that Scratch is available to everyone by making it free and accessible in more than 40 languages! I believe that all children should be exposed to coding at some point in their education, and Scratch is the perfect introductory tool for all grade and skill levels.

Another great aspect of Scratch is their ScratchEd page. Teachers do not need to feel intimidated about using Scratch in their classrooms. ScratchEd provides great stories showing how Scratch is being used by teachers every day around the world. It is a wonderful community that allows educators to connect and share. The creators of Scratch know what teachers and learners need and do a great job putting together a program and network that supports every learner.

As I watch my son explore Scratch each time he asks to play, I know that he is only just beginning to see the potential of the program. I want my son to be an explorer when it comes to learning, and Scratch lets him do that in a fun and safe way. After ten years, Scratch has become a staple in classrooms around the world, but also in my own home. **(HW)**

**Nicholas Provenzano** is a high school English teacher, author, speaker, and consultant. He has been featured on CNN.com, Education Week, The New York Times, and other media outlets.

# GET PROGRAMMING OR GET GOING

After-school clubs provide a great opportunity for students to learn and be creative

STORY BY Nicholas Provenzano

**O**nce the final bell rings, most teachers are ready to pack it up and head home for the day. Others have signed up to support clubs and have to get their rooms ready for these passionate students, looking to explore various topics more closely outside traditional school hours. Lately there has been a rise in the number of programming clubs, and I am

the proud advisor to the club that started this year at my school.

When asked why they want to be in a programming club, Evan N replied, "I wanted to learn to code and have an opportunity to meet people interested in the same thing" and Abbie B said, "I wanted to learn programming and meet new people".Evan, Abbie, and all of the

others share a passion for programming and just want to connect with others who feel the same way.

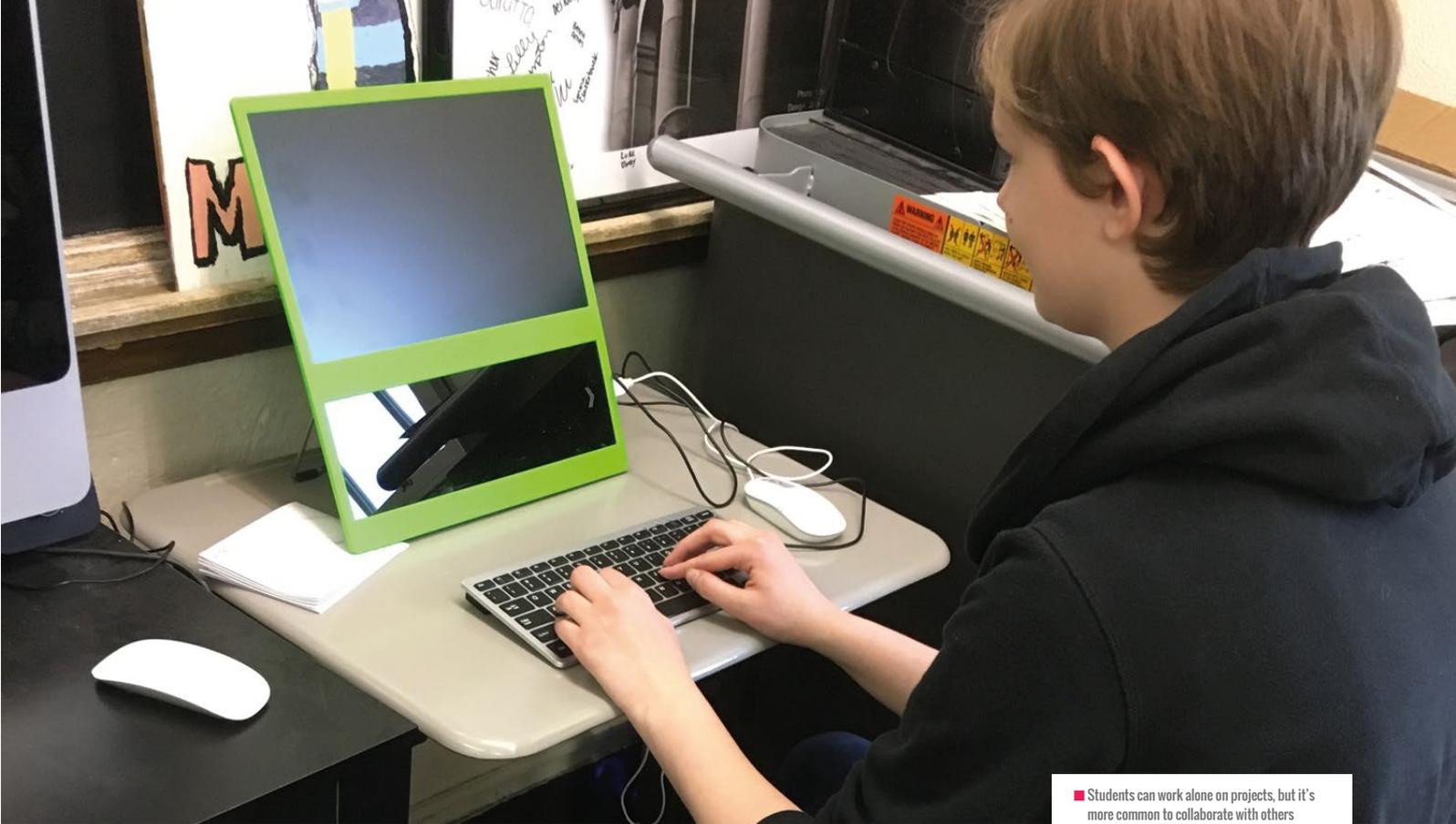
These comments show why clubs are popping up in schools around the world. Each student has their own reasons for learning to code, and it brings them all together because the traditional school system does not allow them to dive into programming in ways that are meaningful to them. After-school activities provide students with opportunities to learn at their own pace and focus on topics that matter to them.

## How it works

In our programming club, there are students who are working in small groups on their own projects, and as one team on a larger project. There are students who are exploring the new Code Kit from littleBits. They are checking out the different types of code they can create, and working on different lessons to share with elementary students in our district to help them learn to code. When they are done, they will be able to publish their code and lessons on the littleBits website. This is such a great opportunity for students to try new things and create amazing code that can be used to support others to learn.

■ Students meet after school to discuss their latest programming projects





■ Students can work alone on projects, but it's more common to collaborate with others

Another group of students is participating in a Raspberry Pi Zero W competition run by the school. They were given a Pi Zero W kit, which they need to use to create something innovative. They are working on their project after school and whenever they have free time. It is the second year that the school has hosted this competition, and the students are excited to see what is possible with the new iteration of the Pi Zero. Watching students excitedly huddle to discuss possible design and coding options is something you do not normally see in traditional computer science classes.

The big project that the entire group is working on is the video game. All the students came together to design their own RPG video game. Since each student

## SUPPORT A PROGRAMMING CLUB

Here are just a few reasons why you should support a programming club in your school.

- Students often feel that the things they care most about do not matter in school. A group of students can feel validated if their passion for programming is accepted as part of school in the form of a club. We want all students to feel that what they care about matters, and a coding club is no different.
- Finding a space to get together and work can be tough for students. Many places are noisy or do not have enough room for their group. Other

students might not be able to travel to a friend's house easily. Providing students with a classroom to get together and explore programming can make a huge difference to their participation.

- A school programming club in your classroom allows other students to learn about the club and possibly join. The more students have heard about a club, the more likely they are to be willing to stop by and see what is going on there. Clubs need to be inclusive, and the school environment is the perfect place to support this. If we want more students exposed to programming, it needs to be accessible.

## “ THE STUDENTS HAVE DONE A GREAT JOB COMING TOGETHER TO SUPPORT THOSE WHO NEED THE EXTRA HELP

has a different skill level when it comes to programming, they quickly divided up into various design groups to help create this game. One group is dedicated to designing the characters and the overall environment of the game; another is focused on the

overarching story; and another is working on basic game mechanics. They created a shared folder where each group can deposit the work they create during the week, and they use the meeting time after school to show what they have completed.

It might seem crazy to have so many different skill levels in one group working on a project, but the students have done a great job coming together to support those who need the extra help.

Advising the programming club has been an amazing experience. I have been so fascinated watching them work that I was a bit jealous I couldn't jump in there and start coding the projects with them. The after-school club has been a great learning experience for the students, but also for their advisor. (HW)



# LEARNERS AT THE HEART OF LEARNING

On 9 May 2012, Woodbridge Primary School became one of the first pilot Code Clubs. Five years on, here are three views of its contribution to learning

STORY BY Mark Krisson, Caroline Moore, Bob Higham

**I**n the 1960s, Seymour Papert asked: “Does the child program the computer or does the computer program the child?” Code Club is a great vehicle to help primary pupils move from passive consumption to active creation and invention. Learning through increasing experimentation and tinkering is certainly this Code Club’s direction of travel.

### Bob Higham: volunteer club leader

We started our Code Club in the summer term of 2012. It was (and still is) very easy to get off the ground, as all the materials needed to get going were freely available and ready to use. Over the last five years we have tinkered and experimented with its structured checklist approach, and wrestled with much more ‘chaotic’ learning. Seymour Papert said

“When you go to school, the trauma is that you must stop learning and must now accept being taught”. As an after-school club, we are thankfully free to try and remove any such trauma.

Crumbles and micro:bits. Some continue to use Code Club project materials and ideas, while others work on much less structured collaborative projects or experiment with ‘remixing’ the work of others.

**“**CODE CLUB HAS NO DOUBT INSPIRED THE PUPILS, AND MEMBERSHIP IS WELL BALANCED BETWEEN GENDERS

Early on, the best ‘lightbulb’ moments occurred when mistakes were made; leaders stood back, and bug fixing forced the tinkering process to start. We now have Year 5 and 6 pupils in Code Club, who first used Scratch in Year 2. They are looking for projects beyond those initially provided, and are now working with Raspberry Pis,

All club members have access to Google G Suite for Education accounts, and use those tools inside and outside the club (and school) to collaborate and communicate with each other and club leaders. Use of the G Suite mixes cloud-based ICT into the learning pot. We are also experimenting with managing Code Club projects using Google Classroom.

Access to additional kit relies on funding, something which is in increasingly short supply. Code Club is free and should definitely remain that way, but mechanisms to fund additional hardware have to be addressed by individual clubs, especially if the learning boundaries are to continue to expand. All of our additional kit has been funded by the leaders themselves, or by donations.

