Quacking Javascript - Webmaking using Javascript to promote student engagement and constructivist learning approaches at KS3

by Michael Chesterman

MMU student 13159960

Table of Contents

1 Introduction

This study has a focus of the creation of a creative and engaging curriculum to teach text-based coding to KS3 students.

Recent government-sponsored reports have placed a focus on computational thinking and have created a set of capabilities that should be taught at Key Stage 3 (KS3) (Computing at School, 2010; Computing at School Working Group, 2012; The Royal Society, 2012). These include using a text based language to write algorithms and using techniques including selection, iteration and data types (Department for Education, 2015).

I wanted to try to teach these core elements in line with an exploratory, constructivist approach. I chose to adapt the Webmaking concept proposed by Mozilla to use the Javascript text coding language in the framework of webpage creation. The scope of the use of the materials created for this course is aimed beyond the walls of the classroom for use in computer clubs, adult education and community-based computer education.

The second part of this essay shows the application of theory in practice. I evaluate my planning, teaching and assessment for learning. In addition particular activities and approaches are selected for analysis, specifically their impact on student motivation, engagement and higher constructivist learning outcomes. Areas for further subject investigation and professional development are dealt with in the final section.

2 PART ONE: Justification of focus for curriculum development

I can remember creating my first web pages in 1995. A volunteer at Hulme Community Computing centre had lent me a 20 page print out of how to create web pages. I used a text editor on a laptop running MSDOS to type HTML code and saved it to a file on a floppy disc. I would take it into the centre and try to open it on one of the computers with Netscape browser installed. If I had made an error then the page wouldn't display and I would look at my code to find out and fix the error I had made.

The incentive to solve these problems was high, as by publishing my writing on the web I would get interesting feedback and information from readers via email. The challenge of problem solving combined with the desire to engage in an emerging community created a far stronger incentive to learn than anything I had experienced at University. This experience influenced my interest in trying out similar approaches to learning in the classroom?

2.1 Student engagement and context

This exploration took place within the context of my placement school in Oldham. It is a mixed secondary Academy with exam results higher than the national average despite intake from a surrounding area of socio-economic hardship. There is an associated pressure on staff to achieve high results.

Computing is not offered at GCSE but a new Head of Department wants to introduce more computing elements at KS3. He is not alone. The hour of code programme encourages children around the world to learn to code. Commentator Douglas Rushkoff appeals to society to engage and understand the technology that is increasingly prevalent in his book Programme or be Programmed (2010)

However, colleagues at my placement identified text coding as a challenging area and it has not been taught in the school yet. Teachers were nervous about the effect the abstract nature of the subject material would have on the behaviour and engagement of students.

Low engagement is not just a problem at school. It is also reflected in low engagement in the modern workplace (Price, 2013, p. 91). Associated figures are bleak. 10% of children say they hate school (Gilby et al., 2008). The number of year 9 to 11 students who self identified as disengaged vary from 20-33 % (Ross, 2009). Furthermore, Price (2013) challenges these figures stating that a much higher number of students are disengaged but maybe not as visibly. However, interestingly, disengagement levels for ICT are lower than many other subjects. This may be due to students seeing the relevance of ICT to the outside world (Ross, 2009, p. 27).

I hoped my choice of programming webpages in Javascript would improve engagement compared to other text-based languages as students would be able to see the link to their own experience. After all, nearly all students at KS3 are using the web and web-based phone applications daily.

Evidence does show that learners are more engaged with subject matter which they see as relevant to their lives (Marzano et al., 2001). However the following extract reveals a key dilemma.

> If the task is judged important, if the probability of success is high, and a positive affect is generated of associated with the task, the individual will be motivated to engage in the new task

(Marzano et al., 2001).

The probability of success has to be high. Making webpages with Javascript contains potential conceptual blocks that prevent an easy trajectory from initial steps to creative use of the technology in situ. Specifically, these include barriers of syntax errors and the complexity interaction between three technologies of HTML, CSS and Javascript.

After negotiating with the department I was given the opportunity to try out my scheme of work with one Year 9 class. This group had started a short unit using Dreamweaver to create web pages, although this had not covered any of the underlying HTML code.

The data of the class showed that they were under-performing to a very high degree with all students apart from one working at a level 5 for ICT at autumn grades when their targets for Year 9 were levels 6 and 7. ICT has a low status in the school with only two small GCSE classes and an even smaller number of students having selected it for next year. No students in this class had chosen ICT as an option.

2.2 Constructivist and Constructionist approaches

At this point in the study it is worth outlining constructivist and constructionist approaches to learning. Piaget explored the active nature of learning (1952). His work emphasised the role of the construction of knowledge rather than passive reception as an empty vessel to be filled with knowledge. One of his students, Papert, drew on these internal constructivist concepts and created a new focus on the importance of more public, externalised constructionist environments (Papert, 1993, 1980).

Papert's work both created and analysed these constructionist environments as educational 'worlds'. His use of the mediums of LOGO programming language and computer aided design tools provided a template which is still being followed in progressive, experimental educational programmes.

His successes involved flipping the motivation of the student from being externally imposed to being directed and discussed by the student. The teacher therefore becomes more of a facilitator than instructor. This thinking reflected a broader movement away from an instruction-based learning which was viewed as repressive and restrictive by many educational theorists (Illich, 1971).

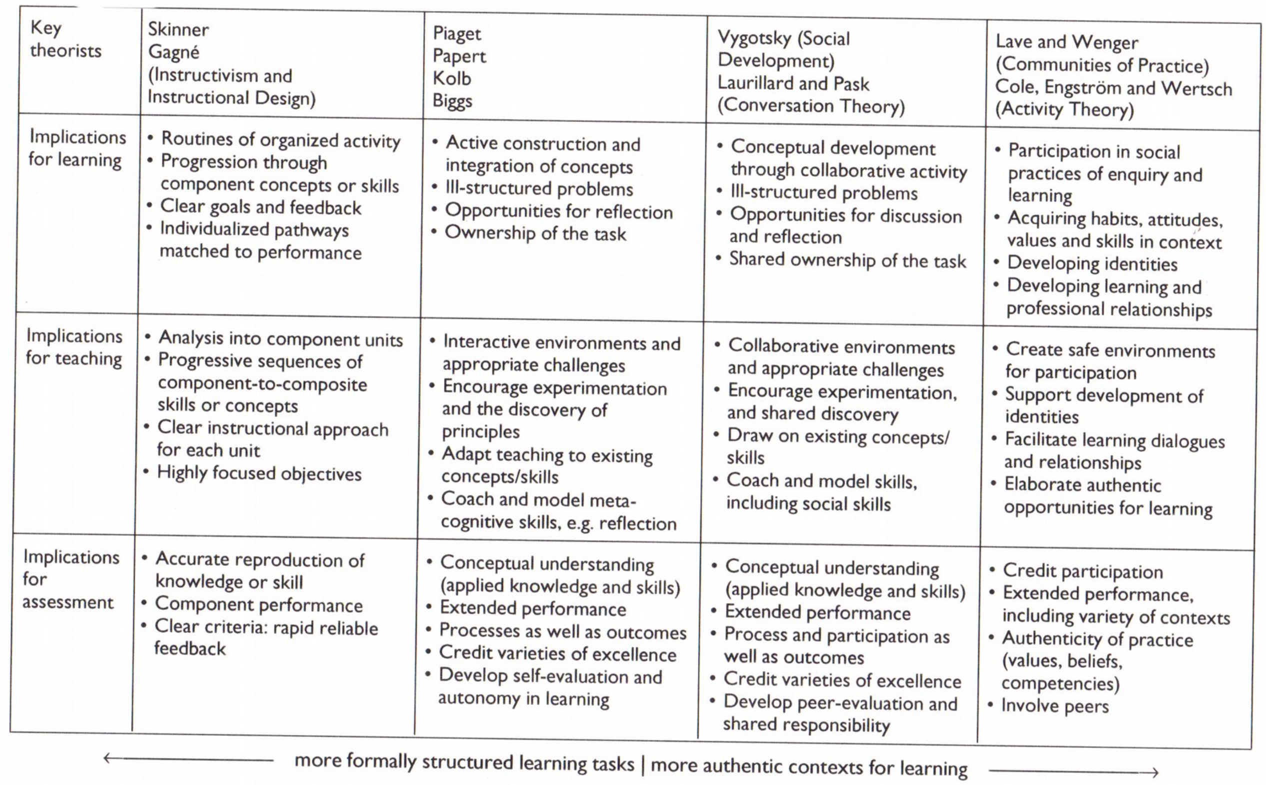
There are many overlapping theories and definitions of different educational approaches. This study uses a framework proposed by Greeno et al (1999) and subsequently developed in the context of digital approaches of learning by Mayes and De Freitas (2013). Four types of learning are proposed:

