Just2easy
Create · Share · Learn

Whitepaper

Computational Thinking and Critical Thinking Skills:

Understanding the Importance of Coding & how to succeed in the Classroom

The background to coding

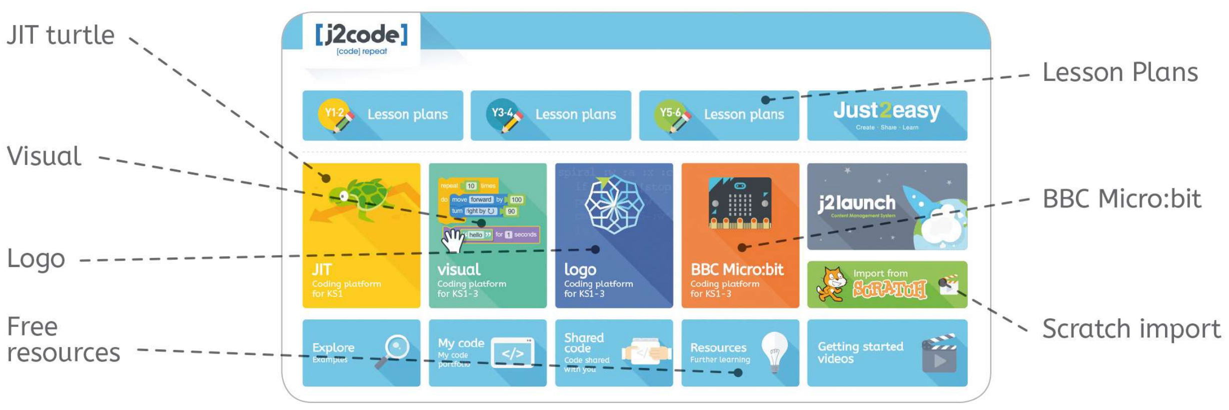
Britain has delivered greatness in the history of computing; from the works of Alan Turing all modern computers as we know them were borne, to the creation of the World Wide Web by Sir Tim Berners- Lee.

This paper shows how teachers can deliver engaging lessons in coding –by utilising easy to use, tried and trusted tools and inspire young learners to learn new skills that will lead to higher employability – and gives practical examples of lessons that can be delivered and assessed at any time.

Why is it so important that children should learn to code? We are already live in a world dominated by software. Our favourite TV shows are downloaded via the internet, we chat with our friends over software managed networks, we shop using the internet and it is expected by 2020 that over 6 Billion people will own and use a smart phone. Cars park and drive themselves, our entire medical history and details of medical procedures are streamed live to the professionals who care for us.

Software has been embedded as a critical layer in every person’s existence. It has a global language and in the near future, not knowing the language of computers will be as challenging as being illiterate or innumerate are today.

Every child has the vital need to learn to code, not because they will necessarily become a computer programmer, but because every child needs to develop the skills to solve problems in the way that software engineers do with computational thinking. This combines mathematics, logic and algorithms, and teaches you a new way to think about the world.



The Theory

Computational thinking enables you to tackle large problems by breaking them down into a sequence of smaller, more manageable problems. It allows you to tackle complex problems in efficient ways that operate at huge scale. The applications of this approach stretch far beyond writing software code. Many diverse fields of work employ a computational thinking approach to problems including mechanical engineering, fluid mechanics, physics, biology, archaeology and music. In business we are beginning to understand that markets often follow rules that can be discerned using computational analysis.