While the core activity of the club has changed little in the past five years, the learning approach has evolved and continues to change. If you are interested in a more detailed assessment of the impact of Code Club then please take a look at the recent Code Club evaluation feature on pages 36-37, the report at [helloworld.cc/2ph2IMl](http://helloworld.cc/2ph2IMl), or at the Code Club blog.

### **Caroline Moore: CS curriculum lead and club host**

Firstly, the support of Bob, as a STEM Ambassador, was an essential part of establishing and running the club. This has given the children the opportunity to develop their skills in computer science, both through the provided activities and through his expertise and interests.

While the original activities provide a good basis from which to start, the introduction of Crumble, CrumbleBot, Arduino, Python, HTML, and Raspberry Pi have extended pupils' skills. Work with BT locally has also provided wider opportunities for pupils to experiment with programming activities such as CoSpace.

Code Club has no doubt inspired the pupils, and membership is well balanced between genders. It provides pupils with the opportunity to work collaboratively and share their achievements, while giving them a chance to tinker and experiment. Many other issues have been addressed informally, including online safety and copyright.

### **Mark Krisson: Headteacher**

The learning opportunities offered to pupils by Code Club and other computing initiatives are both powerful and refreshing. This is particularly so at a time when the Department for Education's harsh focus on narrow, knowledge-based assessment practices restricts curricula and divergent thinking,



■ From the Bee-Bot to the Raspberry Pi: Woodbridge Primary School computer lab ready for action!

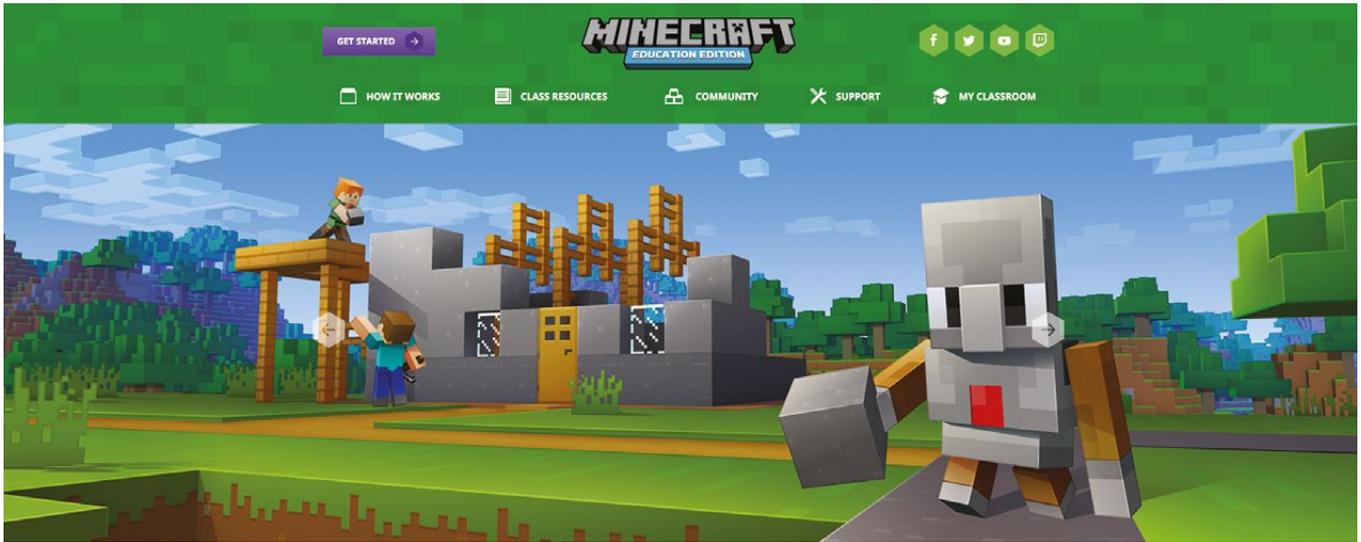
I was very heartened in my first impression of the Code Club materials. They provided children with situations which would stimulate and motivate them, and yet lead them to problems and challenges to overcome.

I was very excited at the learning potential, as I became more aware of the development and easy availability of hardware such as Raspberry Pi and Crumble. It increases the power of the learners and the learning when they are in control of the hardware. That power is

further magnified when the learners are able to play with the hardware; to tinker.

This learning and excitement has grown even more in recent times as I have become aware of Bob taking the club of coders 'off piste', as the children increasingly make the decisions about where their learning will go, and what they want to do.

Learners at the heart of their learning, with the tools and the permission to determine their destination and their route. What could be better than that? (HAW)



# HOW MINECRAFT IS INSPIRING A NEW GENERATION OF PROGRAMMERS

Our world is becoming increasingly reliant on technology, which means more and more people with the necessary skills will be needed to help it thrive and grow. How do we get kids engaged in computing so they can move on from being digital consumers?

STORY BY Lara Lipscombe-Southwell

**T**he 2014 European Commission report, *e-Skills for Jobs in Europe*, identified a possible 900,000 skilled ICT worker shortage by the year 2020. As the demand for STEM professionals increases, the pressure falls on educators' shoulders to ensure that the next generation is ahead of the game.

Using games such as Minecraft, educators can excite students about programming concepts by anchoring them within a familiar environment. As it's a sandbox game, children have both the control and the freedom to create. Whether they are coding Java mini-games in a private server, or building structures to improve spatial reasoning, every child's game is different.

The classroom application of Minecraft is gaining popularity with the release of the educational version, and a growing

number of summer camps, after-school clubs and online courses are now using Minecraft as a learning tool. In one camp in the USA alone, 12,000 8- to 13-year-olds sign up for a week-long course to learn programming, building, and design.

Using Minecraft as an educational tool is not just an excuse to play games. It enables children to use their creativity, improve their problem solving ability and

tackle computational thinking, all of which are important skills that can be applied to a variety of subjects.

Minecraft has the flexibility to be so much more than just a game, and with a growing number of educators learning to appreciate its potential, it could be the way for us to successfully encourage and inspire the next generation of computer scientists. **(HAW)**

## ABOUT CODE KINGDOMS

[www.codekingdoms.com](http://www.codekingdoms.com)

Code Kingdoms is an online platform where kids aged 8-14 can learn to code their own mini-games and mods for Minecraft, using a specially designed Java Code Editor and video tutorials.



# SCRATCHMATHS

## MATHS MASTERY AND COMPUTING

Do you need more time to teach maths? Well look no further! ScratchMaths gives you cross-curricular time to teach maths and computing too!

STORY BY Jane Waite

**W**orking with teachers, the UCL Knowledge Lab has created ScratchMaths, a free programme of maths planning for Year 5 and Year 6 classes that teaches a range of maths objectives. Funded by the Education Endowment Foundation, the material has been trialled in 50 schools across England. You can now access the Year 5 material, with the Year 6 resources due for release in Summer 2017.

ScratchMaths includes activities to teach symmetry, rotation, translation, coordinates, negative numbers, Roman

numerals, pattern, perimeter, sum of angles, multiplication, general problem solving, and much more! The resources teach sequence, repetition, selection, decomposition, debugging, and the tricky Scratch 'broadcast' blocks. They include a fantastic introduction to 'make your own blocks'.

The UCL team has created a way of teaching maths through programming that features lots of great ideas. You could use them to teach maths in other year groups, or even to teach other subjects. There are physical activities to get learners up and moving, as they play turtle to draw shapes

**TEACHERS AND LEARNERS DON'T ASSUME UNDERSTANDING, AND ENSURE THERE IS PROGRESSION AND MASTERY**

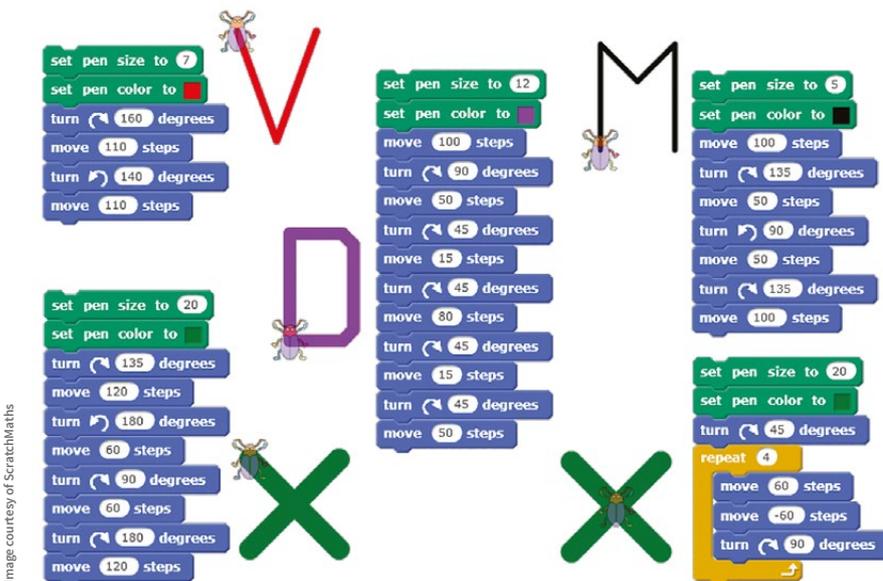


Image courtesy of ScratchMaths

■ Learners look carefully at Roman numerals as they animate a process to draw them

or chant to understand program control. Activities such as reading code, predicting what it will do, and comparing code get learners to look carefully, use logical reasoning, and trace the code they create. Careful, guided exploration of specific key commands mean teachers and learners don't assume understanding, and ensure there is progression and mastery. Creative tasks give learners the chance to show what they have learned and give teachers opportunities for assessment.

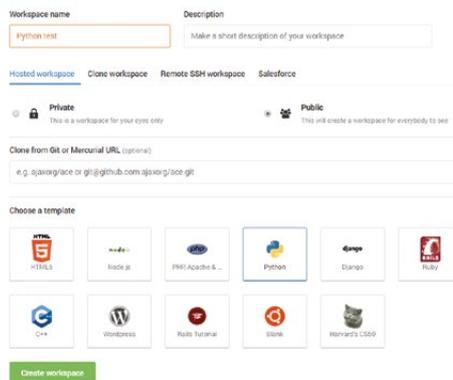
This really is a fantastic set of resources – do have a look, but also read the planning material and follow the approach that has been developed. Try ScratchMaths for your own professional development on how to teach programming and to get more time to teach maths! (HW)

# BROWSER-BASED IDES

Create code in a browser with these web-based development environments

**W**hen programming on a computer, using an IDE (integrated development environment) is the preferred way to code. Students get helpful highlighting to make it just that little bit more readable, and a few IDEs let you run and preview the code as well.