- Associative – key theorists Skinner, Gagne
- Constructive (individual) – key theorists Piaget, Papert
- Constructive (social) – key theorists Vygotsky, Laurillard
- Situative – key theorists Lave, Wegner

The approach does not value any one type higher than others but sees them as overlapping techniques each with merit. If we examine the definitions of the approaches in Illustration 1, it can be seen that much school based activities fit within the associative range. When possible this curriculum has tried to design activities that also include constructivist approaches including the following (Beetham, 2013, chap. Appendix 1):

- designing authentic and realistic activities

- problem based learning
- processes as well as outcomes
- promoting conceptual understanding (applying knowledge and skills)



Key theorists	Skinner Gagné (Instructivism and Instructional Design)	Piaget Papert Kolb Biggs	Vygotsky (Social Development) Laurillard and Pask (Conversation Theory)	Lave and Wenger (Communities of Practice) Cole, Engström and Wertsch (Activity Theory)
Implications for learning	• Routines of organized activity • Progression through component concepts or skills • Clear goals and feedback • Individualized pathways matched to performance	• Active construction and integration of concepts • Ill-structured problems • Opportunities for reflection • Ownership of the task	• Conceptual development through collaborative activity • Ill-structured problems • Opportunities for discussion and reflection • Shared ownership of the task	• Participation in social practices of enquiry and learning • Acquiring habits, attitudes, values and skills in context • Developing identities • Developing learning and professional relationships
Implications for teaching	• Analysis into component units • Progressive sequences of component-to-composite skills or concepts • Clear instructional approach for each unit • Highly focused objectives	• Interactive environments and appropriate challenges • Encourage experimentation and the discovery of principles • Adapt teaching to existing concepts/skills • Coach and model meta-cognitive skills, e.g. reflection	• Collaborative environments and appropriate challenges • Encourage experimentation, and shared discovery • Draw on existing concepts/ skills • Coach and model skills, including social skills	• Create safe environments for participation • Support development of identities • Facilitate learning dialogues and relationships • Elaborate authentic opportunities for learning
Implications for assessment	• Accurate reproduction of knowledge or skill • Component performance • Clear criteria: rapid reliable feedback	• Conceptual understanding (applied knowledge and skills) • Extended performance • Processes as well as outcomes • Credit varieties of excellence • Develop self-evaluation and autonomy in learning	• Conceptual understanding (applied knowledge and skills) • Extended performance • Process and participation as well as outcomes • Credit varieties of excellence • Develop peer-evaluation and shared responsibility	• Credit participation • Extended performance, including variety of contexts • Authenticity of practice (values, beliefs, competencies) • Involve peers

← more formally structured learning tasks | more authentic contexts for learning →

Illustration 1: Different learning environments (Mayes & De Freitas) – The columns from left to right represent: associative, constructive (individual), constructive (social), situative

Constructivist learning happens most effectively when students can construct and share external artefacts (Papert, 1993, 1980). We can look to other research for inspiration including the use of computer game design to engage young students in coding (Kafai, 1995; Kafai and Resnick, 1996).

The work of Resnick in this area has contributed to the tool Scratch which is widely used at primary and KS3 to introduce computational thinking. It uses a scaffolding system called 'block coding'. MIT, Scratch's creators, has been successful in convening a community of users sharing their Scratch games and creations (Fields et al., 2013; Resnick, 2012). The team have also used this experience to create guides for those interested in creating similar constructivist educational environments (Resnick and Silverman, 2005). There are some particular tips of Resnick and Silverman (2005) that seem particularly relevant to this study:

- DESIGN FOR DESIGNERS: create tools that learner can use to design their own artefacts
- LOW FLOOR AND WIDE WALLS: allow users to get started quickly but then continue to find new ways of using them
- SUPPORT MANY PATHS, MANY STYLES: tools should allow for great divergence in creative approaches
- CHOOSE BLACK BOXES CAREFULLY: decisions to simplify tools have a great impact on what students learners

- A LITTLE BIT OF PROGRAMMING GOES A LONG WAY: learning contexts should be able to allow creativity with only basic programming concepts

This study's curriculum is tested against some of these concepts in the second part of this study.

The Maker Movement has also adopted new technology of 3D printing, low cost robotics and open hardware to create rich exploratory learning kits (Vossoughi and Bevan, 2015).

2.2.1 Webmaking as an engaging way to teach coding

The Mozilla Foundation has embraced a constructivist approach for its educational programme. They have appropriated the terminology of the 'Maker Movement' and is promoting a set of tools and a participatory community to undertake 'Webmaking' (Mozilla Foundation, 2015a, 2014). The motivations are not purely to teach computing but also to promote wider connections between individuals whilst also promoting an appreciation of the creative and social possibilities offered by the web.

These approaches resonated with me as they reflected what had motivated me to learn web coding. In this curriculum I try to design activities in the spirit of 'Webmaking'. I hope this contribution to existing Webmaker resources will help to enrich an emerging constructivist environment for teaching text based coding, computational thinking and self expression.

2.3 Contents of the Unit

In order to follow the guideline of 'low walls', the early sessions created for this unit only deal with very simple programming concepts. They include a mixture of elements of motivational presentation, theory and icebreaker exercises. Simple webmaking activities allow students to alter (remix) existing pages. They reinforce these starting coding concepts with hands-on work.

The second half of the curriculum features a greater emphasis on constructivist activities. Creative tasks are assigned targets to different levels of confidence based on the more instructional work completed in the first lessons.

The curriculum as presented in the Appendix 1 is the latest version of several iterations based on delivery in school, at Manchester Coder Dojo and the latest version of resources created for feedback and dissemination as part of the Mozilla 'Teach the Web' program.

Details of the structure of initial lessons taught in school iteration of this course is available in the form of lesson plans in Appendix 3.

The most recent curriculum is available as a series of web pages here: http://mickfuzz.github.io/webmaker-curriculum/QuackingJavascript.

While Appendix 1 contains the contents of these web lessons in the form of a printable document, readers of this study are strongly encouraged to view the course contents as web pages online. Details are lost in translation from one format to another and not all hyperlinks will work.

There is a complexity involved in the interaction of Javascript, HTML and css elements needed to create an interactive web page. After discussions with my Head of Department, I took his advice of trying to focus as much as possible only on the Javascript.

I started to design activities where students interact with and alter the Javascript in real webpages but where other elements are shielded as much as possible. The following table contains a quick summary of what is covered in each session and the aims of activities created.

	Subject contents	Key activities
Session One	Variables and values	Thimble remix activity Katy Perry's tour name.
Session Two	Maths, operators and data types	Hack Cookie Clicker game & T-shirt shop Thimble remix
Session Three	Procedural approaches and HTML	Create a web page step by step using Javascript variables and functions
Session Four	Selection, if statements and code blocks	Daddy or Chips, video, game and Thimble remix activity
Session Five A	Iteration, arrays and for loops	Thimble Remix activities to output Swimming Pool times
Session Five B	Exploratory constructive lesson	Focus on creating web pages from scratch, bug fixing and creative coding

The learning outcomes match areas from the most recent government recommendations for KS3 curriculum including text based coding and understanding algorithms (Department for Education, 2015).

More details of the learning activities and analysis of the impact of these learning episodes is included in part two of this study.