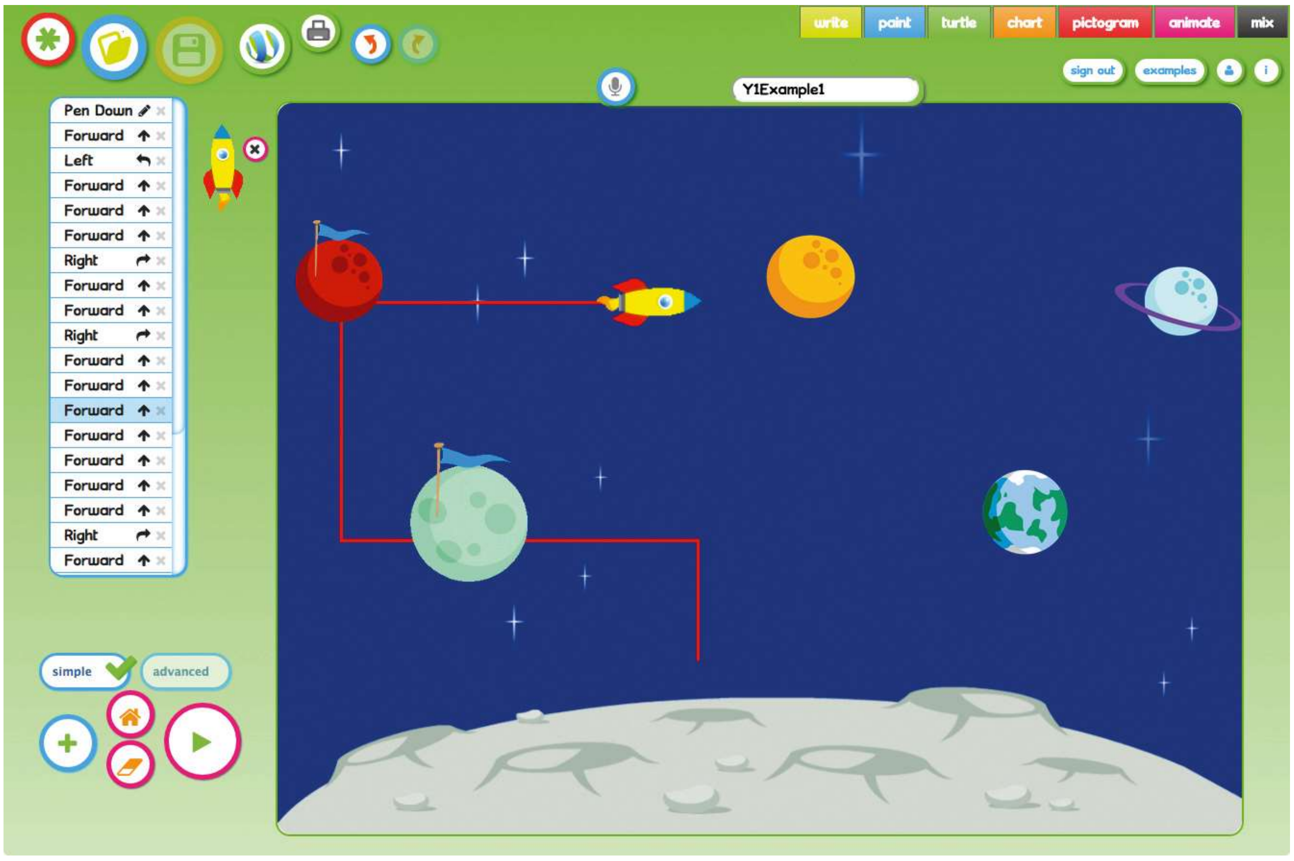
Every person can benefit from acquiring the skills of Computational thinking. Even if your students become everything from sports psychologists to commercial lawyers, they will benefit from knowing how to think this way. It will help you understand and master technology of all sorts and solve problems in almost any discipline.

Computational thinking is seen as a prerequisite for problem solving and follows hand in hand with another prerequisite skill of critical thinking and the development of higher order thinking applied when resolving a problematic situation with skills considered by most authors to include: Analysis and synthesis, producing judgements, decision making and generalisations.

Research shows that students immersed in a technology rich environment develop stronger critical thinking skills, and that to develop higher order thinking skills, schools should integrate technology across all areas of learning. The studies also show that these skills can be acquired by using appropriate software yet the age or function of the hardware used has no positive or negative effect on learner outcome.
Further demonstrated commitment to the development of these skills can be seen by the adoption of Computer Science for all in the United States, where students from K-12 are taught computer science and these critical skills to ensure their stake as active citizens in an ever increasingly technology driven world.

The Practice

Using Turtle for Teaching Computational Thinking and Introducing Algorithms



```
Pen Down
Forward
Left
Forward
Forward
Forward
Right
Forward
Forward
Right
Forward
Forward
Forward
Forward
Forward
Forward
Forward
Right
Forward
```