Instead of maintaining the libraries and IDEs on all your machines, a great alternative is to use a web IDE: an IDE that lives on the internet. These generally have the same features as an IDE application, and can be used anywhere with an internet connection. Here are some of the best... **(HW)**



## QUICK FACTS

- Supports specific teacher and student accounts
- Allows all major languages
- Runs code in browser

■ With a tabbed view, you can work on multiple files at once

## CLOUD9

**INFO** FROM Cloud9 IDE, Inc | PRICE \$1/£0.80 per month | URL [c9.io](http://c9.io)

**C**loud9 goes beyond the concept of a simple web IDE, and into the realm of a web coding workspace. While this may sound a little daunting, the interface is easy to use and the experience is quite similar to creating files locally on a computer. It's designed for hobbyists and professionals, so it is powerful, yet accessible. This has the effect of allowing users to

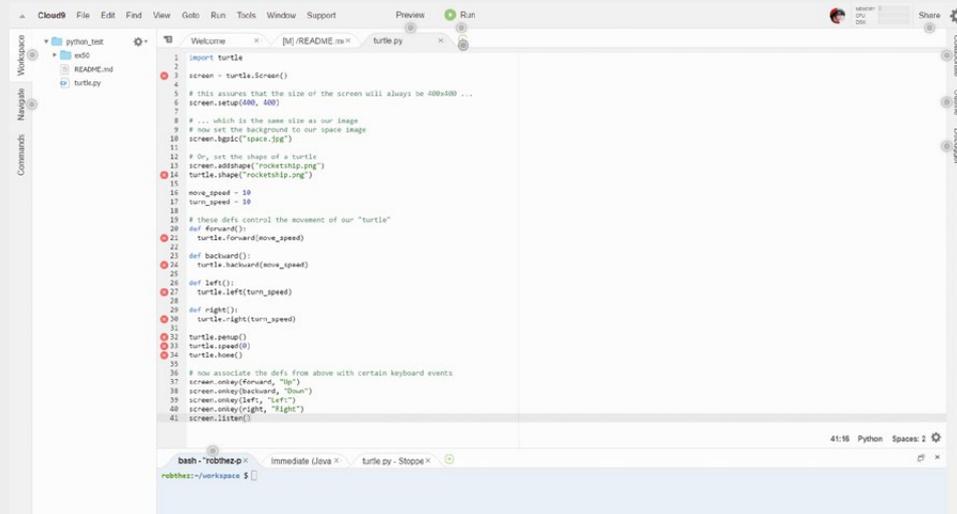
discover new features as they grow, which will potentially encourage students to self-teach as they code.

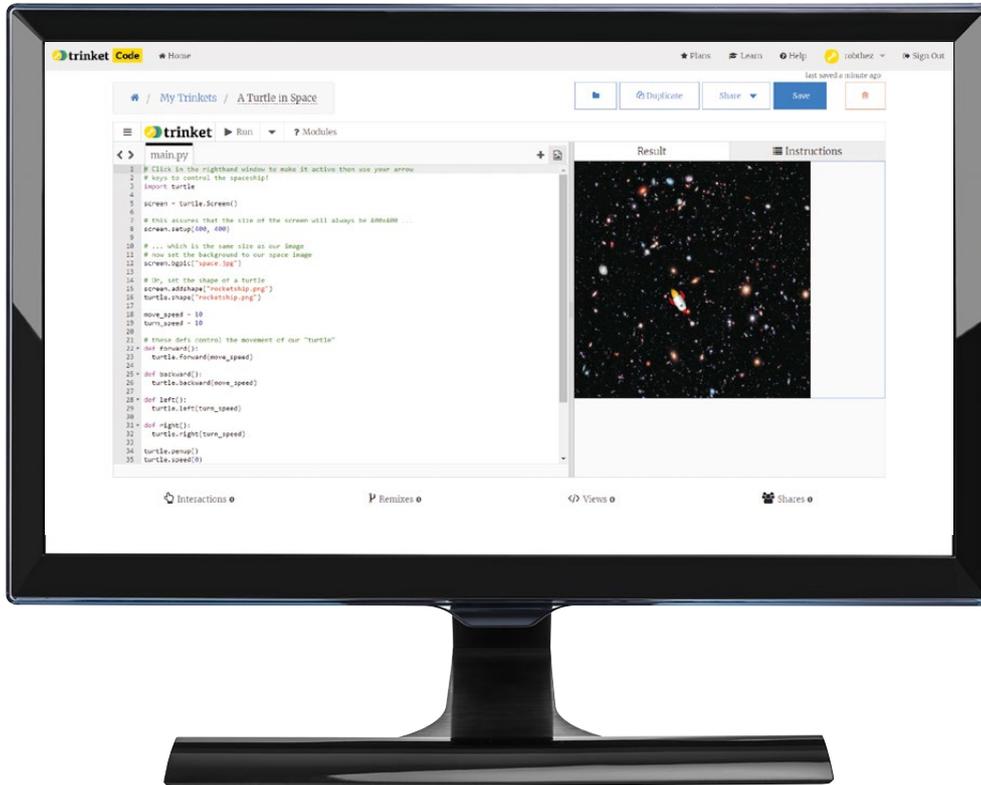
It also has a very interesting educator account option, where a teacher can have unlimited students under their own account. Students don't need a credit card to access the workspaces (as free account users do). Students can then have their own private

workspaces, and interact with any public workspaces created inside or outside the teacher's account. It even hooks up to online repositories such as GitHub, for students who want to learn about more advanced version control and code sharing.

The interface of the IDE itself is great, working very much like a desktop IDE. The tabbed browsing of your files, with a project file system ever-present on the side, makes coding easy. There are loads of unobtrusive prompts as well, to warn you about any issues with the code. Highlighting is very light by default for some languages, compared to most IDEs. However, you can change the theme to something more familiar if you prefer a different look.

It's a good option for slightly older students, or for those wanting to graduate from basic coding exercises. You can also try it out with the free account option to see if it's suitable. **(HW)**





## QUICK FACTS

- Python-focused
- Includes tutorials
- Ability to use blocks instead of code

# TRINKET

**INFO** FROM Trinket | PRICE Free | URL [trinket.io](http://trinket.io)

**W**e like Trinket, because you can just go to the website and test out some code without even creating a (free) account. It is focused on Python, but it is popular with educators, so we thought it was definitely worth checking out as a teaching tool. With a free account (no credit card details required), students can copy, share, and save code to their own profiles, then run it live on the Trinket website to see how it works. The website also includes graphical interfaces for creating games.

Code can be downloaded to then run on a separate computer, although the modules used for the Trinket code will need to be installed on this machine. It will be apparent from the code which modules you'll need. There are lesson plans, a special turtle module for more classic computing teaching, and it can even emulate a Raspberry Pi Sense HAT. Most interestingly, it

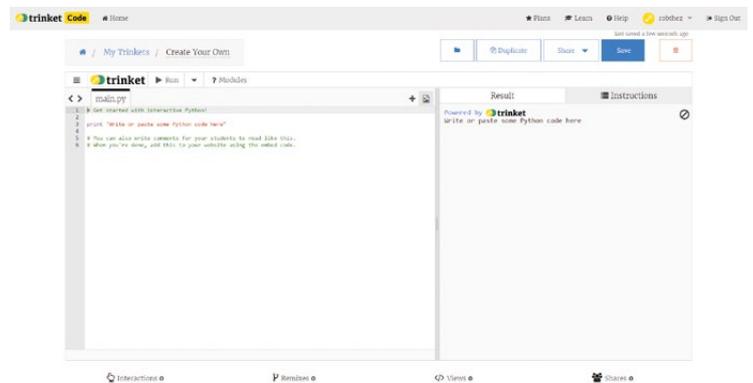
has a special mode that's designed to translate block programming like Scratch into text-based Python programs. It's an important step while learning to code and this is one of the few environments, online or offline, that addresses it. With a paid account, Trinket even supports coding in Python 3.

While Python is the main focus, Trinket has the ability to program

- The layout is simple and easy to work with

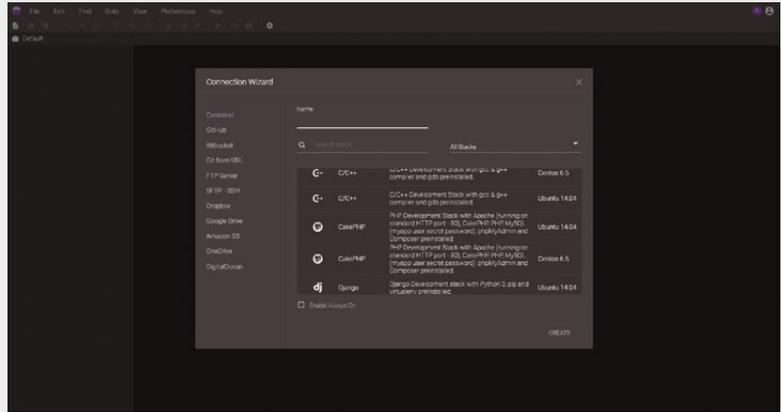
in HTML5, which is very useful for teaching web development. There's also a music creator which uses a simple language to code music, although something like Sonic Pi is much better for linking code and music.

It's simple and fun to use, although it is slightly limited by its focus on Python. The blocks function is a good in-between step for younger students still very used to Scratch. **(HW)**



# CODEANYWHERE

**INFO** FROM Codeanywhere, inc  
 PRICE Varies  
 URL [codeanywhere.com](http://codeanywhere.com)



Codeanywhere is a bit more advanced than your standard IDE. While it does let you program and test code online in the browser like any other IDE, you also have to know what kind of language you're coding in so you can set up a container with the right kind of operating system to run it in. This is great for more advanced or older students who are beyond some of the more basic teaching principles and are interested in making their own programs, as it

gives them a bit more of a taste of being a real developer. There is a free account option (no credit card required), and a discount for educators.

