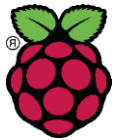
Moving to mainstream …. developing computing for all

Sue Sentance
Raspberry Pi Foundation

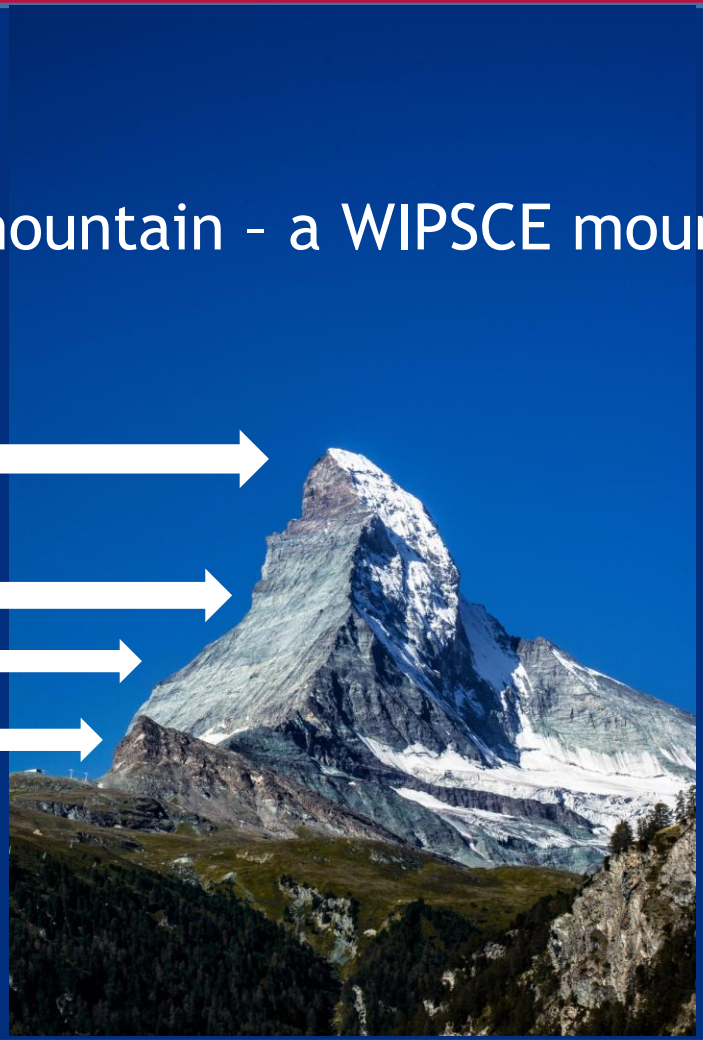
@suesentance
sue@raspberrypi.org



This is our mountain – a WIPSCE mountain?

Computing taught well in every school to every child

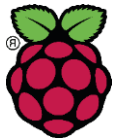
Where are you in your country now? ?



Moving to mainstream – outline of talk

- The Computing context
- The National Centre for Computing Education
- Being research-informed
- Progression
- Supporting teachers' pedagogical content knowledge
- Diversity and inclusion
- A research agenda for a whole country

Teach Computing



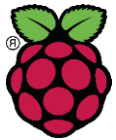
Background

Computing education in England

1970s – 1985
Computer Studies
Elective
14-18 years

1988 – 2013
Information and Communications Technology
Mandatory
5-16 years
Elective CS 16-18

2014 – date
Computing
Mandatory
5-16 years
Elective CS 14-18



Computing for all

Computing for everyone

Figure 1 – Structure of the national curriculum

	Key stage 1	Key stage 2	Key stage 3	Key stage 4
Age	5 – 7	7 – 11	11 – 14	14 – 16
Year groups	1 – 2	3 – 6	7 – 9	10 – 11
Core subjects				
English	✓	✓	✓	✓
Mathematics	✓	✓	✓	✓
Science	✓	✓	✓	✓
Foundation subjects				
Art and design	✓	✓	✓	
Citizenship			✓	✓
Computing	✓	✓	✓	✓
Design and technology	✓	✓	✓	
Languages[4]		✓	✓	
Geography	✓	✓	✓	
History	✓	✓	✓	
Music	✓	✓	✓	
Physical education	✓	✓	✓	✓

Teach Computing

So why every child?

A foundational subject discipline that every child should learn

A subject that will equip our children for the jobs of the future

Teach Computing

Computing for all =

A foundational subject discipline that every child should learn

+

A subject that will equip our children for the jobs of the future

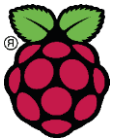
+

An understanding of how to make computing inclusive and accessible

+

Research

Teach Computing



Our young people need …

… to understand the impact of AI on society on a world we haven't met yet

… to have the computational thinking skills needed to solve problems, apply logic and generalise from patterns

… to know how to use computers to model and make predictions about the future

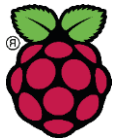
… to understand ethical issues around privacy and security and how technology impacts those

… to be able to create the tools they need for the tasks they have to do

… to be empowered by technology not a slave to it

… to be independent learners who can adapt to technological developments of the future

Teach Computing



Challenges for schools

- **75%** existing teachers of GCSE computer science (elective course offered at age 14-16) do not have academic background in Computer Science
- There are not enough computing teachers: only **53%** secondary schools currently offer GCSE computer science.

After the Reboot (The Royal Society, 2017)

"our evidence shows that computing education across the UK is patchy and fragile. Its future development and sustainability depend on swift and coordinated action by governments, industry, and non-profit organisations".

Recommendations relating to:

1. Computing for all & improving gender balance
2. Support for teachers & increasing supply
3. Improvement of computing education through research

https://royalsociety.org/topics-policy/projects/computing-education/

After the Reboot (contd)

Why a focus on research?

- Pedagogies for computing are less developed than other subjects
- Lack of effective sharing of knowledge between researchers, teachers and teacher trainers.
- Lack of school-level research studies

The National Centre for Computing Education

New funding!

- In December 2017 the Treasury announced £100m funding for computing education in UK (meaning £84m for England)
- In November 2018 the Department for Education announced the **National Centre for Computing Education** to support teachers, with a separate programme on gender balance research
- Teach Computing was launched at http://teachcomputing.org
- The programme runs from Nov 2018 – July 2022

Teach Computing

National Centre for Computing Education

Our vision is for every child in every school in England to have a world-leading computing education.

Raspberry Pi

bcs The Chartered Institute for IT

STEM LEARNING

http://teachcomputing.org

Teach Computing

Professional development opportunites

- Face-to-face CPD being delivered all over England
- School-led model
- Bursaries available for GCSE teachers and priority schools
- Online courses for teachers tailored to the curriculum
- Leading to certification
- Regionally delivered

RASPBERRY PI FOUNDATION

Teaching Programming in Primary Schools

Understand key programming concepts and apply them using Scratch, with this introductory course for primary or K-5 teachers.