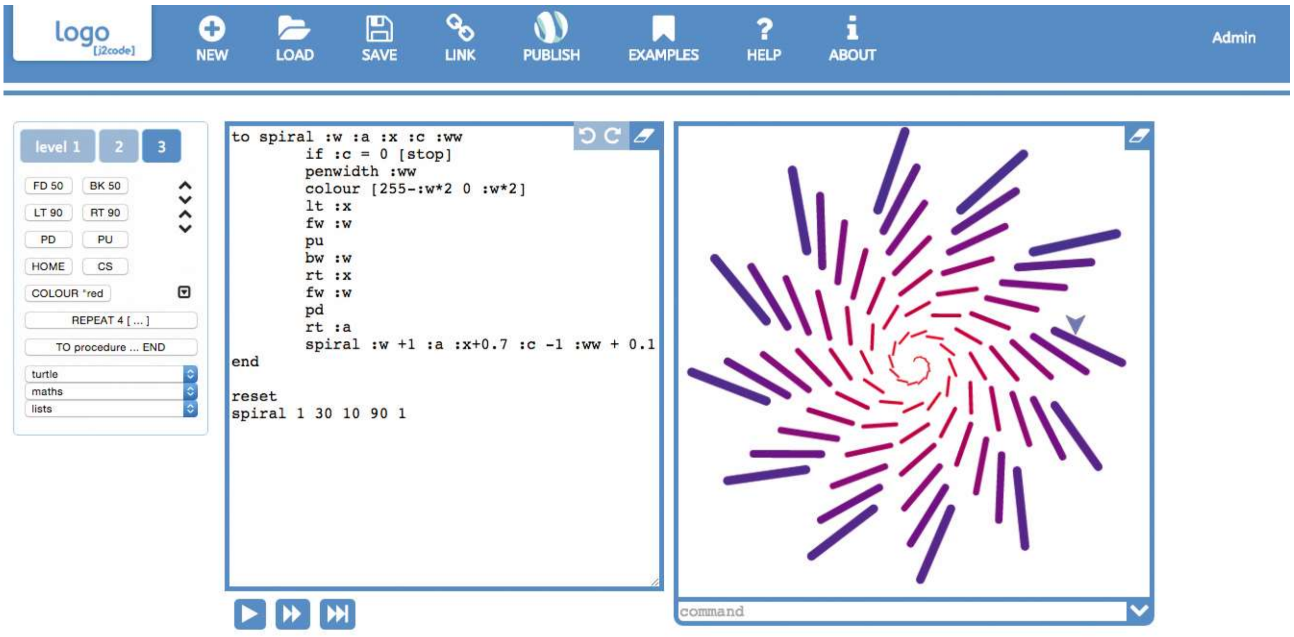
J2code is a set of tools and resources designed to help teachers fulfil the requirements of the new computing curriculum. The emphasis is on children aged 5 to 13. J2Code runs on any modern connected device, including tablets, phones and traditional computers. It is based on three separate coding engines, each with their own advantages. Pupils are encouraged to write their own programs using the coding engines and teachers may want to start this process by following one or more of the suggested lesson plans.

The first coding engine, based on JIT (Just2easy Infant Tools) turtle, is aimed at the youngest learners. It is a bit like a software version of a floor turtle (roamer). It has a bright, colourful user interface. A sprite (animated graphic) has buttons which allow the sprite to be moved and turned. A pen path may be turned on or off to show the path that the sprite has travelled and can run in simple or advanced mode. In simple mode the sprite performs the action immediately, whereas in advanced mode the actions are built up and executed when the user presses play. Commands may be moved, deleted or added to and this introduces the concept of programming.

Using Block Coding to Teach Programming Language and Animation

Visual, is a block based programming language. Blocks (of code) are dragged onto the programming area, where they may be linked together to form a whole program. There are three different levels within Visual which allows a teacher to start with very simple blocks and then gradually introduce more and more. Multiple sprites can have their own program blocks and can interact with each other to form sophisticated programs. Superficially this looks quite like Scratch, which is in use in many schools. However, the fact that it runs on any modern device and that it has different levels to allow differentiation gives it some clear advantages.