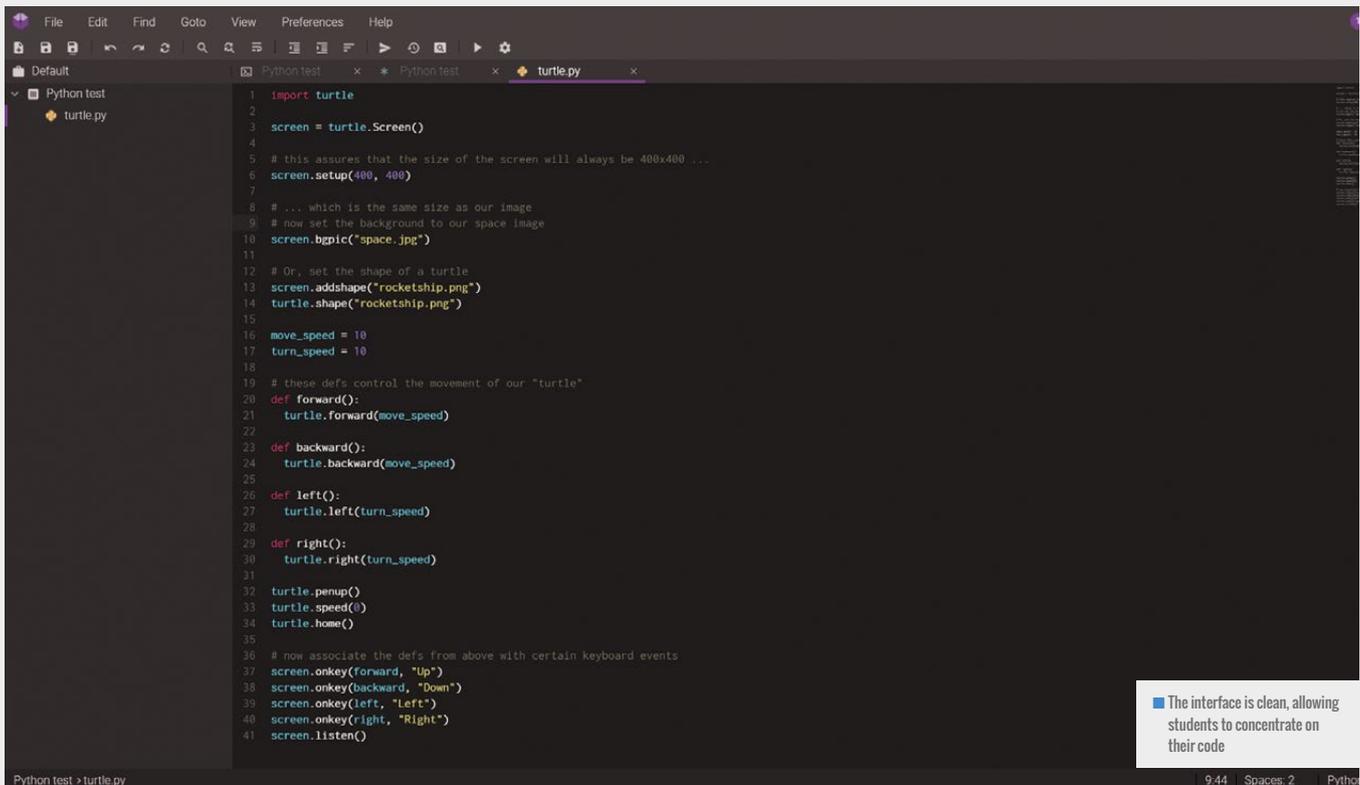
The interface takes some getting used to but it does highlight code, as you would expect from an IDE. You may have to install modules separately to the container you've set up, which can be slightly more tricky to do than on a local computer. Saving any software you create involves either copying the code into a

blank document or saving it to an online repo.

While Codeanywhere is good, it is definitely aimed at more advanced users, and definitely not at school-level learners. Cloud9 does a better job of easing people into a more advanced environment, allowing users to get some code written and running without having to learn about containers. You can customise Codeanywhere by installing modules to the containers, but it's not quite worth the extra hassle. **(HW)**

## QUICK FACTS

- Allows all major code types
- Runs code on online containers
- Connects to GitHub and other repositories



■ The interface is clean, allowing students to concentrate on their code

← my-pi-projects

projects

```

13 x = [1, 2, 3] # a different [1, 2, 3] lis
14 y = x
15 x.append(4)
16 y.append(5)
17 z = [1, 2, 3, 4, 5] # a different list!
18 x.append(6)
19 y.append(7)
20 y = "hello"
21
22
23 def foo(lst):
24     lst.append("hello")
25     bar(lst)
26
27 def bar(myLst):
28     print(myLst)
29

```

Global variables

sleep

x

y "hello"

z

foo

bar

list

list

function sleep(...)

function foo(lst)

function bar(myLst)

foo

lst

You can check out the code flow in a handy diagram mode

### QUICK FACTS

- Runs on Raspberry Pi
- Code saved to the cloud
- Can test code over the network

# ADAFRUIT WEBIDE

**INFO** FROM Adafruit | PRICE Free | URL [learn.adafruit.com/webide/overview](http://learn.adafruit.com/webide/overview)

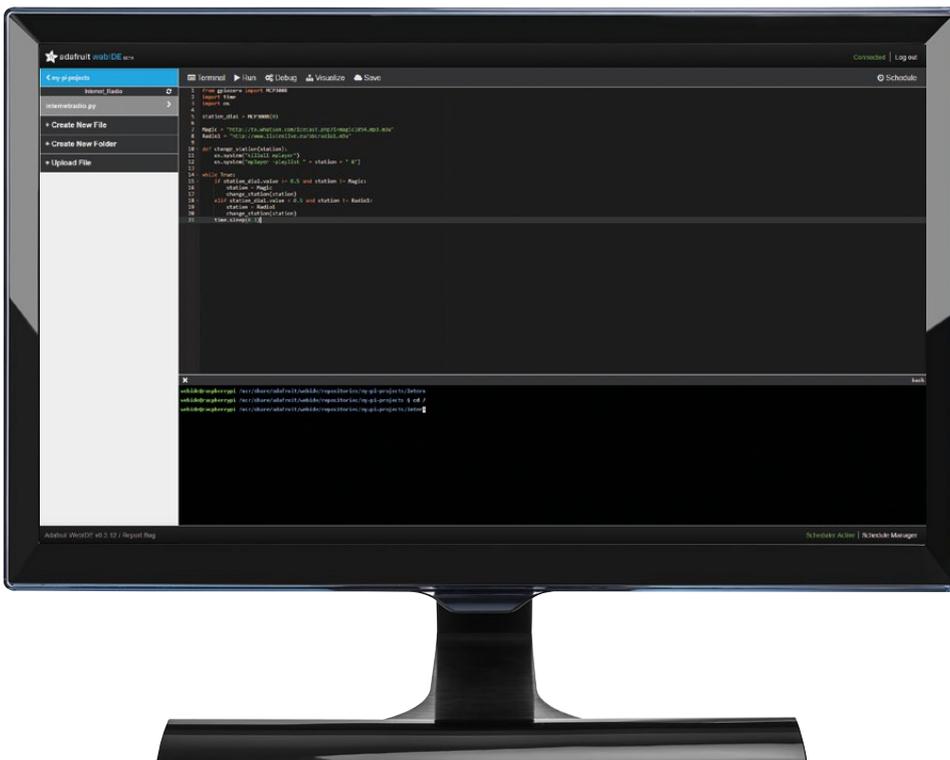
This IDE is slightly different to the others on these pages. While the other IDEs are based solely online, and can be used anywhere with an internet connection, the Adafruit WebIDE is housed on a Raspberry Pi in the local network. Code is all saved on the cloud via a third party, and can then

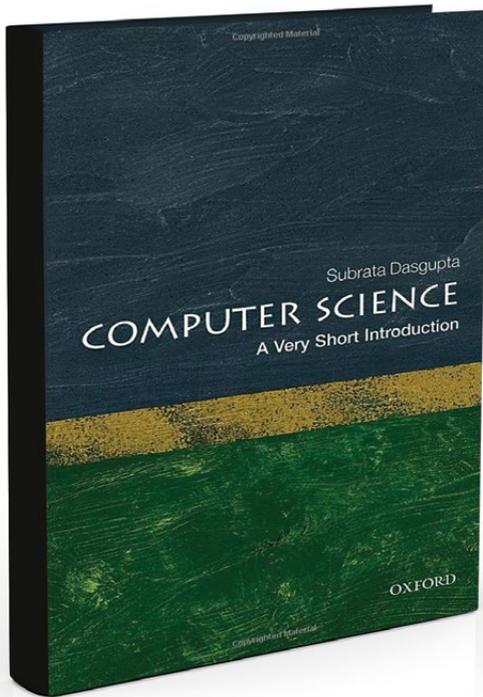
be accessed from anywhere that can connect to the original computer. One of the neat things about this IDE is that the server where you test your code can be set up to have the same software and libraries as the target computer, so it allows you to properly test code you've written using the server.

While it doesn't offer the full freedom of the other IDEs, it does allow for a slightly more closed-off ecosystem, with the code only running and being edited from specific places. That doesn't stop a student from taking the code home and making their own changes, but it does let you upload examples to be accessible as standard from your cloud storage. Students don't have a way to create their own repositories to grab files from, but that's a very different skill from basic coding.

WebIDE is a little limited by the need to have a Raspberry Pi as the main server, although any computer can be used to connect to it - even a mobile device or tablet. Software on school computers can be very tightly controlled, and this it does allow you to get around that to a limited extent.

WebIDE works as well as any offline IDE and has great project management options and testing tools. It definitely offers a unique coding solution. (HW)





# COMPUTER SCIENCE: A VERY SHORT INTRODUCTION

This book is for anyone curious about the science behind our subject: the principles that still apply after the latest language/processor/paradigm is forgotten

## INFO

BY Subrata Dasgupta | PUBLISHER Oxford University Press | PRICE £7.99 | ISBN 9 780198 733461 | URL [helloworld.cc/2pMZzeX](http://helloworld.cc/2pMZzeX)

**D**asgupta starts with the proposition that computer science is a science, but of the artificial, of how things ought to be rather than of what they actually are. For him the stuff of computing is symbols, and so computing (the doing of computer science) is symbol processing.