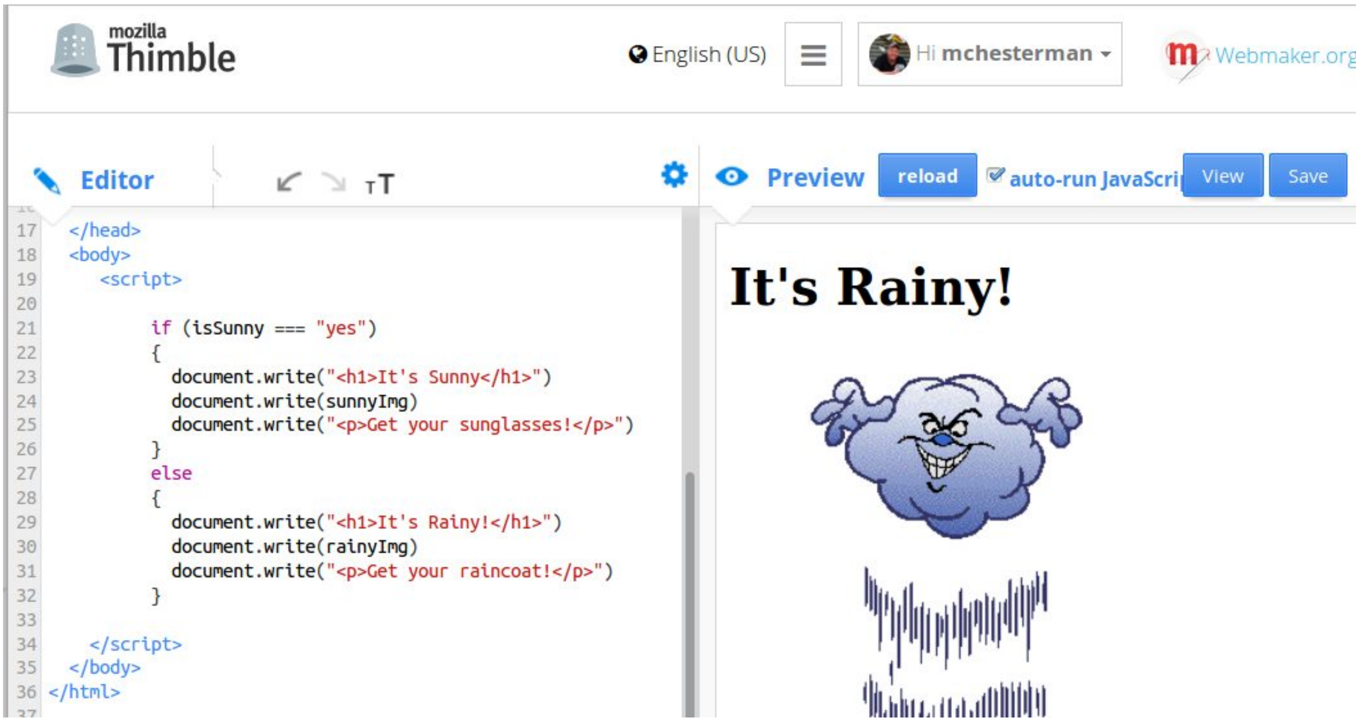
2.4 Choice of Tools

I wanted students to use the familiar environment of the browser as one of their main tools for learning. I also wanted the web authoring environment to be real, thus giving possibility to extend learning into the creation of games and real life web projects. This is compared to tools which teach web coding in artificial, sandboxed environments like Codecademy.

In addition to the use of an Internet Browser for previewing and testing pages I decided to trial the use of Dreamweaver and Thimble as an experiment to compare the strengths and weaknesses of both applications in the context of this study.

Dreamweaver is a web design application from Adobe. It is present on most school networks as part of the Adobe suite of products.

Thimble is an online tool for creating and experimenting with web pages created by the Mozilla Foundation for use in informal and self-directed education. It has a side-by-side code and preview windows as shown in Illustration 2. Thimble does not have a What-You-See-Is-What-You-Get editor. All changes to the page must be made by updating the code.



```
  </head>
  <body>
     <script>

          if (isSunny === "yes")
          {
            document.write("<h1>It's Sunny</h1>")
            document.write(sunnyImg)
            document.write("<p>Get your sunglasses!</p>")
          }
          else
          {
            document.write("<h1>It's Rainy!</h1>")
            document.write(rainyImg)
            document.write("<p>Get your raincoat!</p>")
          }

    </script>
  </body>
</html>
```

```
It's Rainy!
```

Illustration 2: Mozilla Thimble tool with an example of a Webmaker exercise

There is a social focus to the tool. Teachers and students are encouraged to share their Thimble webpages for remixing. This process of viewing a page created by others, clicking on remix, and then investigating and altering the code seems a solid base for a constructivist learning approach.

3 PART TWO: Reflective account and discussion of findings

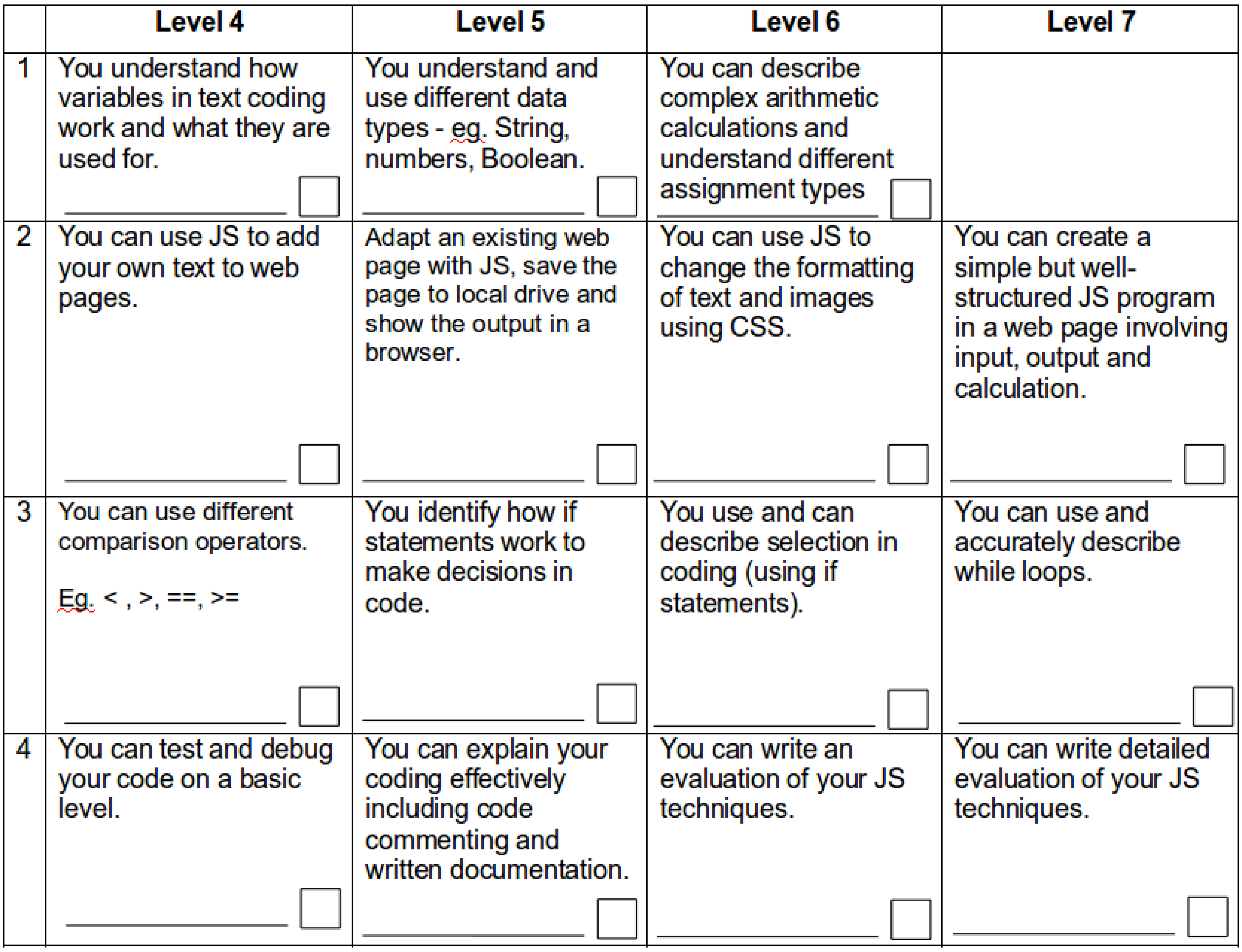
3.1 Assessment of student work and progress

This section begins evaluating the more formal assessment methods, moves on to informal approaches and then considers potential developments in this area. In breaking down the learning of the course in to discreet chunks with boxes to be taught and assessed I was aware that this could limit the potential of constructive and situative learning. My aim was to try to limit the impact grade fixation (Wiliam, 2011), emphasise the importance of process-driven learning whilst still fulfilling the departmental need to provide levels and show weekly progress.

3.1.1 Departmental format assessment sheet

I created an assessment sheet in the same style as other ones created by the ICT department. I also used this as a basis for the self-assessment of learning goals, an approach supported by research on formative assessment (Boekaerts and Corno, 2005). The sheet was based on the KS3 curriculum from CAS and similar learning objectives to the Python unit of PG Online. The sheet is included as Appendix 2m and an extract is shown in Illustration 15.

I scheduled time in plenaries to allow this assessment sheet to be ticked and dated by students as a form of self-assessment and reflection.