4 weeks 2 hrs per week

Join free

RASPBERRY PI FOUNDATION

An Introduction to Computer Networking for Teachers

Build your knowledge and understanding of computer networks as a computer science teacher.

3 weeks 2 hrs per week

Join free

RASPBERRY PI FOUNDATION

Understanding Computer Systems

Understand how the components of a computer system interact with each other on this online course for computing teachers.

3 weeks 2 hrs per week

Join free

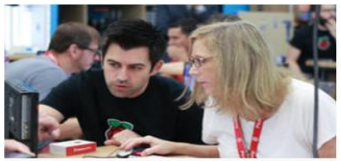
RASPBERRY PI FOUNDATION

Understanding Maths and Logic in Computer Science

Improve your understanding and ability to teach maths and logic in computing while building elements of an escape room.

3 weeks 2 hrs per week

Join free

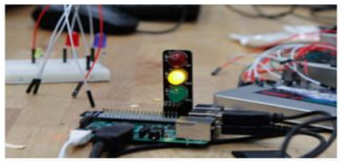


RASPBERRY PI FOUNDATION

Object-oriented Programming in Python: Create Your Own Adventure Game

RASPBERRY PI FOUNDATION

How Computers Work: Demystifying Computation



RASPBERRY PI FOUNDATION

Teaching Physical Computing with Raspberry Pi and Python

RASPBERRY PI FOUNDATION

Build a Makerspace for Young People

Find out how to create and run a

35 online training courses (16 now) https://teachcomputing.org/courses

Teach Computing

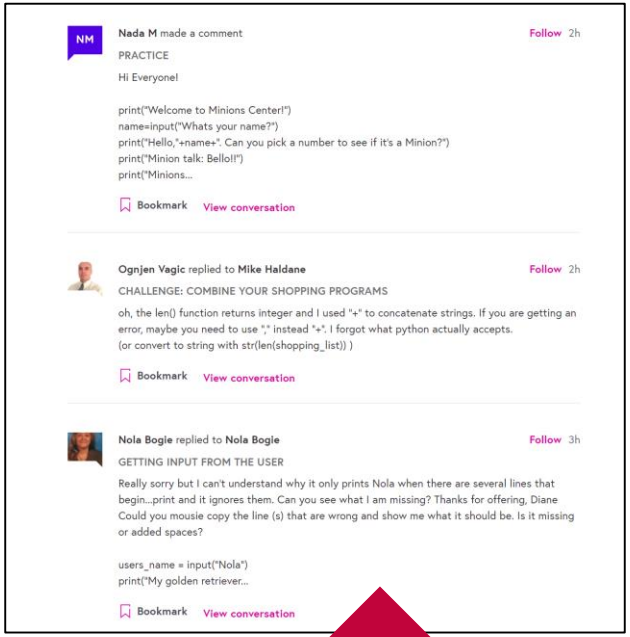
Collaboration and engagement in online courses

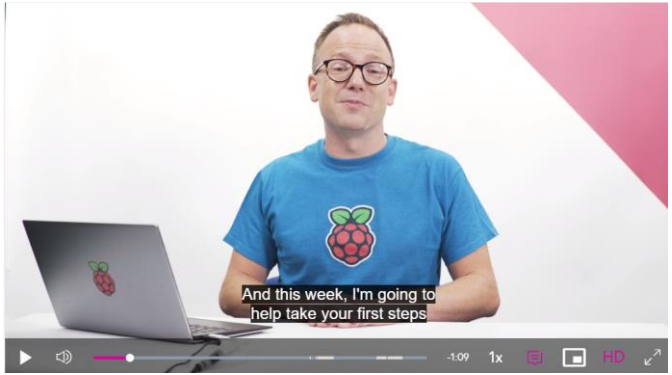
Online course in Tech & Coding

Programming 101: An Introduction to Python for Educators

Explore the basics of Python. Guided by the Raspberry Pi Foundation, you'll learn to code your first program. Supported by Google.

A range of online courses are available





Videos demonstrate each step

Teachers share their progress and ask questions

Teach Computing

Face-to-face training

Algorithms in GCSE computer science CP200

Develop your understanding of algorithms to successfully teach this topic in GCSE computer science.

View dates & locations

Algorithmic thinking | Computing | Mathematics | Key stage 4 | CS Accelerator

GCSE Computer Science – developing outstanding teaching
CP205

This course helps teachers of GCSE to develop their classroom practice and deliver outstanding teaching of GCSE computer science

Computing | Key stage 4

KS4 computing for all CP207

This action-oriented professional development course explores the options available to computing teachers in different school contexts

Ashington 27 November–27 November 2019

Cross curricular | Leadership | Computing | Key stage 4

Networks and cyber-security in GCSE computer science CP202

Develop your knowledge of networks, computer security and guarding against threats to successfully teach your students

View dates & locations

Computer networks | Safety & Security | Computing | Key stage 4 | CS Accelerator

Primary programming and algorithms CP003

Discover engaging and effective ways to help children use computational thinking.

View dates & locations

Cross curricular | Algorithmic thinking | Programming | Computing | Mathematics | Key stage 2

Python programming essentials for GCSE computer science
CP203

Discover the fundamentals of programming and develop your skills to teach GCSE computer science.

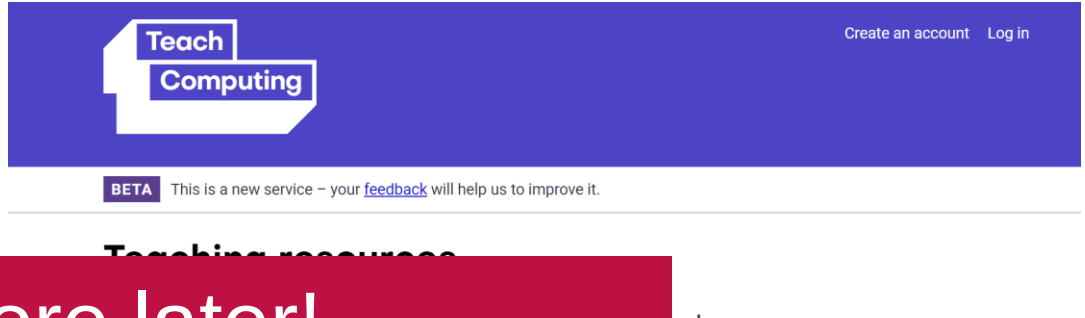
View dates & locations

Regional support

- 30 Computing hubs
- Delivery of CPD and support
- Local “subject matter experts” supporting schools
- Toolkits to help school leaders advance their Computing provision