Using Logo to Teach problem solving and variables



```
to spiral :w :a :x :c :ww
        if :c = 0 [stop]
        penwidth :ww
        colour [255-:w*2 0 :w*2]
        lt :x
        fw :w
        pu
        bw :w
        rt :x
        fw :w
        pd
        rt :a
        spiral :w +1 :a :x+0.7 :c -1 :ww + 0.1
end

reset
spiral 1 30 10 90 1
```

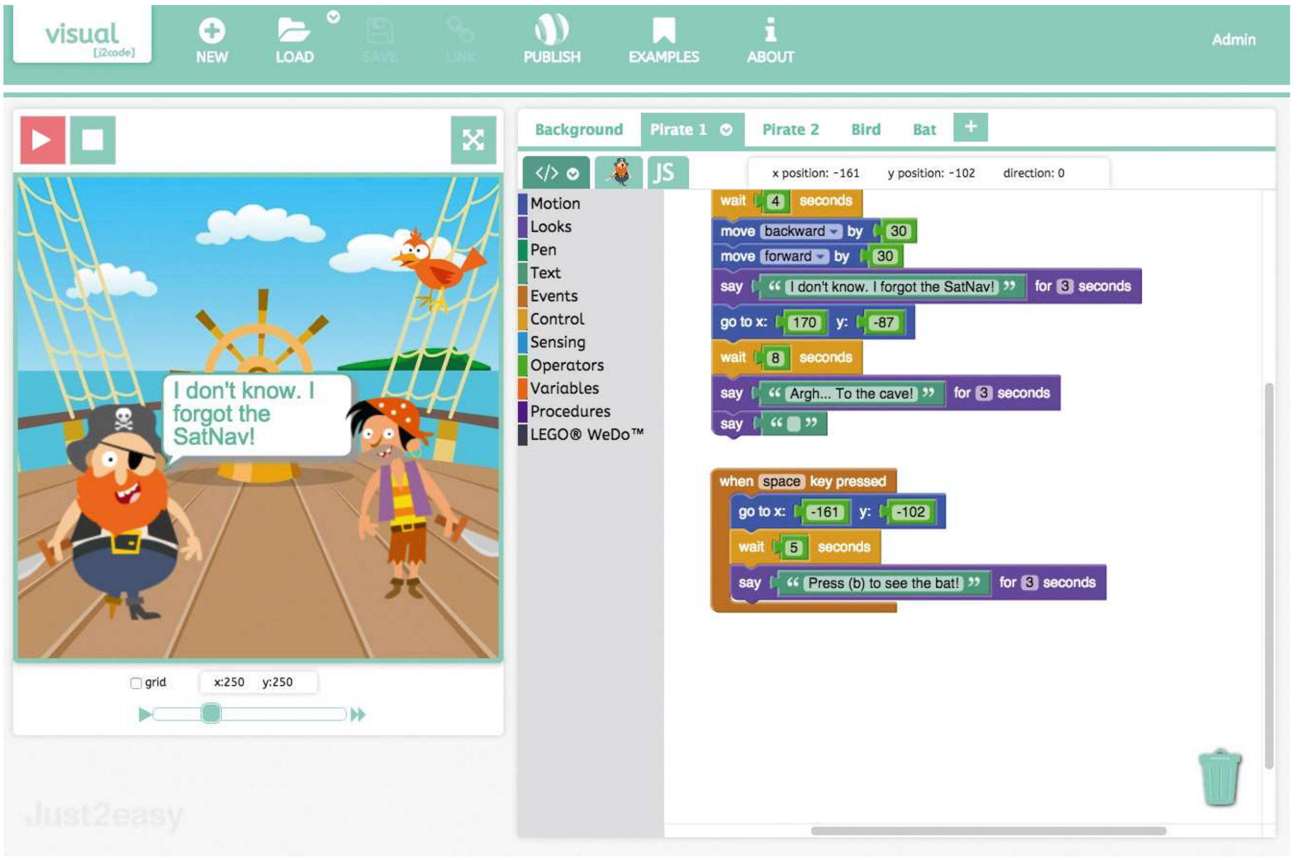
Logo is a completely online version of this classic, text based, educationally focused programming language. There are three levels starting at the most basic and moving up to level three that includes procedures, lists, maths functions and all of the other components of Logo.

Developing higher order thinking with J2e

At the centre of every area of Just2easy lies the ethos of Blooms Taxonomy and provides the opportunity for students to practice and establish their thinking skills, while supporting the learning process.

With J2code, just as in J2write for literacy, students apply their knowledge of coding to solve problems in the form of games or challenges, developing a program that can be broken down in to it's component parts to identify the relational aspects of the code, then provides an environment to derive a set of abstract relations or production of a plan that synthesises all the operations carried out and derive real world meaning from them.

About J2code



```
wait 4 seconds
move backward by 30
move forward by 30
say " I don't know. I forgot the SatNav! " for 3 seconds
go to x: 170 y: -87
wait 8 seconds
say " Argh... To the cave! " for 3 seconds
say "  "

when space key pressed
    go to x: -161 y: -102
    wait 5 seconds
    say " Press (b) to see the bat! " for 3 seconds
```

J2code is an easily accessible platform that enables pupils to quickly move from simple coding to highly sophisticated and exciting programming.