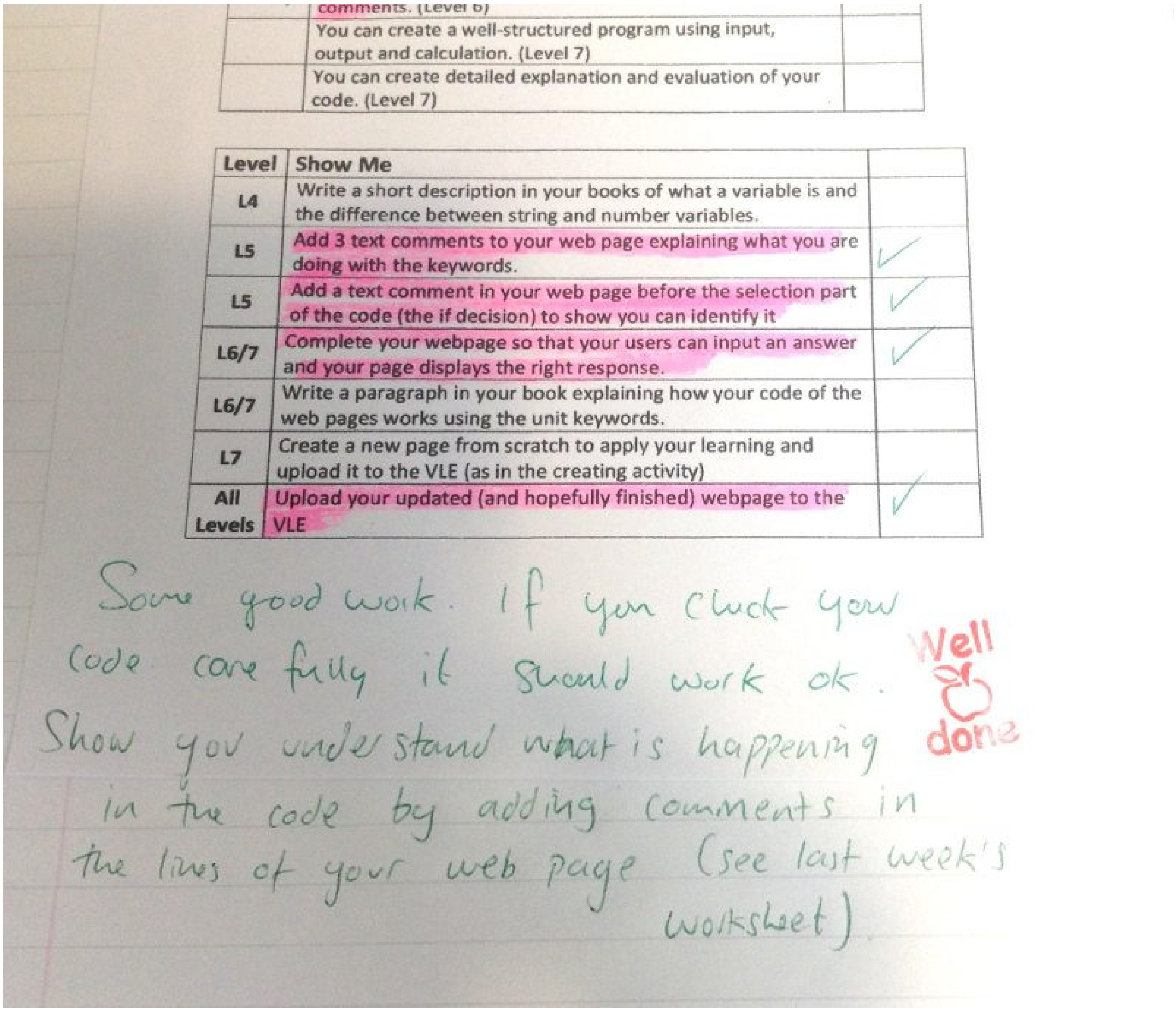
	Level 4	Level 5	Level 6	Level 7
1	You understand how variables in text coding work and what they are used for. ______ □	You understand and use different data types - eg. String, numbers, Boolean. ______ □	You can describe complex arithmetic calculations and understand different assignment types □	
2	You can use JS to add your own text to web pages. ______ □	Adapt an existing web page with JS, save the page to local drive and show the output in a browser. ______ □	You can use JS to change the formatting of text and images using CSS. ______ □	You can create a simple but well-structured JS program in a web page involving input, output and calculation. ______ □
3	You can use different comparison operators. Eg. < , >, ==, >= ______ □	You identify how if statements work to make decisions in code. ______ □	You use and can describe selection in coding (using if statements). ______ □	You can use and accurately describe while loops. ______ □
4	You can test and debug your code on a basic level. ______ □	You can explain your coding effectively including code commenting and written documentation. ______ □	You can write an evaluation of your JS techniques. ______ □	You can write detailed evaluation of your JS techniques. ______ □

Illustration 3: Assessment grid based on departmental guidelines

3.1.2 Departmental mid unit assessment and feedback for DIRT lesson

There was a set structure used in my department for a milestone assessment followed my a DIRT (Directed Improvement and Reflection Time) lesson which is based on the AFL technique of 'closing the gap' (Beere and Gilbert, 2012; Young, 2005).

This was used as a basis for feedback to the student, giving direction on how to use time in the DIRT lesson to improve the level of work. The sheet that I created is included here as Appendix 2m and an example of one used in practice as Illustration 4



	comments. (Level 6)	
	You can create a well-structured program using input, output and calculation. (Level 7)	
	You can create detailed explanation and evaluation of your code. (Level 7)	

Level	Show Me	
L4	Write a short description in your books of what a variable is and the difference between string and number variables.	
L5	Add 3 text comments to your web page explaining what you are doing with the keywords.	✓
L5	Add a text comment in your web page before the selection part of the code (the if decision) to show you can identify it	✓
L6/7	Complete your webpage so that your users can input an answer and your page displays the right response.	✓
L6/7	Write a paragraph in your book explaining how your code of the web pages works using the unit keywords.	
L7	Create a new page from scratch to apply your learning and upload it to the VLE (as in the creating activity)	
All Levels	Upload your updated (and hopefully finished) webpage to the VLE	✓

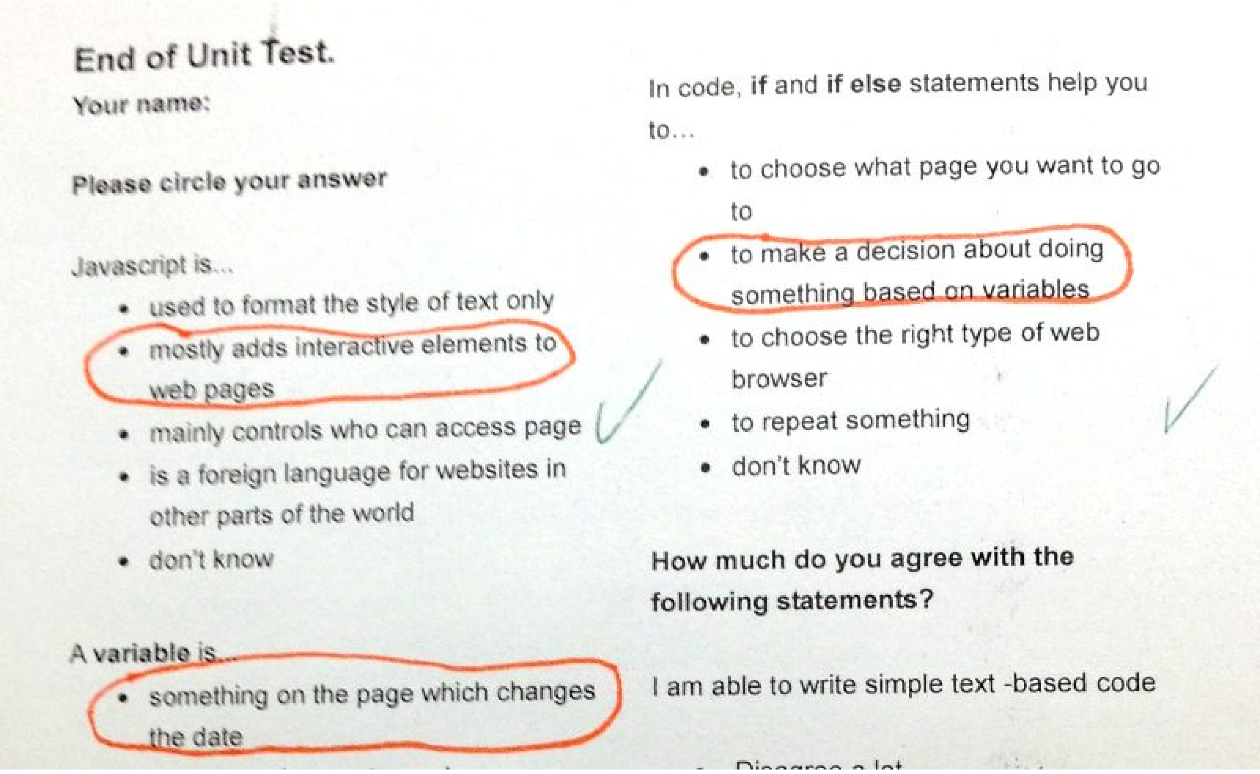
Some good work. If you check your code carefully it should work ok. Show you understand what is happening in the code by adding comments in the lines of your web page (see last week's worksheet)

Well done

Illustration 4: DIRT (closing the gap) activity feedback sheet

3.1.3 Baseline and end of unit quiz

I created a baseline / end of unit quiz which is included as Appendix 2a and a completed version as shown in Illustration 5. To ensure that the retention of data was tested, I waited until four weeks had passed since teaching the last lesson of the unit before giving the final test.



End of Unit Test.

Your name:

Please circle your answer

Javascript is...