Computational artefacts are shown as organised in hierarchies, to manage their complexity, and classified as material, abstract, and

introduce algorithms and their implementations, with short introductions to big O notation, the aesthetics of algorithms (a recursive implementation of factorial is more appealing than an iterative one, despite recursion itself not being explained), intractable problems, and a short P = NP? discussion.

Programming is discussed as an art, a science, and as engineering. The Sapir-Whorf hypothesis (language determines and defines

## 'Good enough' algorithms

Heuristic algorithms get their own chapter, and although AI is discussed there's nothing about machine learning: the First Come First Served heuristic is used for parallel scheduling. Much is made of heuristics for satisficing (satisfying and sufficing) and getting a solution that's 'good enough'.

The chapter on computational thinking sees Jeanette Wing's approach as (perhaps) more prosaic than Papert's, but then covers the topic's influence on other fields, although for bioinformatics it is badly explained.

The epilogue asks if CS is a universal science like physics or chemistry, but without coming to any definite conclusion.

The book is an odd mixture, ranging from philosophical discussions to (quickly glossed over) examples of assembly language, and parts seem strangely dated (despite being published in 2016). I didn't feel I learnt a great deal, and I think a non-specialist might get rather confused, so it's difficult to recommend it. (HW)

“ Computational artefacts are shown as organised in hierarchies, to manage their complexity, and classified as material, abstract, and 'liminal'

'liminal' (straddling a boundary). Programs are liminal because they are concrete implementations of abstract algorithms. (Universal) Turing machines get special mention here as the 'great unifier', since a Turing machine can implement all computational artefacts and behaviours.

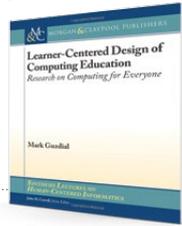
Chapters three and four

thought) is mentioned, although none of the different language paradigms other than procedural is used to illustrate the idea.

Chapter five on computer architecture nicely shows hardware as having inner and outer implementations, although whether this is really relevant to computer science is arguable.

# LEARNER-CENTRED DESIGN OF COMPUTING EDUCATION

Mark Guzdial offers an authoritative survey of the state of the art in CS education research, and recommends educators focus on where the learner is, and where they want to get



**INFO** BY Mark Guzdial | PUBLISHER Morgan & Claypool  
PRICE £46.50 | ISBN 9781627053518 | URL [helloworld.cc/2qzkG7n](http://helloworld.cc/2qzkG7n)

**M**ark Guzdial developed Georgia Tech's highly regarded Media Computation course, and writes a great CS education blog at [computinged.wordpress.com](http://computinged.wordpress.com). He really knows what he's talking about in terms of computing education for all, and this expertise comes through loud and clear in this book.

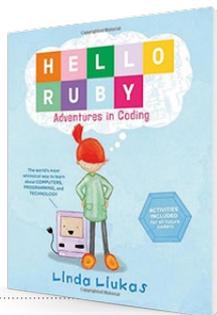
He starts by summarising a number of the arguments explaining why everyone should learn some computing, and gives an overview on why learning to program is hard. There are some critical insights into the debates on 'computational thinking', which Mark sees as

making more sense when linked to computation rather than applied to more general problem solving.

The thrust of Mark's argument in the rest of the book is that teachers need to know where learners are starting from, and where they want to get, before they can design learning. Such a learner-centred approach helps to make computing more engaging, more relevant, and more inclusive. The book is full of practical advice on how to make computing education more effective, drawing on Mark's authoritative grasp of the state of the art in CS education research. **(HW)**

# HELLO RUBY

A beautifully illustrated, captivating introduction to some of the big ideas in computing for young children, plus playful, unplugged ideas to explore them



**INFO** BY Linda Luukas | PUBLISHER Feiwel & Friends  
PRICE £13.23 | ISBN 9781250065001 | URL [helloruby.com](http://helloruby.com)

**T**his is a beautiful book! At first glance, Hello Ruby is an engaging children's story. Ruby is a small girl with curiosity, imagination, and a tendency to take things a little too literally. Her father leaves her a set of clues which set her on a quest to recover five gems. On the way Ruby meets some friends: some eccentric penguins, given to communicating in short sentences; a beautiful snow leopard; some

semi-anarchic foxes; androids that love cooking; and Django and his pet python. Ruby learns about the importance of planning, of breaking big problems down, and of giving really precise instructions.

The second part of the book, and the [helloruby.com](http://helloruby.com) site, provide some original unplugged and computational thinking activities, well suited to the book's target audience of four- to eight-year-olds. **(HW)**

## ESSENTIAL READING:

A few newly published titles – no actual coding here, but plenty to think about!

### DETECTIVE DOT

BY Sophie Deen and her friends  
PUBLISHER Bright Little Labs  
PRICE £12.99  
ISBN 9780995636200  
URL [detectivedot.org](http://detectivedot.org)

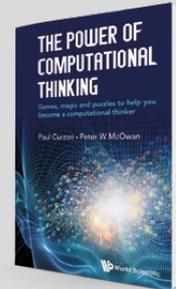
Dot is a nine year old geek, working part time for the Children's Intelligence Agency. She's equipped with courage, curiosity, and the ability to hear tech talking. Sophie's story teaches some coding ideas, and gets children thinking about how consumer goods get made.



### THE POWER OF COMPUTATIONAL THINKING

BY Paul Curzon and Peter W McOwan  
PUBLISHER World Scientific  
PRICE £20  
ISBN 9781786341846  
URL [helloworld.cc/2qHrkWc](http://helloworld.cc/2qHrkWc)

Paul and Peter draw together in one volume many of the computational thinking puzzles and ideas they've explored through Queen Mary University of London's CS4FN magazine. Some inspirational material here.



### HACKING THE CURRICULUM: CREATIVE COMPUTING AND THE POWER OF PLAY

BY Ian Livingstone and Shahneila Saeed  
PUBLISHER John Catt Educational  
PRICE £14.00  
ISBN 9781909717824  
URL [helloworld.cc/2q6tguw](http://helloworld.cc/2q6tguw)

Ian and Shahneila discuss computing in schools, and how play, game-based learning, and game design can offer particularly effective approaches to developing pupils' skills and knowledge.





Participants work in teams to develop their response to a given challenge

# ENGAGE YOUR COMMUNITY WITH A HACK JAM

Hack Jams support teacher development and provide inspiring and engaging experiences for students

STORY BY Alan O'Donohoe

**W**ith all the external pressures that teachers are under (curriculum reform, performance-related pay, and school league table comparisons to name a few), teaching currently feels like a very high-stakes profession. Can we really expect teachers to take risks, experiment, and be more creative when such great emphasis is placed on national tests and exam performance?

Hack Jam events provide teachers with a sandbox learning environment, where they can feel free to innovate and experiment with new resources and approaches to teaching and learning. They provide learners with a social experience that facilitates access to fun, stimulating, engaging challenges:

activities that they might not normally encounter in a timetabled lesson, without fear of failure. At the same time, parents and other adults are able to support these learning activities without the

Hack Jams I have taken part in gave me some great ideas. I found that I was able to incorporate the best features of the Hack Jams into my everyday teaching, for the benefit of all my students.

**“ A SANDBOX LEARNING ENVIRONMENT, WHERE THEY CAN FEEL FREE TO INNOVATE AND EXPERIMENT**

usual barriers or constraints of a normal school or business day.

If you're looking to put the fun back into learning, then a Hack Jam will absolutely help you to achieve that. Better still, the

## What happens at a Hack Jam?

The Hack Jams that I've organised on behalf of exa.foundation typically start in the late afternoon or early evening. They last for two to three hours, are free to

## RESOURCING A HACK JAM

A Hack Jam might focus on creating with:

- micro-bit
- Raspberry Pi
- Physical computing
- CodeBug
- Scratch
- Python
- Minecraft
- Mozilla Webmaker

A collection of Hack Jam resources can be found at [helloworld.cc/2p1jwwG](http://helloworld.cc/2p1jwwG)



■ Physical computing: mother and son collaborate to create an alarm on a breadboard controlled by a Raspberry Pi

attend, and attract between six and 60 children and adults with a wide range of digital making experience. Occasionally I've organised Hack Jams with one specific purpose in mind: to heighten awareness of an important issue, or to target certain groups, such as learners from a particular background, or under-represented group.

To allow for a wide variety of experience in the group, I like to start Hack Jam events with a 'mini-teach' introduction, lasting

around 20 minutes. I use this time to demonstrate one simple example showing how the project resource can be used.

Following the mini-teach, all participants work in pairs or teams to develop their response or solution to the challenge they have been set. Throughout the event, additional activities, and mini-

■ "I like to start Hack Jam events with a 'mini-teach' introduction, lasting around 20 minutes. I use this time to demonstrate one simple example."

challenges are set to encourage sharing and collaboration. Sometimes we include inspiring lightning talks during the course of the Hack Jam.

Just before the conclusion of the event, all the teams take part in a joint show-and-tell activity. They exhibit their creations to other participants, who move around the space to visit the different exhibits.

### Organising a Hack Jam?

■ **Investigate:** If you're considering hosting your own Hack Jam, start by visiting one, reading more about them online, and contacting other Hack Jam hosts to find out what they think works best. No two Hack Jam events are the same.

■ **Commitment:** Seriously commit yourself to a future date on the calendar, and invite one or two local teacher friends and a handful of their students to a small Hack Jam at your school or library. Don't worry too much about prizes or refreshments: these are nice to have, but not essential to the success of your event. A BYOD (Bring Your Own Device) Hack Jam reduces the stress on the host to provide the necessary kit or devices.