Most programs used for teaching computing in schools are either open ended or prescriptive. Both have their advantages and disadvantages. Open ended programs, like Scratch, allow the user to create sophisticated programs, but may well be daunting for a teacher or pupil to know where to start. Conversely, more prescribed programs may help the teacher get started but often limit what can be achieved with little regard for problem solving and limited higher order thinking opportunities.

With J2Code we combine the best of both situations. By using three coding engines we can introduce the teachers and pupils to the basics of programming, and then gradually take them on further in their journey. The two more sophisticated coding engines are based on industry standards and open source. This ensures that the concepts that are learnt translate directly into real world programming. J2Visual is based on Blockly, a derivative of Scratch. Logo is text based and is a well-known and used programming language within the education community. Each of the coding engines has a number of levels of complexity allowing for both progression and differentiation.

For learners of all ages – anytime, anywhere

Because there are three coding engines, each with differentiated levels of coding complexity, j2code is accessible to all learners including those with special educational needs and the gifted and talented. Lesson plans with video tutorials and examples of outcomes have been provided throughout which can be easily modified for differing scenarios or curriculum requirements. The use of different levels in each of the coding engines allows the teacher to adapt what is being taught to the target audience. For example, using JIT in simple mode for the youngest or least able pupils and then gradually moving to advanced mode as the users become more experienced. Visual and Logo have three separate levels designed to allow progression. However, these levels also enable a teacher to ensure that the lesson is inclusive even in a class with mixed abilities.

No software needs to be installed meaning that children can continue their programming at home just as easily as at school.

Effective Learning Outcomes

J2Code had been specifically designed to engage children by using fun and appealing graphics with seasonal themes and allows differentiation through the structured levels in each coding engine. As a result, effective learning is enabled. With the additional features of J2review where assessment of student work, progress and feedback statements take place, schools see improvement in home-school links and a greater level of support from parents in the learning activities within coding.

What does the future hold?

Given the ever increasing rate of change in the world of technology and computing, predicting the future of coding is an almost impossible art. There are some mid term certainties that can be applied and that is new protocols, new standards, new display methods and above all else, the pressing requirement for Databases to manage the ever increasing level of information being created, stored and shared. J2e firmly commit to ensuring delivery of tools to teach to curriculum requirements and included within it are powerful database tools such as J2data. It is a certainty that the needs of teachers to be able to deliver meaningful, rewarding and engaging lessons to meet the computing and coding elements of the curriculum are ever evolving.

About Just2easy

At Just2easy we are passionate about bringing the best technology into education in a way that is easy to use yet has powerful learning outcomes. We have been creating exciting and engaging educational software since 1994 and are responsible for some of the UK's most popular educational software, winning many BETT awards, including j2webby, J2e5, and j2code. We produce flexible software tools that allow children to be truly creative whilst they learn.

Our software is designed to enable teachers to focus on teaching and learning. All Just2easy products make anytime anywhere learning a reality. FREE home access is given as part of a school licence, giving teachers the ability to set homework for children online. We are a of friendly and knowledgeable educationalists dedicated to providing the best software for teaching and learning.

References:

B. Jack Copeland, Diane Proudfoot . “Alan Turing Father of the Modern Computer” http://www.rutherfordjournal.org/article040101.html

Crewton, Ramone. “Computation vs Critical Thinking” http://www.selfgrowth.com/articles/math-computation-versus-critical-thinking

Buckley, Sheryl. “The Role of Computational Thinking and Critical Thinking in Problem Solving in a Learning Environment” School of Computing UNISA Guateng, South Africa

Crow, Dan. “Why every child should learn to code” The Guardian , Feb 7, 2014 https://www.theguardian.com/technology/2014/feb/07/year-of-code-dan-crow-songkick

Copeland, B. Jack, "The Modern History of Computing", *The Stanford Encyclopedia of Philosophy* (Fall 2008 Edition), Edward N. Zalta (ed.), URL = <https://plato.stanford.edu/archives/fall2008/entries/computing-history/>.

McMahon, G. (2009). Critical Thinking and ICT Integration in a Western Australian Secondary School. Educational Technology & Society, 12 (4), 269–281.