- used to format the style of text only
- [✓ mostly adds interactive elements to web pages] ✓
- mainly controls who can access page
- is a foreign language for websites in other parts of the world
- don't know

A variable is...

- [✓ something on the page which changes the date]

In code, **if** and **if else** statements help you to...

- to choose what page you want to go to
- [✓ to make a decision about doing something based on variables] ✓
- to choose the right type of web browser
- to repeat something
- don't know

How much do you agree with the following statements?

I am able to write simple text -based code

- Disagree a lot

Illustration 5: Extract of completed end of unit test

While the contents of the test were intentionally limited in scope, the answers do provide some data that can be visualised in the following Illustrations.

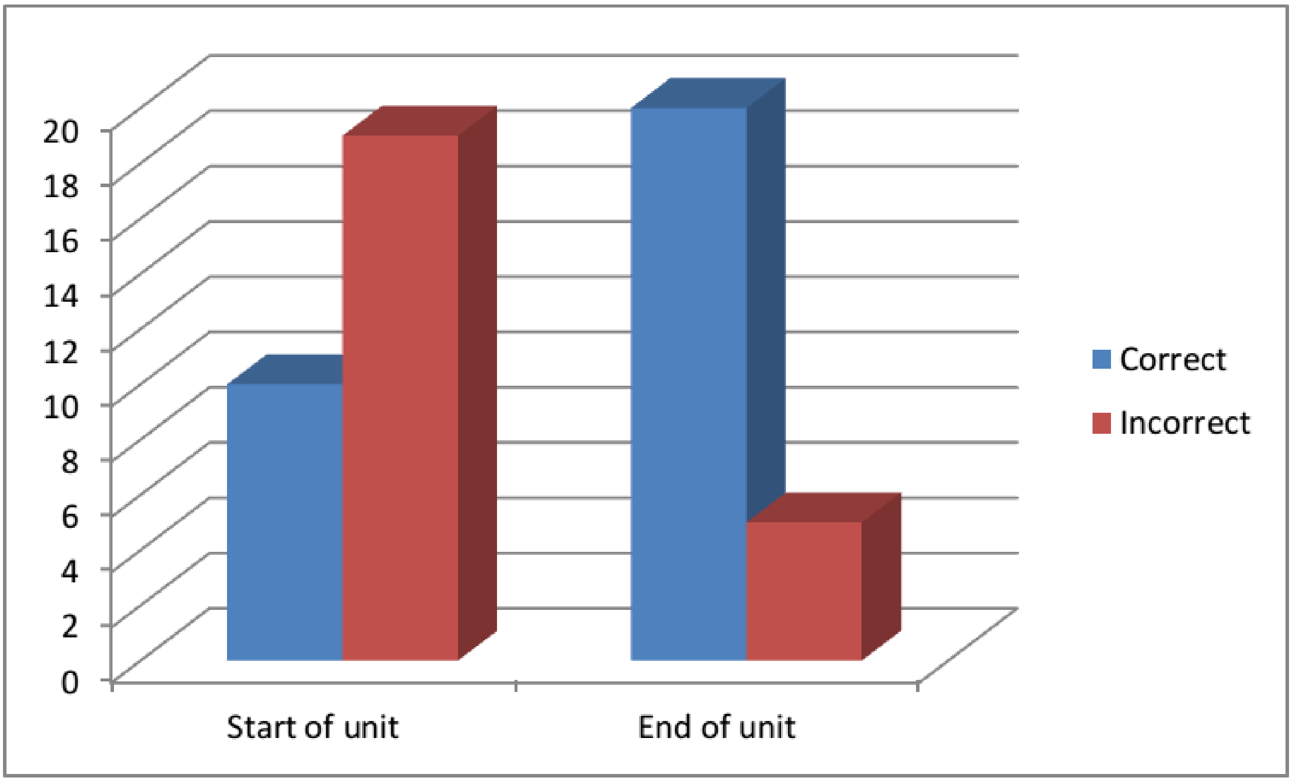


Illustration 6: Baseline quiz question one: "Javascript is..."

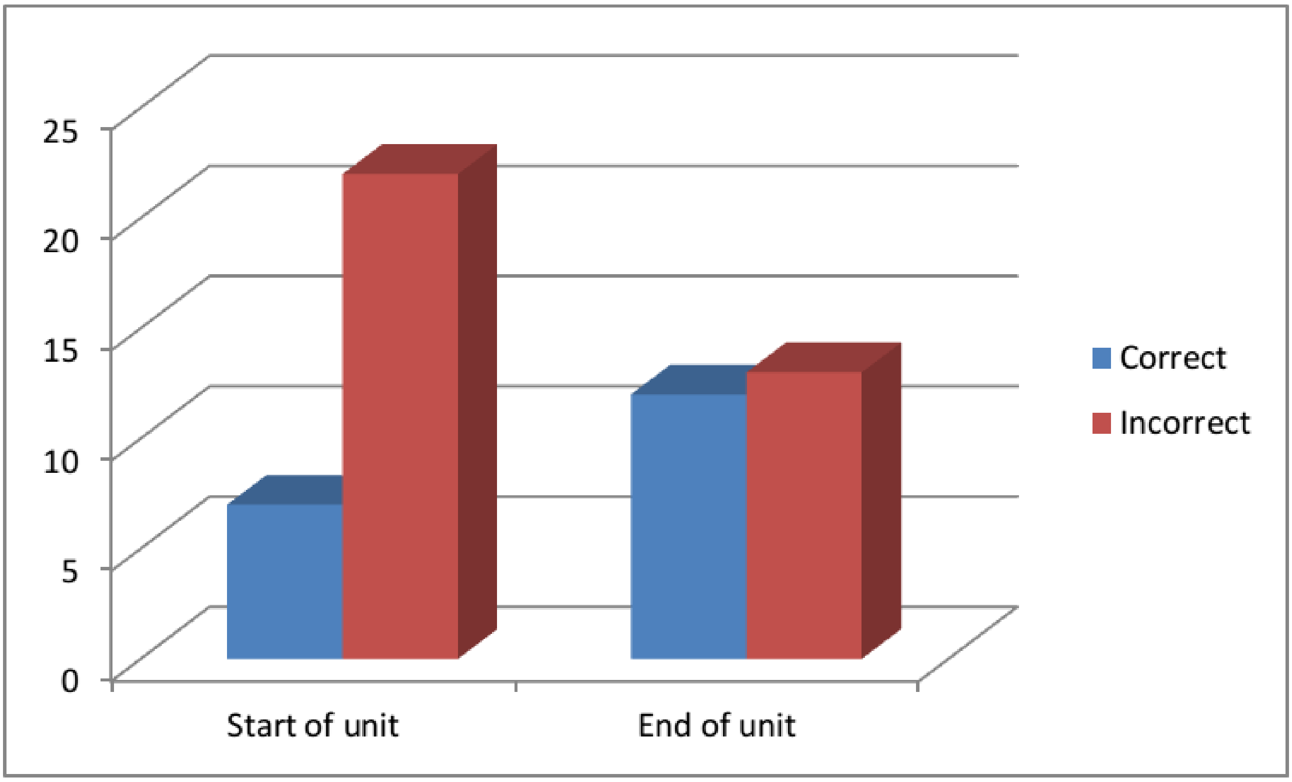


Illustration 7: Baseline quiz question two: "A variable is..."