■ **Focus:** Identify a specific group you would like to target with a Hack Jam event. Don't worry too much about the challenge activity, and never give up after the first one – the first event is always the hardest. Instead, make Hack Jams a regular occurrence on your calendar. After three Hack Jams you'll have cracked it, and you'll start to see the benefit in your own teaching practice. (Hw)





GREG MICHAELSON PROFESSOR OF COMPUTER SCIENCE

# TOO MUCH CHOICE: SELECTING INFORMATION STRUCTURES IN PROBLEM SOLVING

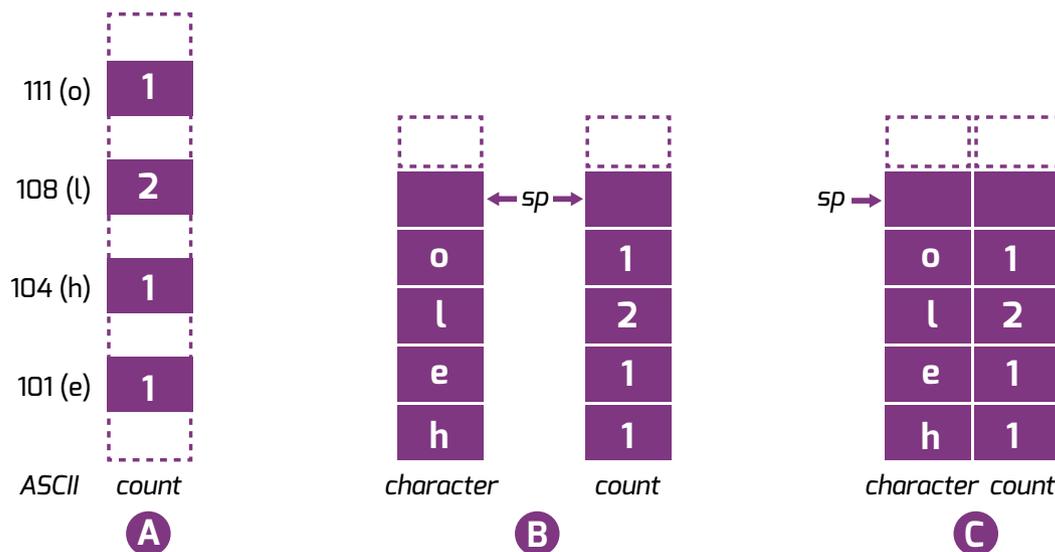
Even for simple problems, there is often a wide range of alternative information structures, and it's not always clear which might be best

To explore this issue, we're going to look at counting how often each unique character occurs in a file. We'll consider six possible structures, and discuss how easy they are to construct and access, and how much memory they require. We'll suppose that our test file starts: 'hello...'

### Count array

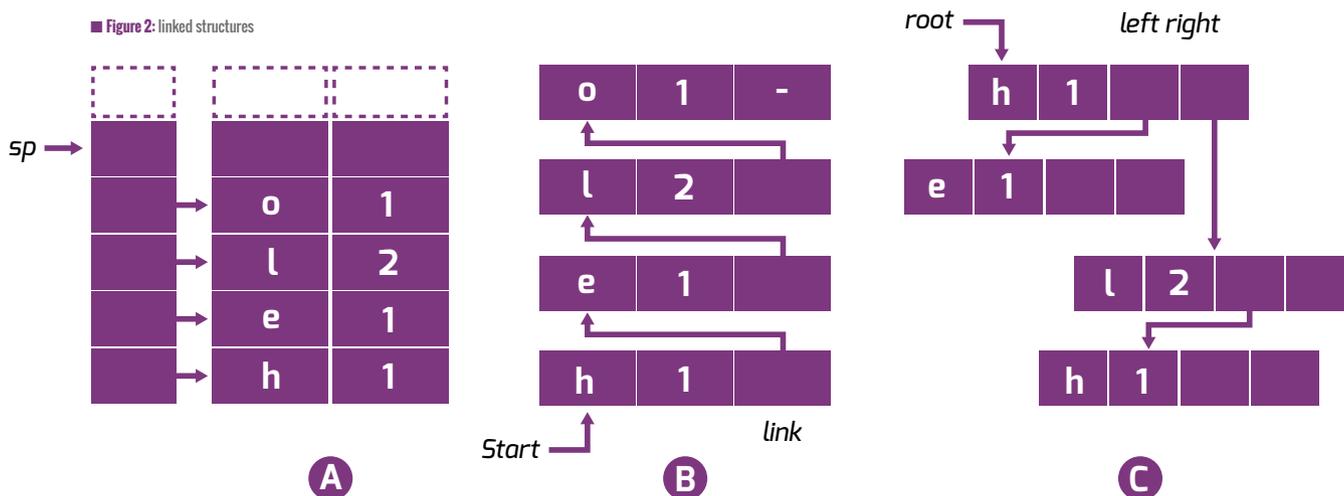
To begin with, let's work with 1-byte, i.e. 8-bit, characters. We'll define an array of counts with 256 ( $2^8$ ) elements, which we'll initialise to 0. See Fig. 1(A). Then, as we input each character, we'll convert it to an integer, say the ASCII value, and add one to the element with that index. Our access time is constant.

Figure 1: array, stacks, and records



Counting unique characters with A direct index, B parallel array stacks & C stack of records

Figure 2: linked structures



Counting unique characters with A stack of pointers, B linked list & C binary tree

Suppose, though, that we decide to use the Unicode standard of 4 bytes, i.e. 32 bits. Then we need to allocate space for 4GB (2<sup>32</sup>) elements. This seems profligate: for many languages, almost all of this space won't be needed.

### Stacks and records

Instead, let's use parallel arrays as stacks of characters and counts. See Fig. 1(B). For a new character, we'll search the stack from bottom to top. If we find the character, we'll increment the corresponding count. Otherwise, we'll push the new character onto one stack and a 1 count onto the other. We can pre-allocate space for relatively small stacks, provided we check for overflow, but may we still waste some space. The worst case access time is now linear as we may have to check every entry.

Alternatively, we could have a single stack of records of characters and counts. See Fig. 1(C). This has the same properties as the parallel arrays.

### Linked structures

Yet another approach is to have a stack of links to records. See Fig. 2(A). Here, we allocate records dynamically for each unique character. We may waste less space as we'll

pre-allocate a stack for links, not whole records. But a full stack of links plus records will require more space than the same stack of records alone.

Why not dispense with arrays altogether, and use a linked list of records? See Fig. 2(B). We'll search the linked list from the start and add new nodes dynamically at the end. The access time is still linear, but now we allocate only the space we need. However, compared with a full stack of records, we'll again allocate more space because of the links.

Last, we could build a binary tree. See Fig. 2(C). We'll need space for twice as many branches as links in the list, but, for a balanced tree, access is faster as we halve the search possibilities on each node.

So, which should we choose? Well, as the Fig. 3 table shows, there's a classic time/space trade-off: faster access requires more memory. But, as Dr Who knows, time and space aren't everything... (HW)

**Greg Michaelson** is interested in the design, implementation and analysis of programming languages, especially for multiprocessors. He has taught programming for nearly 40 years.

Figure 3: the time/space trade-off

Structure with N entries	Time	Space
Array	O(1)	4GB × integer
Parallel stacks	O(N)	N × (character + integer)
Stack of records	O(N)	N × (character + integer)
Stack of record links	O(N)	N × (character + integer + link)
Linked list	O(N)	N × (character + integer + link)
Binary tree (balanced)	O(log <sub>2</sub> N)	N × (character + integer + 2 × link)



■ Ben Nuttall leads a workshop at the Manchester Raspberry Jam

# EVENTFUL INSPIRATION

A guide to running your successful tech-based event, packed with advice from people who are already running theirs

**H**ave you ever thought about running a hackathon, TeachMeet, or Jam? Getting like-minded people together to encourage digital making, coding, and community? The Raspberry Pi Foundation has just produced a guide to running your first event. *The Raspberry Jam Guidebook* is aimed at Jam organisers, but the advice it offers will be helpful to anyone setting out to run a tech-based community event.

## What is a Raspberry Jam?

Raspberry Jams are independently organised community events where people get together to share knowledge, learn new things, and meet other Raspberry Pi enthusiasts. Jams provide opportunities for people to get involved in digital making, develop their abilities, get together, have fun, and socialise. They are usually free or very cheap to attend.

Raspberry Jams are something of a phenomenon. Run by volunteers, they gather together people from all walks of life to celebrate the Raspberry Pi in particular, and digital making in general. They have spread throughout the world, from humble beginnings in the north of England to the Far East and North America, and they show no signs of slowing down.

Ben Nuttall, Community Manager at the Raspberry Pi Foundation, has put together the guide to running your own event. He has collected advice from Jam organisers throughout the UK, on everything from finding a venue to managing your finances, and from planning your activities to managing social media. Packed full of great, first-hand advice, the guidebook is designed to help you to run the best event you possibly can.

## Getting Started

Every tech-based community event has a different flavour, every event attracts a different group of people, and every event will have a different buzz about it. As an event organiser, you need enthusiasm and a desire to ask those vital questions: where will I hold my event, and what will it look like?

The guidebook is here to help. Over the next few issues, Hello World will bring you advice and inspiration from the guidebook to help with the planning of your community event.

"DON'T BE AFRAID TO START SMALL: IT HELPS YOU TO GAUGE INTEREST. TRY SOMEWHERE OUT AND FIND OUT WHAT SHAPE YOUR EVENT WILL EVENTUALLY BE."

**MIKE HORNE**  
CAMBRIDGE RASPBERRY JAM



## THE RASPBERRY JAM GUIDEBOOK

The Raspberry Jam Guidebook is provided by the Raspberry Pi Foundation to help members of the community to set up and run Raspberry Jam events. Raspberry Jam events are organised and owned by the community. They can be started by anyone who is enthusiastic about digital making. The Raspberry Pi Foundation is recommending that Jam organisers follow these guidelines, which have been gathered from community members running events in the UK. If you are running an event in another country then it is your responsibility to research and follow the legislation that governs your region.



### Building your team

Running a community event should not be a one-person operation. If you are working on this alone, try to find some helpers. Look for other people interested in running a tech-based event, and involve them in the planning stages. Ask around for volunteers

### Finding a venue

Finding a venue is an important step in the planning of your event. If this is your first event, you don't need to find a permanent venue. If you run further events, and you find that you need a bigger venue in the future, you can look for one then. Many

## “ FINDING A VENUE IS AN IMPORTANT STEP IN THE PLANNING OF YOUR EVENT ”

to help on the day, or to help set up the venue before the event. People will be willing to help: they just need an invitation. Find jobs for your helpers, ask them for their advice to help you make decisions, and involve them in coordinating the event.