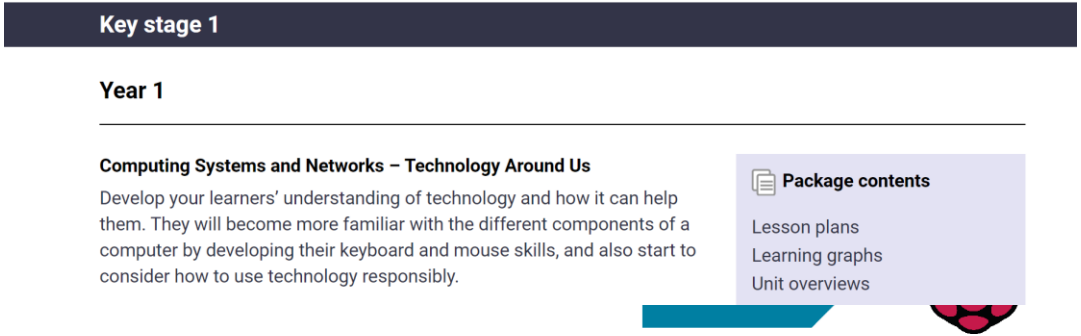
Full curriculum resources for Computing

A comprehensive collection of material to support 500 hours of teaching materials, facilitating the delivery of the computing curriculum age 5-16 by July



More later!

All resource repository content is free, and editable ensuring the resources can be tailored to each individual teacher and school setting.

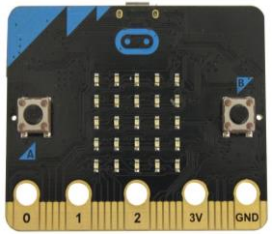


Physical computing kits on loan

Computing hub schools will have class sets of hardware to loan

All resources will be available with training provided

Crumbles



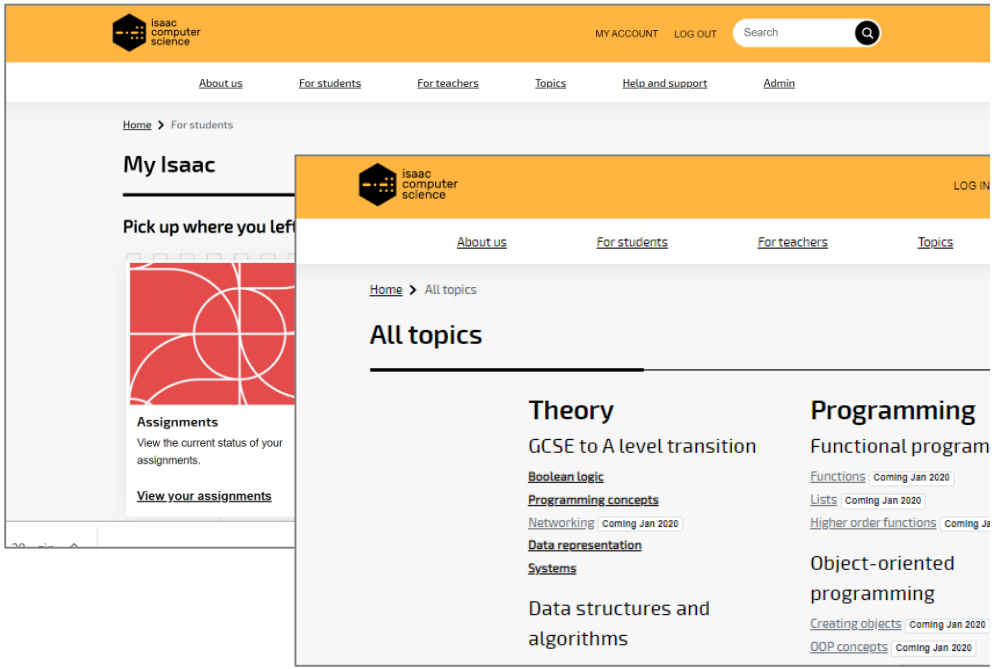
Microbits

Raspberry Pis

Teach Computing

Dedicated support for students 16-18

Isaac Computer Science



- A Level computer science
- Complete specification coverage
- Students learn using content and questions
- Teachers can set up groups and set assignments
- Events for students and teachers

http://isaaccomputerscience.org

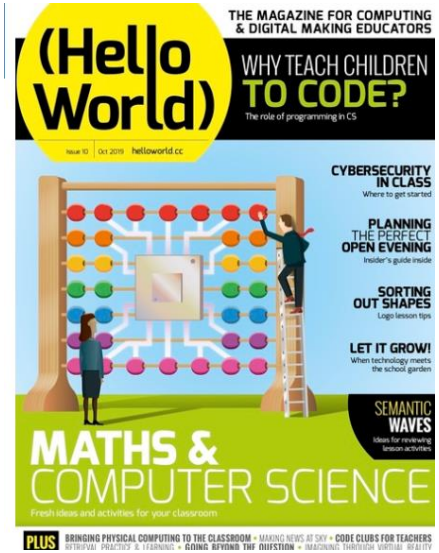
isaac computer science

isaac Physics. You work it out.

Certification

NCCE Certificate in Primary Computing Teaching
NCCE Certificate in Secondary Computing Teaching
NCCE Certificate in GCSE CS Subject Knowledge

Hello World!



Free magazine for Computing teachers
5 issues per year
http://helloworld.cc

Teach Computing

Research-informed

Professional learning opportunities

Draw on research from professional development

- Link to pupil outcomes
- Sustained and collaborative
- Research-informed

Teach Computing

Creating materials - principles

Large development team – including contractors. All resource writers follow key principes

Ten Key Principles

1
2
3
4
5
6
7
8
9
1

Pedagogy - general

(including pro
digital literacy

Techniques

1. Collab
2. Disco
3. Unplu
4. Sema
5. Storyt
6. Mode
7. Choos
8. Cross
9. Explo
10. Guide
11. Makin
12. Targe
13. Self-r
14. Tutori

Pedagogy - specific to programming

Techniques & Approaches - how to realise the principles

1. Pair programming
2. Peer instruction
3. Modelling/ Live coding
4. Design as a software building process
5. Models and frameworks such as Use-Modify-Create, PRIMM, FACT, UDL, ATT, LOA
6. Cognitive apprenticeship
7. Reading/tracing code
8. Worked examples & sub-goal labelling
9. Targeted tasks - eg debugging, sabotage, fill in the gaps, annotation, Parson's problems
10. Physical computing

Progression

Computing Taxonomy	5-7	7-11	11-14	14-16 Core	14-16 Elective	16-18 Elective
Algorithms	✓	✓	✓	✓	✓	✓
Creating Media	✓	✓	✓	✓	(✓)	(✓)
Computer Networks	✓	✓	✓		✓	✓
Computer Systems	✓	✓	✓		✓	✓
Data & Information	✓	✓	✓		✓	✓
Design & Development	✓	✓	✓	✓	✓	✓
Effective use of tools	✓	✓	✓	✓	✓	
Impact of technology	✓	✓	✓	✓	✓	✓
Programming	✓	✓	✓	(✓)	✓	✓
Safety & Security	✓	✓	✓	✓	✓	(✓)

Progression in Computing – when do you teach what?