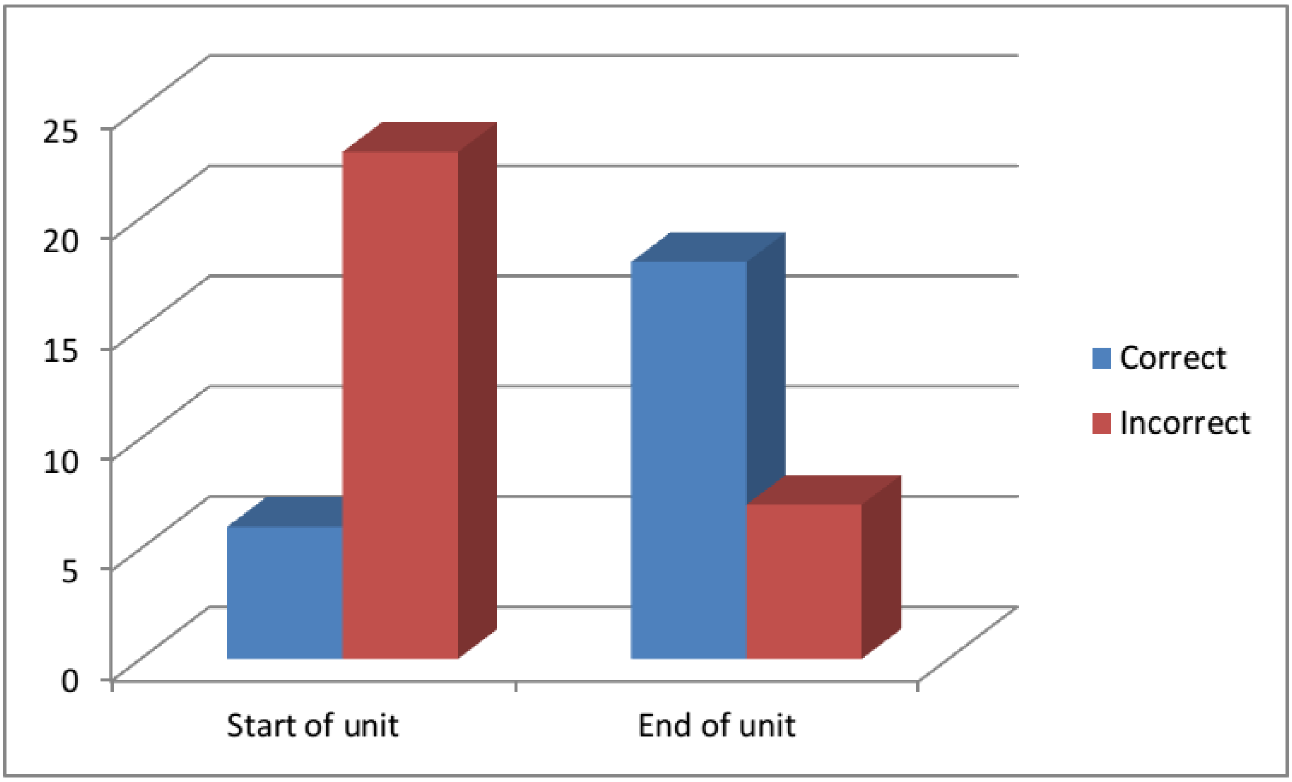


Illustration 8: Baseline quiz question three: The data type of a variable means..

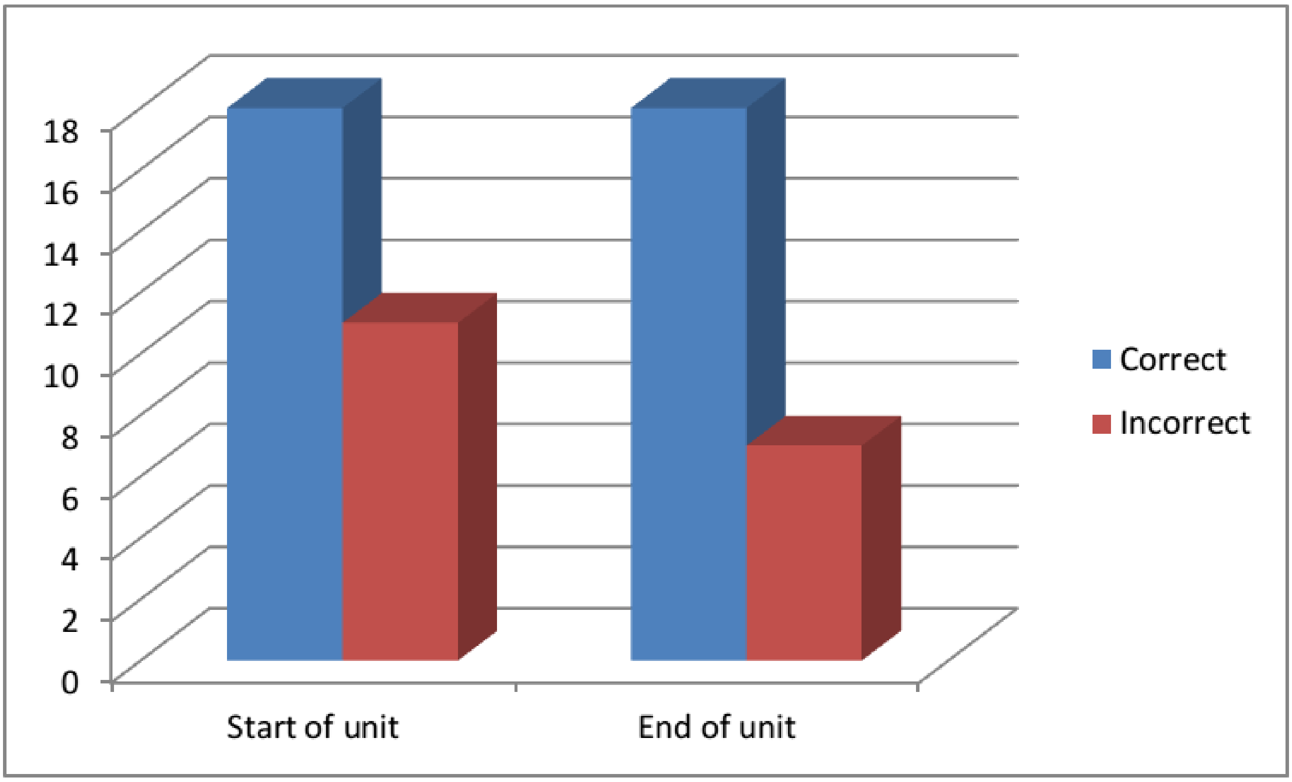


Illustration 9: Baseline quiz question four: To 'assign a value' means...

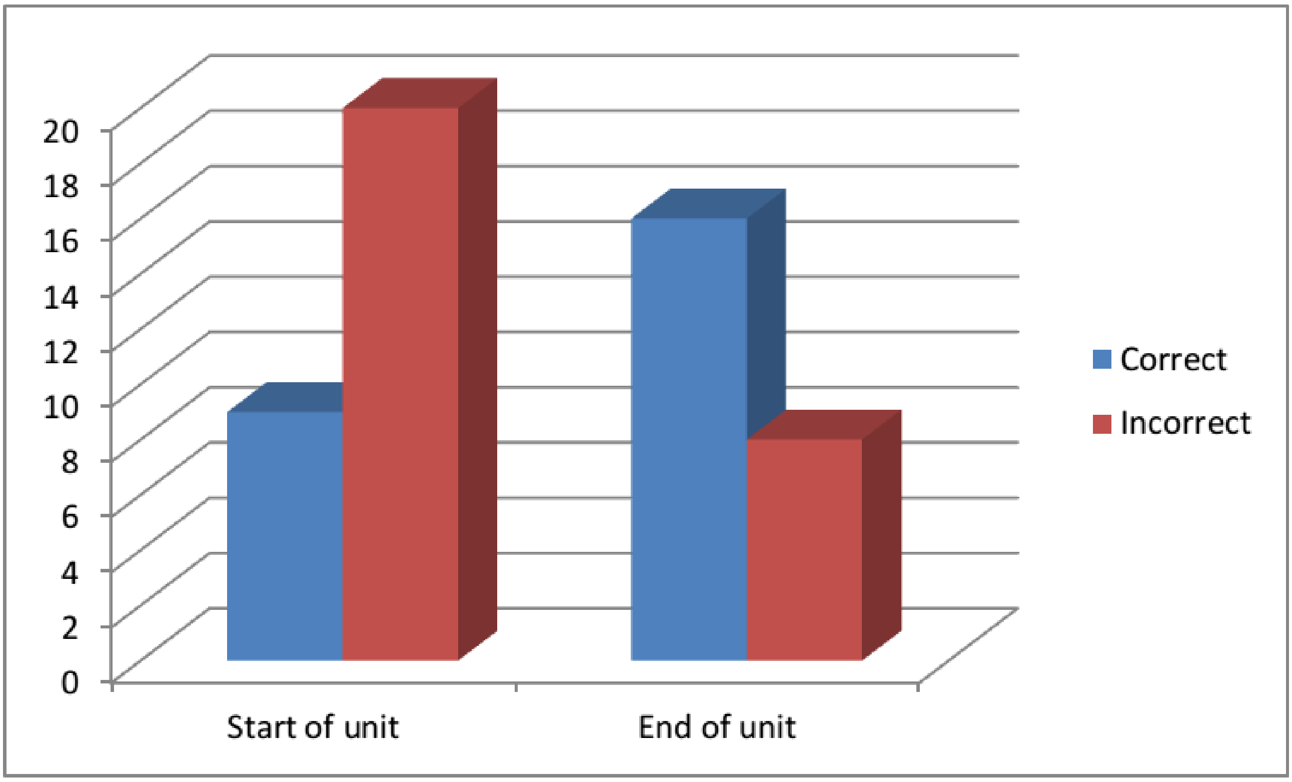


Illustration 10: Baseline quiz question five: In code if and if else statements help you to...