“TO FIND ADULTS TO HELP, WE STARTED LOOKING AT LINUX GROUPS, TECHY COMPANIES, AND DEVELOPER MEETUPS. HAM RADIO GROUPS ARE USUALLY INTERESTED, AND HAVE GOOD SKILLS TOO. WE ALSO GET ALL THE CHILDREN TO BRING THEIR PARENTS: SOME OF THE PARENTS ARE QUITE TECHY, AND WILLING TO HELP.”

**ANDY MELDER**  
SOUTHEND RASPBERRY JAM

regular community events begin with a small number of people, and grow into larger gatherings later. If you can, choose a venue that is easy to find, easy to reach by public transport and accessible to people who use mobility aids.

When you are looking for a venue, consider local libraries, makerspaces, schools, colleges, and universities. Try to find out where other community groups in your area meet up. Use websites such as Eventbrite and Meetup to research suitable venues.

If you can, find a venue that will be free of charge to use. Some events do pay for their venues, but this requires fundraising and can be hard to sustain. If the venue usually

charges for bookings, they may allow you to hold your event free of charge if you explain that it is a community meeting, rather than a commercial event. Alternatively, you could offer to donate all proceeds from your event to the venue to help cover their costs. You could charge a small amount for tickets, or ask for donations on the day. You can find more information on issuing tickets in chapter two of the guidebook, or later in this article. ▶

“OUR LOCAL LIBRARY GAVE US A VENUE FREE OF CHARGE: THEY'RE JUST PLEASED THAT SOMEONE IS USING THE SPACE.”

**CAT LAMIN**  
WIMBLEDON RASPBERRY JAM

■ Consider universities and colleges when looking for a venue



### ▶ Naming your event

Choosing a name for your event is an important part of your planning process. It is best to include your location in the name, but think carefully about the area you serve or represent. Don't be too broad. For example, calling yourself 'London Techmeet' would make it look as if you are the only techmeet in London. If you are running a regular event, be consistent in the use of your name.

Once you have decided on a name, it is a good idea to set up an email address for your event. You can use this to answer queries from attendees and to set up

accounts with Eventbrite and social media platforms. Getting your email address set up now will save you time when you come to issue tickets and promote your event.

### Support

Remember: when you are organising a community event, you are not alone. There are many other people in the digital making and coding communities with experience of running events. Reach out to them: ask for help, share your ideas, and ask for advice.

It can be very helpful to meet people who are already running successful events,

own event. Give yourself a chance to ask questions and share your ideas with them. Don't fixate on trying to match exactly what their event does: do what you think is suitable and manageable for you. To find a local Raspberry Jam to attend, take a look at the Jam map on the Raspberry Pi website. For CodePen and CoderDojo events, check out their websites.

Don't forget to look for support online. The Raspberry Pi Foundation forums are full of discussions on topics including programming, hardware and projects. There is a forum for user groups and

## SUPPORT LINKS

Eventbrite:  
[eventbrite.com](http://eventbrite.com)

Meetup:  
[meetup.com](http://meetup.com)

Raspberry Jam map:  
[rpf.io/jam](http://rpf.io/jam)

CoderDojo map:  
[zen.coderdojo.com](http://zen.coderdojo.com)

CodePen events:  
[helloworld.cc/2qcfcw](http://helloworld.cc/2qcfcw)

Raspberry Pi Foundation forums:  
[rpf.io/forums](http://rpf.io/forums)

“REMEMBER: WHEN YOU ARE ORGANISING A COMMUNITY EVENT, YOU ARE NOT ALONE.”

especially when you are getting started. Make an effort to find out what else is going on in your area. Are there any local Code Clubs? Is there a CoderDojo or similar organisation? Is there a local Raspberry Jam? Is there a hackerspace, or a tech community space? Are there tech user groups? Reach out to the people behind these events. Can you work together to promote your event? Can you make use of each other's resources or experience?

Find another tech-based community event to attend. Contact the organiser and explain that you are planning to start your

events, which could help you to find local sources of support in your area. The Jam community uses a Slack team to share ideas and support. Other communities may offer similar central points of contact, or you could consider starting your own.

“WE TEND TO VISIT OTHER EVENTS TO SEE WHAT INSPIRES US, WE THEN APPROACH THAT PERSON TO COME AND RUN THE SAME WORKSHOP AT OUR JAM.”

MARCUS TYLER-MOOR  
HORSHAM RASPBERRY JAM

## Lead Times

People who run big community events usually give themselves a couple of months to plan everything, as there is a lot to organise when you are arranging talks, workshops, and volunteers. Smaller events with fewer activities do not take as much planning, but it is important to give yourself plenty of time for everything to come together, and to deal with contingencies. Most organisers make their tickets available a month or so before each event. This gives people the chance to plan their attendance in advance, and lets the organisers know how many attendees to expect.

If you are planning to run a repeating event, it is helpful if you are able to keep it to a set day, for example the first Saturday of each month. This allows people to plan ahead, and ensures that they know when the event will take place. If possible, book your regular date with your venue in advance. Make the dates available in advance to your attendees, on your website or by email.

■ Your community event could provide digital making and physical computing opportunities to people of all ages

## Raspberry Jam starter kit

If you are planning a Raspberry Jam, don't forget to request your Raspberry Jam starter kit from the Raspberry Pi Foundation.

Your kit will contain flyers, sets of stickers, magazines, printed worksheets, and more! You will also receive a pack of design assets and templates to help you make your own posters and other resources. Head to [rpf.io/jamkit](http://rpf.io/jamkit) to request your kit.

"OUR JAM IN CAMBRIDGE STARTED SMALL, WITH JUST 30 PEOPLE IN ONE ROOM. IT HAS NOW GROWN TO ROUGHLY 150-200 PEOPLE PER EVENT. IT HAS TAKEN A LOT OF HARD WORK AND DEDICATION TO GET IT TO THIS SIZE. WE HAVE DEVELOPED OUR JAM INTO OTHER EVENTS: PI WARS, A ROBOTICS COMPETITION, WHICH THIS YEAR TOOK PLACE OVER TWO DAYS; AND POTTON PI AND PINTS, WHICH IS A MUCH SMALLER EVENT HELD UPSTAIRS IN A PUB. ALL OF THESE MODELS WORK, AS DO MANY OTHERS. THERE ARE JAMS TAKING PLACE AT WORK VENUES, UNIVERSITIES, LIBRARIES, HACKERSPACES, CONFERENCE CENTRES, AND EVEN MUSEUMS."

**MIKE HORNE**  
CAMBRIDGE RASPBERRY JAM

## TICKETING WITH EVENTBRITE

01. Set up an Eventbrite account for your event
02. Create your event
03. Create tickets
04. Make your event live
05. Share your ticket link online

## Ticketing

It is a good idea to set up a ticketing system for people attending your event. This allows you to keep track of the number of people planning to attend, keep in contact with them before and after the event, and manage attendance on the day. Many events use Eventbrite for their ticketing: take a look at the Eventbrite pages of the Jams on the Raspberry Jam map to see how the process works. Eventbrite can be used for ticketing a free event, but it also allows you to sell tickets.

It is easy to set up an Eventbrite page and start publicising your event. Start by creating an email account for your event, if you haven't done this already. Use this >



➤ to create an Eventbrite account. Create an event; provide the date, time, location and other information; and create some tickets. If appropriate, you could provide adult tickets and child tickets, or morning and afternoon tickets. Eventbrite allows your attendees to choose between registering for a free ticket, or paying a small donation to the event.

account, consider the number of people you are expecting to attend, and work out how much you need to charge.

For child-friendly community events, consider providing free tickets to children, and covering your costs by charging for adult tickets. You could also make free tickets available to students, and to out-

take the form of a 'pay what you want' ticketing system, a donation option on your Eventbrite page, or a collection at the event.

## “ FOR CHILD-FRIENDLY COMMUNITY EVENTS, CONSIDER PROVIDING FREE TICKETS TO CHILDREN

Your Eventbrite page could function as the home page for your first event. For further information, see Eventbrite's getting started guide: [helloworld.cc/2r4Om9P](http://helloworld.cc/2r4Om9P).

### Pricing

Ideally your event will be free to attend for everyone, but this is not always possible. If you have costs to cover, you will need to find a way to raise money. Charging a small amount for tickets is a fair way of covering your costs. Take any costs you incur into

of-work adults. Finding a way to balance freebies with paid tickets can be tricky, but free tickets can dramatically lower the barrier to entry for some groups of people.

Consider running your first event in a free venue, and moving to another venue later if necessary. Raspberry Jams in the UK often charge around £5 for tickets, but you will know what is right for your event, your region and your community.

As an alternative to charging for tickets, you could ask for donations. This could

WHEN PROMOTING YOUR EVENT, KEEP IT LOCAL. A LOCAL RADIO OR NEWSPAPER FEATURE IS WORTH A DOZEN INTERNATIONAL TV SLOTS. NOBODY WATCHING IN CALIFORNIA IS GOING TO MAKE THE TRIP TO CHELTENHAM, WHEREAS QUITE A LOT OF MUMS AND DADS READING THE GLOUCESTERSHIRE ECHO MIGHT TAKE THEIR KIDS ALONG TO OUR EVENTS.

ANDREW OAKLEY  
COTSWOLD JAM

### Promotion

When you have set up your ticketing page, you need to tell people about it. Think about who you want to attend your event, who might want to come, and how you can reach them.

If you are running a Raspberry Jam, the first step is to submit your event to the Raspberry Pi website. If it is a Raspberry Pi-focused event that is free or cheap to attend, it will appear on the Jam map

Free tickets for children will lower the barriers to entry for families



and in the calendar. Submit your Jam at [rpf.io/addjam](http://rpf.io/addjam).

Make use of social media to promote your event. Set up a Twitter account for your event and tweet about it. Contact the organisers of other local tech events and ask them to spread the word by tweeting about your event, or promoting it to people on their mailing list. Look for CoderDojos, Code Clubs, hackerspaces, user groups, and other meetups.

Create a mailing list for your event using MailChimp or similar. Invite people to register for updates about your events. Tell your local school, college, or university about your event, and ask them to share the details with their students.

Publicise your event in local newspapers, on community boards in shops and libraries, on local news websites, and on local event listing websites. Ask your venue to promote your event.