Goal

- Develop a learning journey from 5-16, sequencing all topics

Process

- Reviewed curriculum, specs, products, text books, existing resources
- Reviewed research, where available
- Created a high level taxonomy of concepts and skills
- Use taxonomy strands to begin creating concept maps and learning trajectories

Teach Computing

Mapping and trajectories

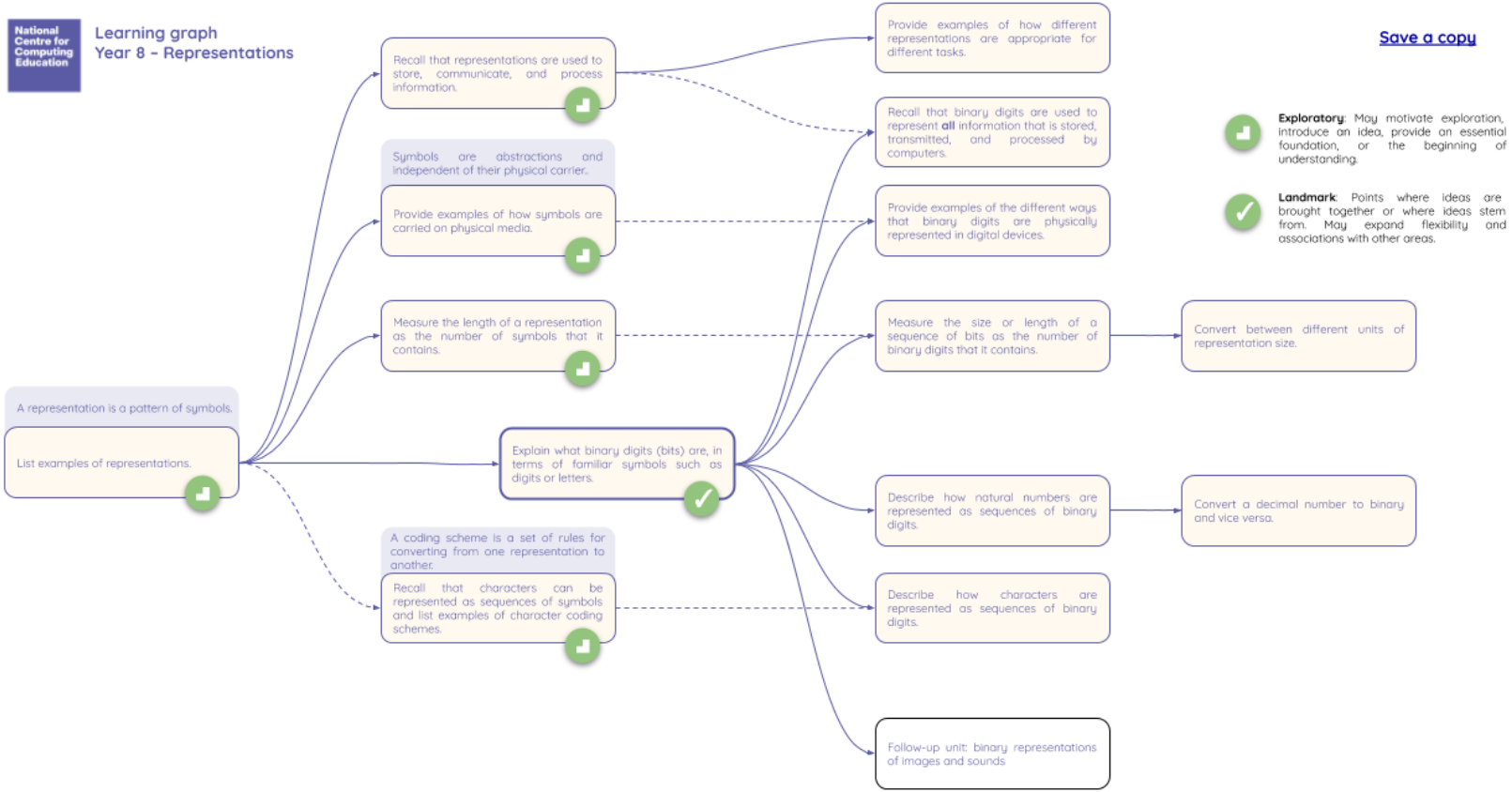
- Concept maps are graphical tools for organising and representing knowledge. They include concepts, usually enclosed in circles or boxes of some type, and relationships between concepts indicated by a connecting line linking two concepts.
- Children follow natural developmental progressions in learning and development which we can call learning trajectories, made up of a goal, a developmental path, and a set of instructional activities or tasks matched to each of the levels of thinking in that path that help children develop higher levels of thinking.

Andreas Mühling (2016) Aggregating concept map data to investigate the knowledge of beginning CS students, Computer Science Education, 26:2-3, 176-191

Rich, K. M., Strickland, C., Binkowski, T. A., Moran, C., & Franklin, D. (2017). K-8 learning trajectories derived from research literature: Sequence, repetition, conditionals. In *Proceedings of the 2017 ACM conference on international computing education research* (pp. 182-190). ACM.