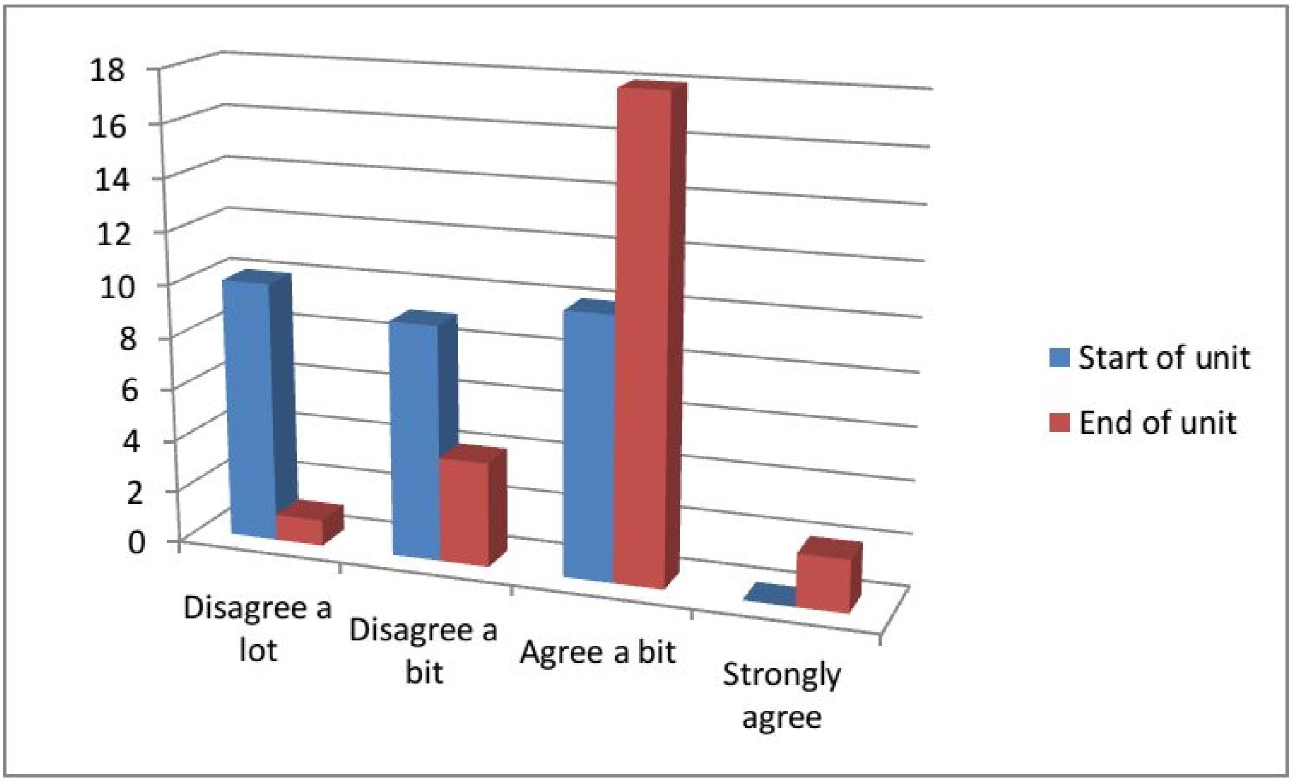


Illustration 11: Baseline quiz question six: I am able to write simple text-based code

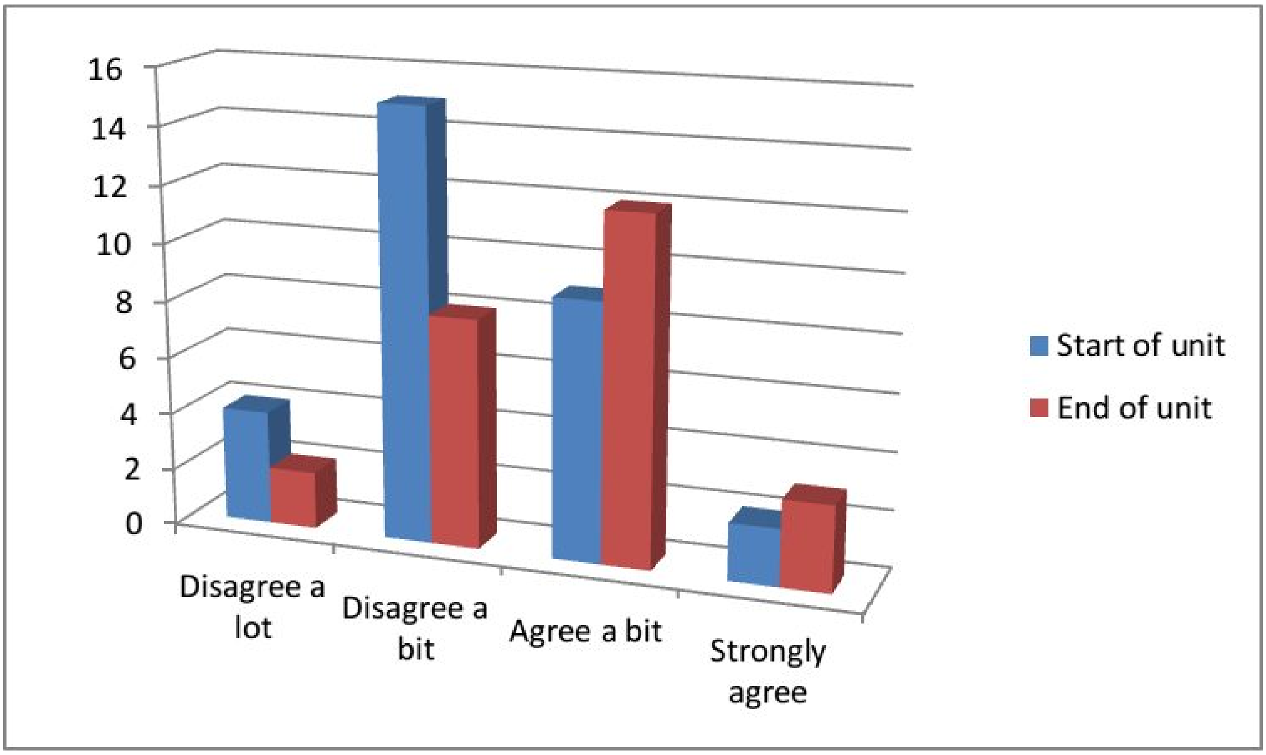


Illustration 12: Baseline quiz question seven: I know where to find help on the web to learn about coding

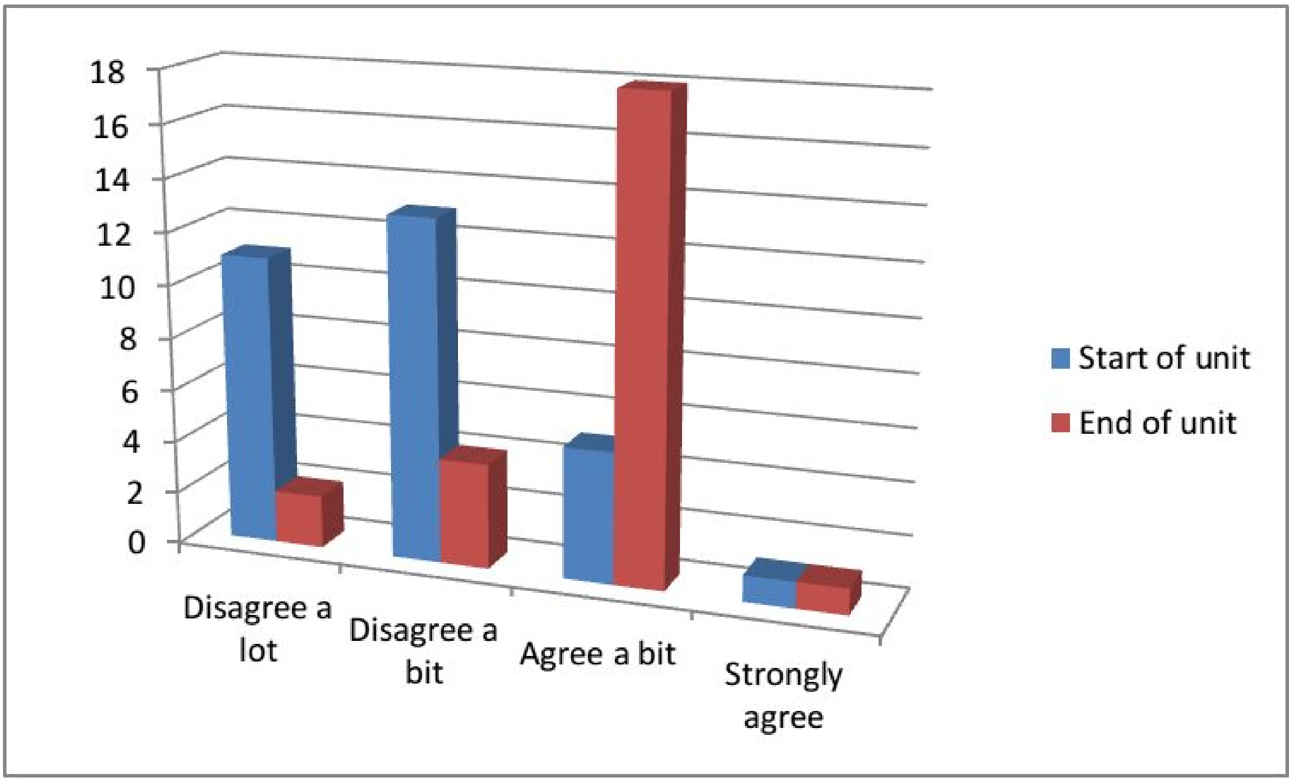


Illustration 13: Baseline quiz question eight: I know about the principles of text-based coding (like variables, data-types and if decisions)