## Kit

If you are planning to run practical activities at your event, it is a good idea to provide some basic kit. You don't have to buy any kit to get started. You can ask attendees to bring their own kit from home, or you could contact the organisers of other events and ask to borrow their kit for your first event. If your venue offers PC suites, this may be all that you need to provide. If you are using Raspberry Pis or other single board computers, use the PC suite monitors and ask people to bring their own SBCs and peripherals.

As an example, each participant in a practical Raspberry Pi-based activity will probably need:

- Raspberry Pi 3
- SD card
- power supply
- monitor
- HDMI cable
- USB mouse and keyboard

For specific activities, you could also provide electronics kit, Camera Modules, and HATS.

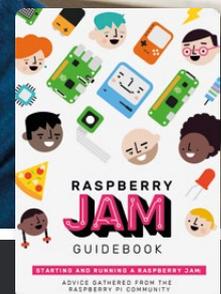
## Next time

In our next issue, we will take a detailed look at managing activities for tech-based community events: planning, kit, staffing, and some tips and ideas from current event organisers. [\(HAW\)](#)



**DOWNLOAD**

Download your copy of the Raspberry Jam Guidebook from: [helloworld.cc/2p7ZX98](http://helloworld.cc/2p7ZX98)





■ Ellora helping others on their tech journeys.

# FROM IDEA TO REALITY: CREATING THE ENVIROCACHE APP

Ellora is 16 years old and a student at Wick High School in Caithness, Scotland. In 2014, she took the Apps for Good course at her school, creating the Envirocache App. Here, she explains how she's launched her own technology company

..... **STORY BY** Ellora from Wick High School in Caithness, Scotland .....

**I**'ve always had an interest in technology, but computing wasn't taught in primary school so I never had the opportunity to pursue it further. We had computing in the first three years of high school, and while I enjoyed it, it was never something I thought of pursuing as a career. When I started the Apps for Good course at school and started doing some proper coding on the Raspberry Pi, I started to realise that I actually really enjoyed coding. That's also when I started to think that maybe coding is something I'm enjoying more now than I did

previously. I always thought I was going to be a vet. When I found out how much I enjoyed coding, I thought to myself, "hang on a minute, I could do this instead."

Right now I'm studying higher computing, where I'm learning Python and PHP. My next step is to start looking more at the electronics side of things. I have just finished making another Raspberry Pi alarm clock, since I managed to break my first one. I definitely want to study computing at university. I don't know what I'll end up doing yet because there's so much, and right now I like everything.

## **Making Envirocache happen**

Ever since I saw the names of successful Apps for Good entrants in our local newspaper, I wanted to take part in the course. Our team spent months searching for an amazing idea before settling on the plan for our app. Living in the countryside, we knew how important it was for young people to spend more time outside, rather than indoors on gadgets. We decided to see how we could use technology to help connect people with the outdoors, which is how Envirocache was born. The app allows people to upload walking routes with

## APPS FOR GOOD

Apps for Good is a charity that works with schools and learning centres to teach young people to build technology products to solve problems they care about. The programme is free of charge to all non-fee-paying schools and centres, and teachers get full access to a dashboard with online resources, lesson plans and CPD support. To learn more visit [appsforgood.org](http://appsforgood.org)

tagged plants and wildlife and interesting facts about them. Children and their families can then walk the routes, seeing what they can spot and learn along the way, like a nature treasure hunt.

While our idea wasn't selected for the Apps for Good UK competition, that was definitely not the end. When co-founder Mari-Ann and I first decided to continue work on Envirocache, we had no idea where to start. We spoke to quite a few people, including our teacher and the team at Apps for Good. We eventually ended up in contact with Trudy Morris from the local Chamber of Commerce in Caithness. She helped us with our business plan, and helped us to set Envirocache up as a company. Her support was invaluable, as



### Showing what's possible

I do want to be a role model for others. That's part of the reason for setting up my blog. I'm getting all of these cool tech opportunities: I did work experience at Thomson Reuters,

she wanted to apply, I thought that was great. I loved sharing my experience and encouraging her to have the confidence to do it herself.

### The role of the teacher

My teacher from Wick High School, Chris Aitken, has been the one to help with a lot of the things I've done. He supported me to take part in Apps for Good in my own time, let me borrow a Raspberry Pi to take home, and helped organise my work experience at Thomson Reuters among other things. All these opportunities have genuinely made me the person I am today, because they've helped build my confidence. It's very important to have someone who will help you apply for things, and give you that extra encouragement.

I think more young people need to be encouraged to access opportunities in technology, because the more you do, the more you believe in yourself. When you believe in yourself, you allow yourself to get involved with more and it has a knock-on effect. If more opportunities existed, it would be even more common for young people who have great ideas to aspire to do great things with technology. (Hw)

## “ IN MARCH I WON THE ONE TO WATCH CATEGORY AT THE 2017 FDM EVERYWOMAN IN TECHNOLOGY AWARDS

these are all things that we wouldn't have had a clue about by ourselves.

Trudy put us in touch with a team at Robert Gordon University, and the Envirocache project was funded by an Innovation Voucher. I'm happy to say that we'll be starting app development really soon. It's something we've been working on for a long time, so I'm super excited. Seeing our idea come to life and knowing that all our hard work will have paid off when we finally launch this year is going to be amazing. We are both so passionate about our idea, so our reward will simply be seeing people use and enjoy the app we have created.

attended a Raspberry Pi Day, spoke at the BETT conference, and I get to travel to different Apps for Good events. In March I won the One To Watch category at the 2017 FDM Everywoman in Technology Awards. I'm always trying to put myself out there and show others with Envirocache that it's possible to make your ideas happen. If I can show even one other young person that things like that are possible, I would have achieved what I set out to do.

Recently a younger Apps for Good Fellow asked me about an opportunity I attended with Stemettes called Outbox Incubator. I loved it because it meant that she read my blog! When she said

# “HELLO, WORLD!”

Everything you need to know about the new computing and digital making magazine for educators...

## Q WHAT IS HELLO WORLD?

**A** Hello World magazine is a new 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). Hello World is sponsored by BT.

## 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 inside and outside the classroom.

## Q WHEN IS IT AVAILABLE?

**A** Your new 100-page magazine will be available three times per year in time for each new term in January, April, and September. Would you like it to be available more frequently? Let us know!



## IT'S FREE!

Hello World is free now and forever as a Creative Commons PDF download. You can download every issue from [helloworld.cc](http://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 for a FAQ section or a bugbear you'd like to share? We'll feature your thoughts and ideas.

- **Tell us your story**

Have you had a recent success (or failure) you think the wider community would benefit from hearing? We'd like to share it.

- **Write for the magazine**

Do you have an interesting article idea? We'd love to hear from you.

---

## GET IN TOUCH

Want to talk? You can reach us at:  
**[contact@helloworld.cc](mailto:contact@helloworld.cc)**

---

## FIND US ONLINE

**[www.helloworld.cc](http://www.helloworld.cc)**

 **[@HelloWorld\\_Edu](https://twitter.com>HelloWorld_Edu)**

 **[fb.com/HelloWorldEduMag](https://fb.com/HelloWorldEduMag)**

**SUBSCRIBE  
IN PRINT  
TODAY!**

**PAGES 28-29**



**MILES BERRY** PRINCIPAL LECTURER

# PROJECT WORK: DOES IT WORK?

Miles Berry questions whether GCSE coursework is fit for purpose, and proposes an engaging alternative

I'm a firm believer that computing is a practical subject, having at least as much in common with art, design and technology, and music as it does with mathematics and the (other) sciences. Papert's great insight, that we learn best through making things to show to others, seems to apply to computing as much as any other subject. I believe, although I've not as much evidence as I'd like for this, that practical programming projects are the best way to learn the fundamental principles of computer science.

## The problems with the NEA

All that said, I'm far from convinced that the NEA (non-examination assessment, i.e. coursework) components of GCSE computer science are really what any of us would want. There are a number of problems, for example: the tasks are far from authentic or interesting, and allow little scope for creativity. Teachers and pupils find themselves in morally conflicted positions, due to the artificial conditions imposed on the conduct of the assessment. Even the exam boards have doubts over the reliability of this component of the the assessment. Computing is the only English Baccalaureate subject with an NEA, and large numbers of independent schools (who have the choice) choose to enter students for IGCSE computer science, which is assessed through exams only, rather than the GCSE.

## Learning from the other sciences

What then can be done? Similar issues arose over assessment of practical work for GCSE science; their solution might work for us too. For the other sciences, there are a list of required practical tasks, which pupils have to undertake, and which the school has to confirm

to the exam board have been completed at a required standard. These don't count for grades, but the concepts and practical techniques covered in these tasks provide content for written questions on the exam papers. For computer science, we could imagine a number of specific programming challenges (e.g. write a program to sort a list of a million 32-bit integers; write a program to compress a text file), that have to be completed, and which might be asked about in the exam. Although many bemoaned the removal of practicals from GCSE science, the result of this shift has been a strengthening of practical, experimental science in schools. Similarly, the independent schools I know that offer IGCSE computing are doing more practical programming, not less.

## Show us what you can do!

An alternative would be to go for something much more open-ended. Instead of 'solve this problem subject to these constraints', what about 'show me what you can do'? This does seem to work for A level computer science (and similar qualifications at 18+, like the EPQ and AP CS Principles in the US): students get to choose a worthwhile, challenging project; have scope for independent, creative work; and get something to talk about at interview. There are some great examples out there, more than a few involving Raspberry Pis. I wonder if this sort of open-ended, student-led project could work for GCSE, too? (HW)

**Miles Berry** is principal lecturer in computing education at the University of Roehampton. He helped draft the English computing curriculum and is a Member of the Raspberry Pi Foundation.



# JOIN THE GLOBAL CODERDOJO COMMUNITY

Free, global, volunteer-led, coding clubs for young people aged 7-17. Code apps, build websites, make games, hack hardware and enjoy exploring technology together.

**START A DOJO**

**VOLUNTEER AT A DOJO**

**SUPPORT THE MOVEMENT**

## RESOURCES & SUPPORTS

There is a host of tools & supports to enable you to empower youths:



**ONLINE TRAINING**



**LEARNING RESOURCES**



**COMMUNITY SUPPORT**



**DIGITAL BADGES**

See how you can join the community at [coderdojo.com](http://coderdojo.com) or get in touch with us at [info@coderdojo.com](mailto:info@coderdojo.com)



[helloworld.cc](http://helloworld.cc)