National Centre for Computing Education

Learning graph
Year 8 – Representations



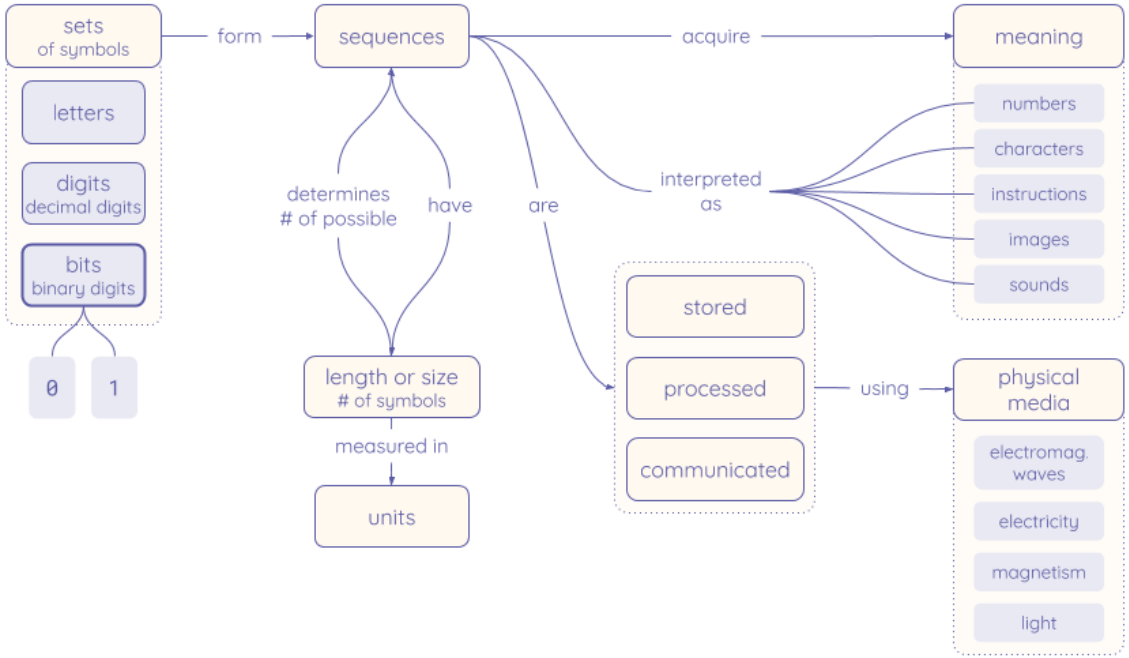
Save a copy

Exploratory: May motivate exploration, introduce an idea, provide an essential foundation, or the beginning of understanding.

Landmark: Points where ideas are brought together or where ideas stem from. May expand flexibility and associations with other areas.

National Centre for Computing Education

Key Stage 3 – Year 8
Representations I
Binary representations concept map

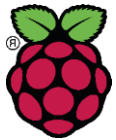


Teach Computing

Key Stage 3 (11 – 14 years old) Units

Year 7	Collaborating Online Respectfully Lab Introduction Responsible Use Cyberbullying Security	Media Unit: Gain support for a cause Word Processing Licensing Plagiarism Source Credibility	Networks I Networks Internet Web Cybersecurity Safety	Spreadsheets Basic Formulas Basic Functions Formatting (incl. conditional)	Programming 1 Scratch PRIMM Form. Assessment Id. Misconceptions	Programming 2 Scratch Lists Subroutines Deeper dive
Year 8	Media Unit E.g. vector graphics (inkscape)	Computing systems From programs down to bits.	Networks 2 Web HTML, CSS Search	Representations I Representations Binary Numbers Text Units	Programming 3 AppLab Pair Programming Product Design GUI Design	Programming 4 Intro to Python “Guess the number” game Algorithms Searching
Year 9	Media Unit E.g. animations (blender) E.g. video editing	Impact of Technology -	Databases and Cyber Security Big Data SQL queries Security GDPR Social Engineering	Representations 2 Images Sounds Manipulation	Programming 5 Physical Computing Python + micro:bit Python + sensehat	Programming 6 Projects Physical Computing Algorithms Data Apps Web

Teach Computing



14-16 years old - Key Stage 4 Units

theory oriented

Data Representation
12-16 hrs

Binary, Hex
Conversions & Ops.
Text
Images & Sound
Data Capacity
Compression

Computer Systems
12-15 hrs

Components
Architecture
Storage
Software
Boolean logic

Networks
8 hrs

Components
Classifications
Protocols
Layers

Security
6-8 hrs

Vulnerabilities
Forms of Attack
Techniques for:
Identification
Protection

Impacts
6-8 hrs

Ethical
Legal
Environmental
(inc. privacy and cyber security)

algorithms programming

Algorithms
12-14 hrs

Tracing & Exec.
Representation
Searching
Sorting
Efficiency
Comp. Thinking

Programming
30-36 hrs

Tracing & Exec.
Prog. constructs
Data types, structs
Modularity
Quality
Translators

Project (NEA)
30 hrs

Preparation
Implementation

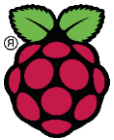
Teach Computing

Computer systems and networks Strand - Primary

Computer Systems and networks

Year 1	Year 2	Year 3	Year 4	Year 5	Year 6
Technology around you 6 hrs Learners become familiar with the different components of a computer by developing their keyboard and mouse skills	Information technology around you 6 hrs Learners will be looking at information technology at school and beyond, in settings such as in shops, hospitals and libraries. It will investigate how information technology improves our world	Connecting computers 6 hrs Introduction to computer networks, including devices that make up a networks infrastructure, such as wireless access points and switches	Data storage and the internet 6 hrs	Online collaboration 6 hrs	Searching and online communication 6 hrs

Teach Computing



Research and evaluation

- Iterative development of learning graphs with teachers
- How are learning graphs used?
- What does progression look like in the Computing classroom?

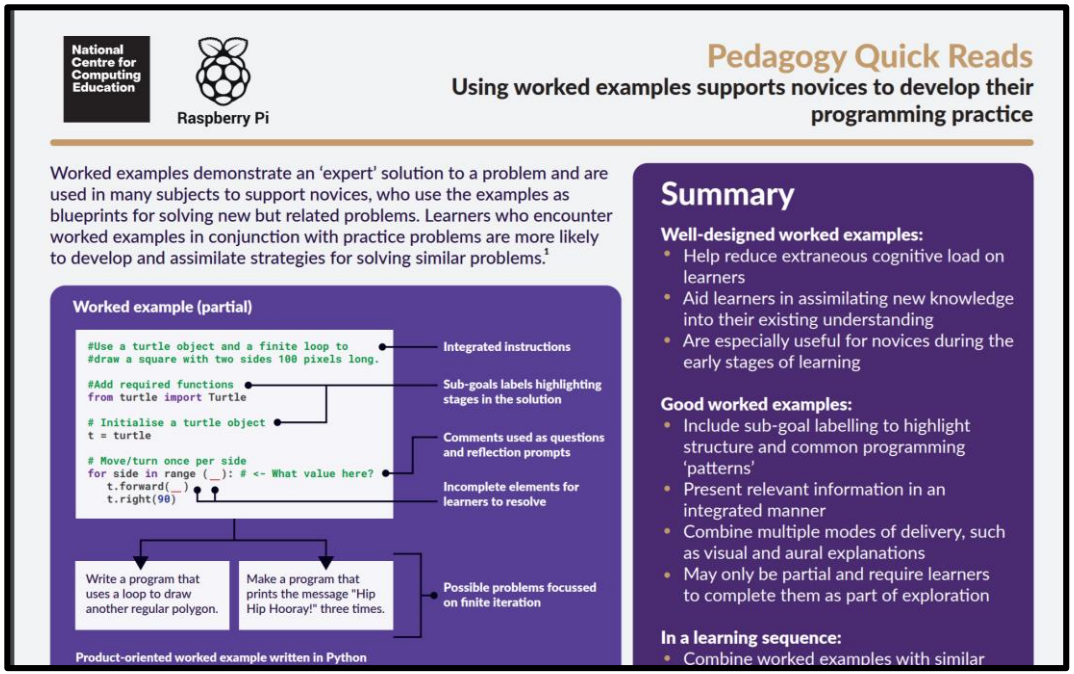
Supporting teachers’ PCK

Where we were … in 2013

Exclusive focus on subject knowledge

Teach Computing

Where we are now … in 2019



National Centre for Computing Education

Raspberry Pi

Pedagogy Quick Reads

Using worked examples supports novices to develop their programming practice

Worked examples demonstrate an 'expert' solution to a problem and are used in many subjects to support novices, who use the examples as blueprints for solving new but related problems. Learners who encounter worked examples in conjunction with practice problems are more likely to develop and assimilate strategies for solving similar problems.[1]

Worked example (partial)

```
#Use a turtle object and a finite loop to
#draw a square with two sides 100 pixels long.

#Add required functions
from turtle import Turtle

# Initialise a turtle object
t = turtle

# Move/turn once per side
for side in range (__): # <- What value here?
    t.forward(__)
    t.right(90)
```

Integrated instructions

Sub-goals labels highlighting stages in the solution

Comments used as questions and reflection prompts

Incomplete elements for learners to resolve

Write a program that uses a loop to draw another regular polygon.

Make a program that prints the message "Hip Hip Hooray!" three times.

Possible problems focussed on finite iteration

Product-oriented worked example written in Python

Summary

Well-designed worked examples:

- Help reduce extraneous cognitive load on learners
- Aid learners in assimilating new knowledge into their existing understanding
- Are especially useful for novices during the early stages of learning

Good worked examples:

- Include sub-goal labelling to highlight structure and common programming 'patterns'
- Present relevant information in an integrated manner
- Combine multiple modes of delivery, such as visual and aural explanations
- May only be partial and require learners to complete them as part of exploration

In a learning sequence:

- Combine worked examples with similar

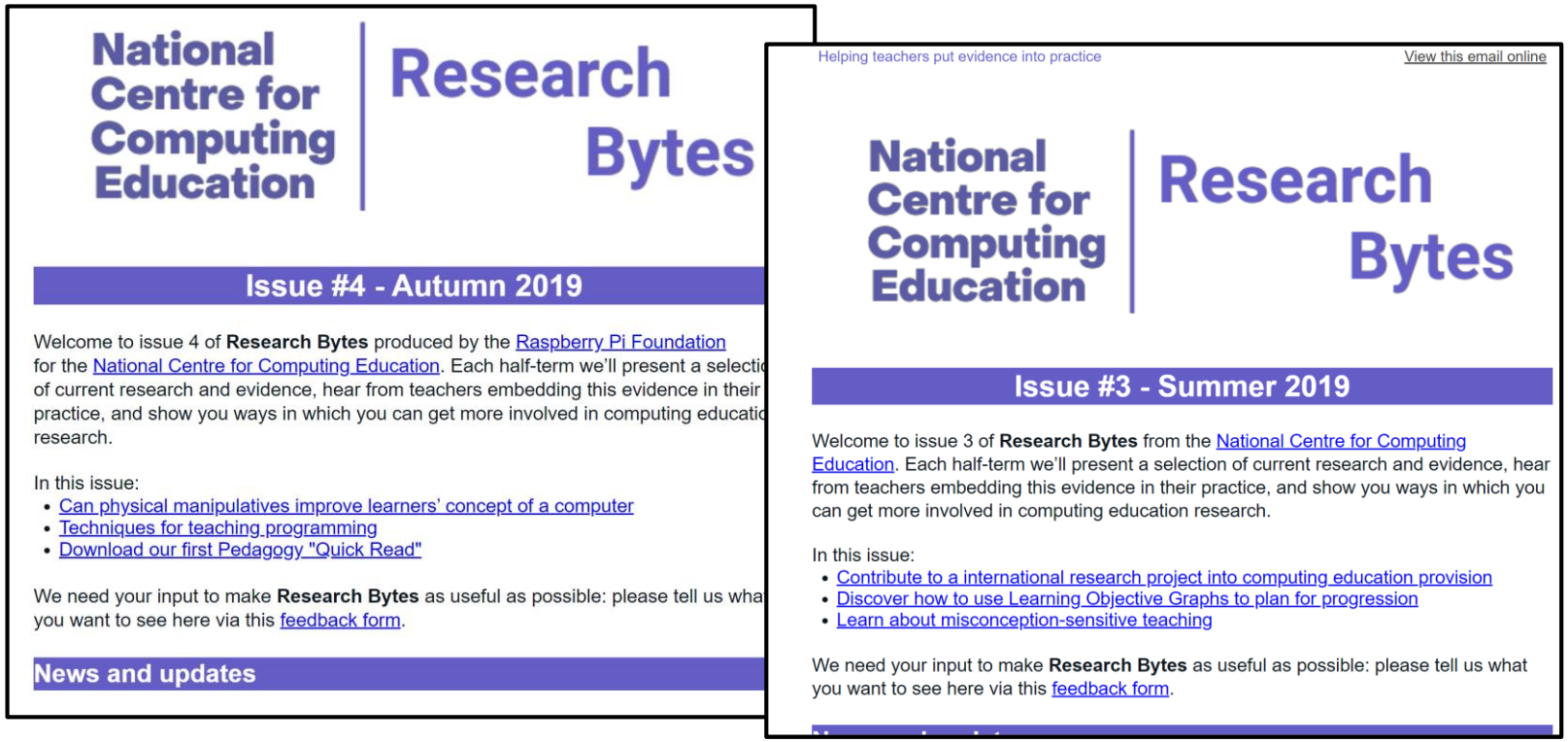
Short summaries of research around topics such as cognitive load, worked examples, pair programming

Examples of application to practice

Built into resources

Teach Computing

Helping teachers put evidence into practice



National Centre for Computing Education | Research Bytes

Issue #4 - Autumn 2019

Welcome to issue 4 of **Research Bytes** produced by the Raspberry Pi Foundation for the National Centre for Computing Education. Each half-term we'll present a selecti of current research and evidence, hear from teachers embedding this evidence in their practice, and show you ways in which you can get more involved in computing educati research.

In this issue:

- Can physical manipulatives improve learners' concept of a computer
- Techniques for teaching programming
- Download our first Pedagogy "Quick Read"

We need your input to make **Research Bytes** as useful as possible: please tell us wha you want to see here via this feedback form.

News and updates

Helping teachers put evidence into practice — View this email online

National Centre for Computing Education | Research Bytes

Issue #3 - Summer 2019

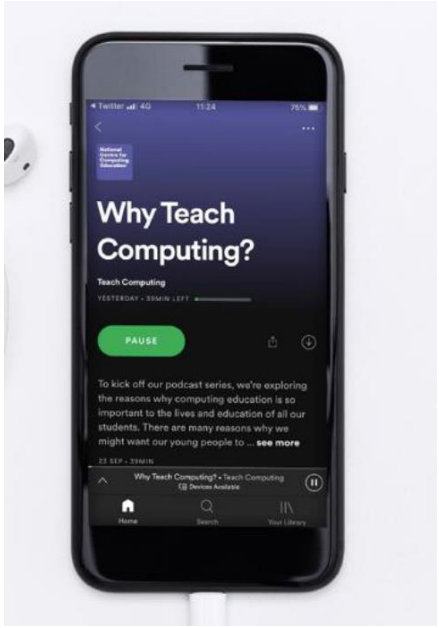
Welcome to issue 3 of **Research Bytes** from the National Centre for Computing Education. Each half-term we'll present a selection of current research and evidence, hear from teachers embedding this evidence in their practice, and show you ways in which you can get more involved in computing education research.

In this issue:

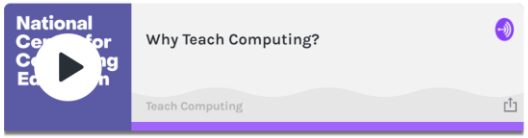
- Contribute to a international research project into computing education provision
- Discover how to use Learning Objective Graphs to plan for progression
- Learn about misconception-sensitive teaching

We need your input to make **Research Bytes** as useful as possible: please tell us what you want to see here via this feedback form.

Teach Computing podcast



Episode #1: Why teach computing?



To kick off our podcast series, we're exploring the reasons why computing education is so important to the lives and education of **all** our students. There are many reasons why we might want our young people to be confident and literate, not only in how to use technology, but also in how it works and is created.

- First and foremost, computing should be tremendous fun and is one of the most creative disciplines; it enables our learners to create, invent, explore and simulate the world around them.
- Exposing students to computing skills and concepts is incredibly empowering, giving them new ways to solve problems, represent their world, and express their ideas.

Research and evaluation

- Which pedagogical approaches impact learning in the K-12 classroom?
- How does professional learning around Computing pedagogy impact student learning?

Diversity and inclusion

Gender Balance in Computing Project

A new programme of research to investigate which interventions may be effective in school to both engage female students and to increase numbers selecting computer science at GCSE and A level.

Trials will run from 2019–2022 in key stages 1–4, and over 15,000 students and 550 schools will be involved. It will be the largest national research effort to tackle this issue to date.

Five interventions + an innovation strand 2019-2022

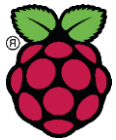
Intervention 1: Teaching approach
Intervention 2: Timetabling
Intervention 3: Belonging
Intervention 4: Relevance
Intervention 5: Non-formal learning
Intervention 6: Innovation strand

All interventions are randomised control trials in schools except timetabling

Inclusion and support

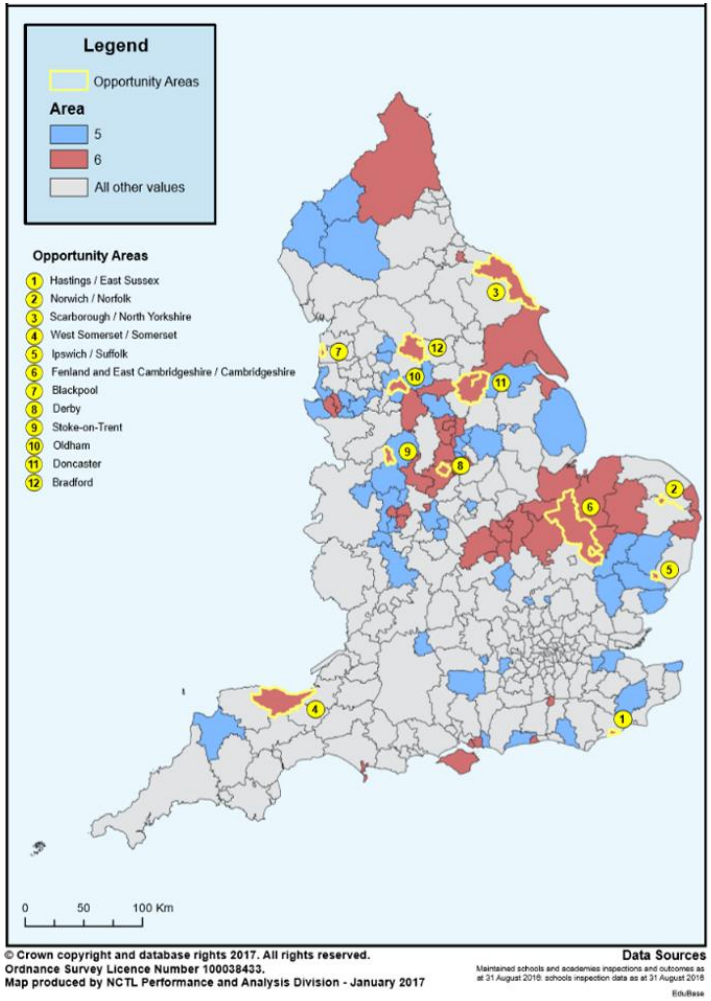
Online course for teachers out soon to support teachers to:

- Recognise the barriers faced by a range of students with special educational needs and disabilities in the computing classroom
- Evaluate and use a range of general technology to support students with SEND. Identify effective strategies for supporting individuals in computing.
- Identify effective strategies for supporting individuals in computing.
- Recognise the range of specific computing resources available to support students with SEND.
- Understand the benefits for all students in developing inclusive practice.



Socio-economic inclusion

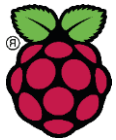
- A significant focus of the National Centre is on reaching schools in “Opportunity areas”
- Opportunity areas are defined as having low social mobility and capacity to improve
- In these areas there are less opportunities for young people to achieve
- The NCCE is focusing on reaching these areas



Research and evaluation

- Which interventions have a measurable impact on interest, engagement and motivation in computing?
- Which approaches are effective in making computing accessible to students with special educational needs?
- To what extent does incentivising and external support impact take up of computing in areas of low social mobility?

Teach Computing



Challenges

What are the challenges?

Reach and Scale

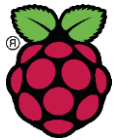
- Early adopters & enthusiasts have all been reached
- How to reach schools where Computing is not a priority

Getting schools started from a low base

- Teachers shortages and retention
- Training takes time

Research

- Not currently being explicitly funded



Research agenda

Research agenda – Part 1

As already discussed:

- How can we frame the learning journey in computing through a progression framework?
- Which pedagogical approaches are most appropriate for different topics, contexts, educational phases?
- What interventions have an impact on gender balance in computing

Research agenda – Part 2

Scale

- How can countries develop professional development journeys that are appropriate, at scale?
- What online support for teachers has impact in the classroom?

Hard to reach

- What interventions support schools and teachers who are motivationally challenged by computing?

Computing for all

- How do we ensure the subject is accessible to all our children (All 8 million of them!)

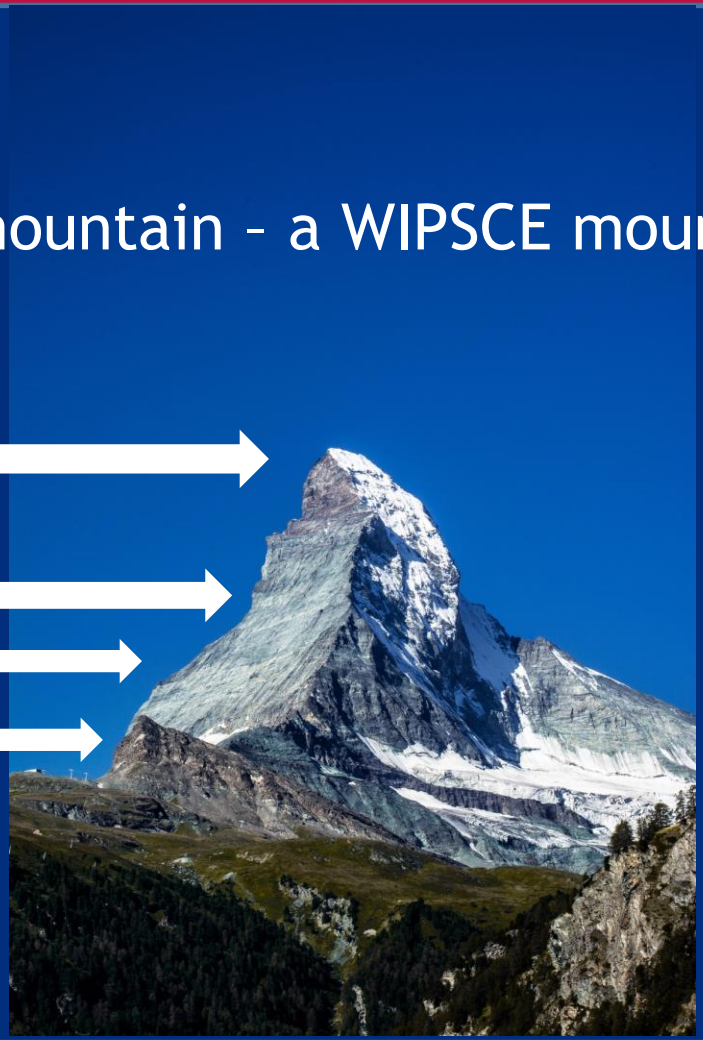
Research agenda – Part 3

What do you think…… ?

This is our mountain – a WIPSCE mountain?

Computing taught well in every school to every child

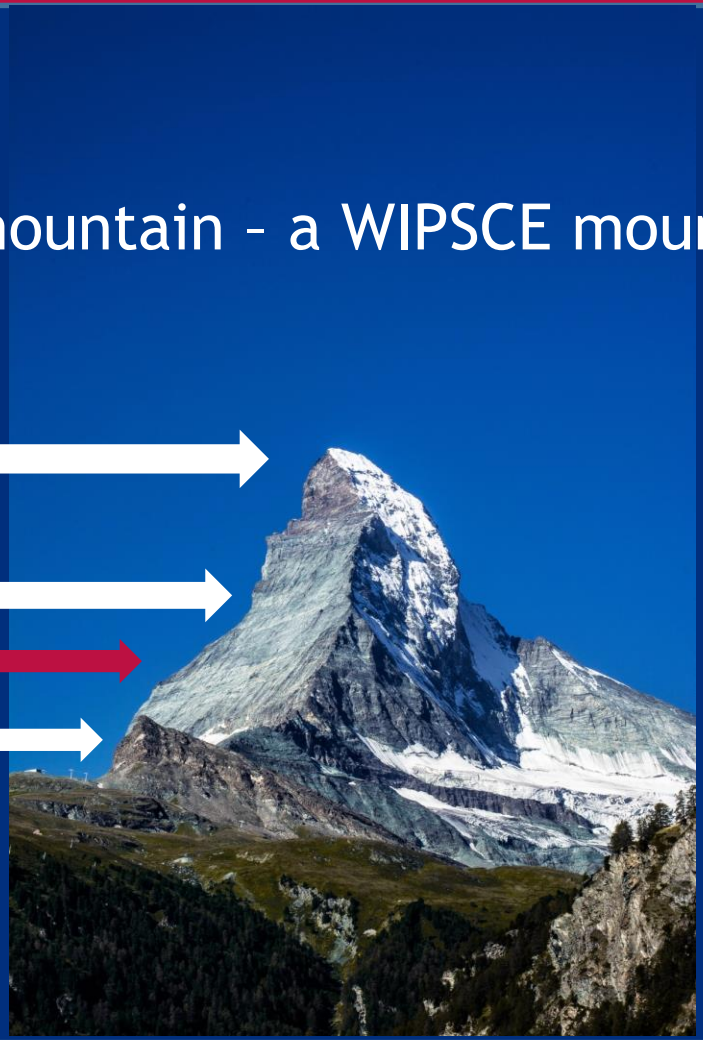
Where are you in your country now? ?



This is our mountain – a WIPSCE mountain?

Computing taught well in every school to every child

Where are you in your country now? ?



Find out more – get involved!

- Read all out news at http://teachcomputing.org
- Try our A Level platform at http://isaaccomputerscience.org
- Explore our beta teaching resources http://teachcomputing.org/resources
- Write for Hello World! http://helloworld.cc
- Explore Raspberry Pi Projects http://projects.raspberrypi.org
- Sign up for online courses http://rpf.io/courses
- Enter our international Coolest Projects competition https://coolestprojects.org/international/

Teach Computing

Questions?

Sue Sentance
Raspberry Pi Foundation

@suesentance
sue@raspberrypi.org