3.1.4 Other strategies for formative assessment

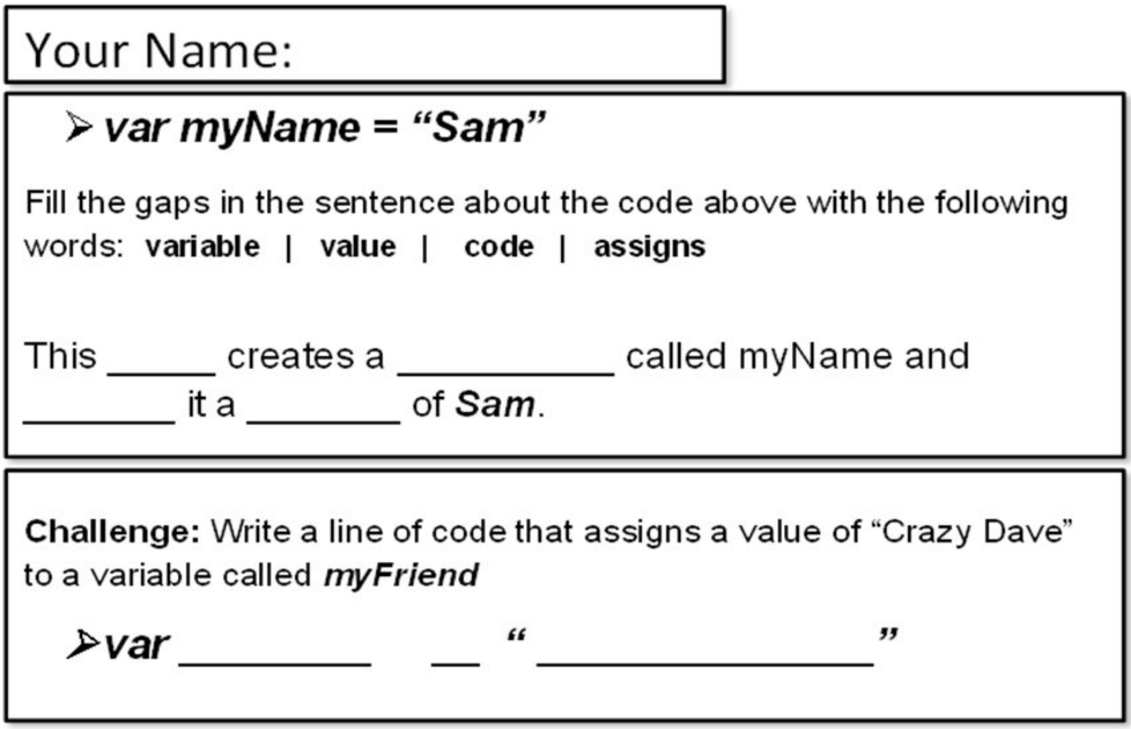
Formative assessment also know as AfL (assessment for learning) has been very influential in shaping governmental, Ofsted and school agendas (Oftsed, 2011). Section 3.1.2 dealt with a type of formative assessment called closing the gap. This section outlines other techniques.

Student questioning is a key factor in maintaining student engagement (Beaumont et al., 2011; Black and Wiliam, 1998; Wiliam, 2006). I employed several techniques to improve my questioning sessions including:

- use of a student randomiser to avoid favouring higher ability students with questions (in my case idoceo app for iPad)
- bouncing questions from one student to another
- coming back to students who answer, “I don't know.”, to show non-participation in questioning sessions is not ignored
- asking students to nominate other tables or individual students for the next question to 'gamify' the process

Traffic light feedback methods were also used. The class had red, amber and green cards in their student planer and I used these to create RAG feedback activities in mini plenaries to gauge the progress and understanding of students. Specifically, this was done before the worksheet and creative tasks were set, allowing me to set different tasks depending on responses given by students.

At the end of lessons the use of comprehension multiple choices quizzes or simple paper exit activities were effective in informing future planning of activities. One such exit activity is included as Illustration 14.



Your Name:

➢ ***var myName = "Sam"***

Fill the gaps in the sentence about the code above with the following words: **variable** | **value** | **code** | **assigns**

This _____ creates a __________ called myName and ________ it a ________ of ***Sam***.

Challenge: Write a line of code that assigns a value of "Crazy Dave" to a variable called ***myFriend***

➢***var*** _________ __ " ________________"

Illustration 14: Exit activity for lesson one

3.1.5 Improving assessment of Webmaking tasks

Assessing progress and learning processes during constructivist, creative tasks was more challenging. As an example, a student called Adam was making good progress on Webmaking tasks. However, he was frustrated, as errors in his code were preventing the correct display of his chosen images.

We compared actual results with expected results and looked at his code. Adam was using variables effectively. He was also using if / else code blocks and functions. He had identified some syntax errors and had debugged them.

In the plenary, I used this example to give feedback on skills being learned by students. I asked students to be aware of all the different processes they were learning and the progress being made and to update their self assessment sheets. If they had fixed an error, or shown understanding of objectives they were asked to tick off the relevant outcomes.

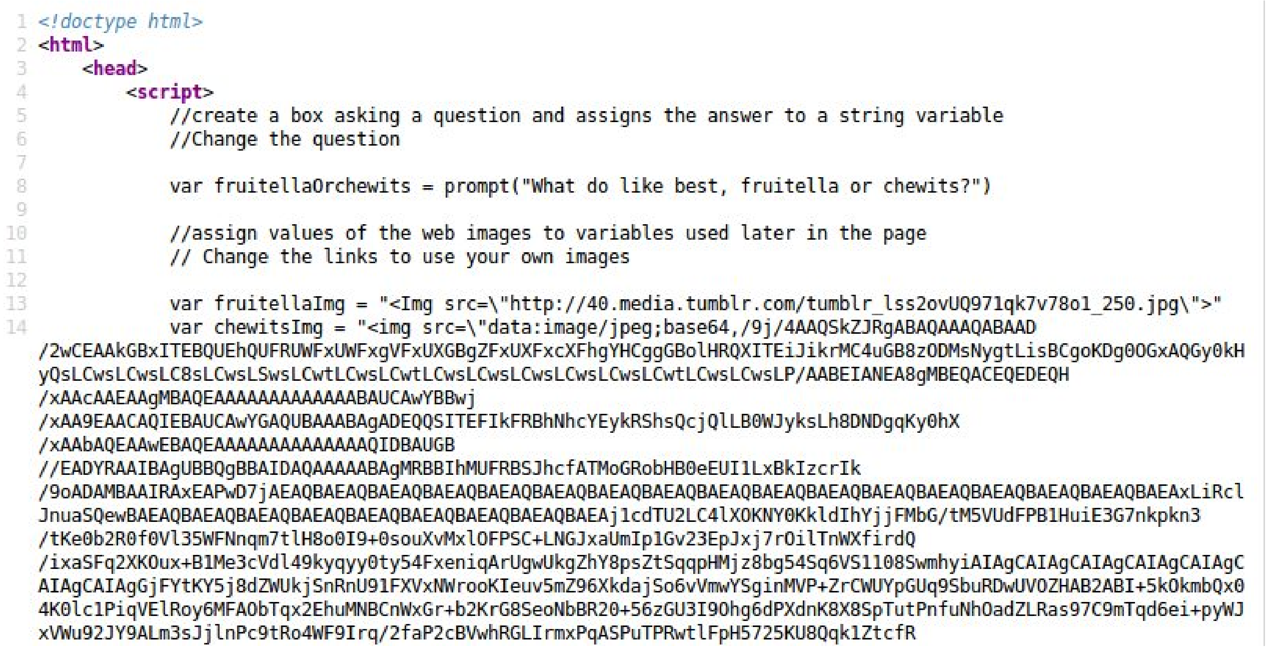
In teaching this unit I experienced a dilemma of whether to prioritise the use of Webmaking activities or paper-based ones for assessment. With only six one-hour lessons, a safer approach is to focus on paper based implementation which is easy to mark and demonstrate in books. However, the tendency to 'play it safe' often results in the overuse of associative learning for students.

The situation may be improved with increased use of peer feedback. In this context it could promote procedural skills of bug checking and testing readability of code. Increased peer assessment could also act as a way of encouraging greater creativity and deviation from the guided tasks.

3.2 Evaluation of specific curriculum elements

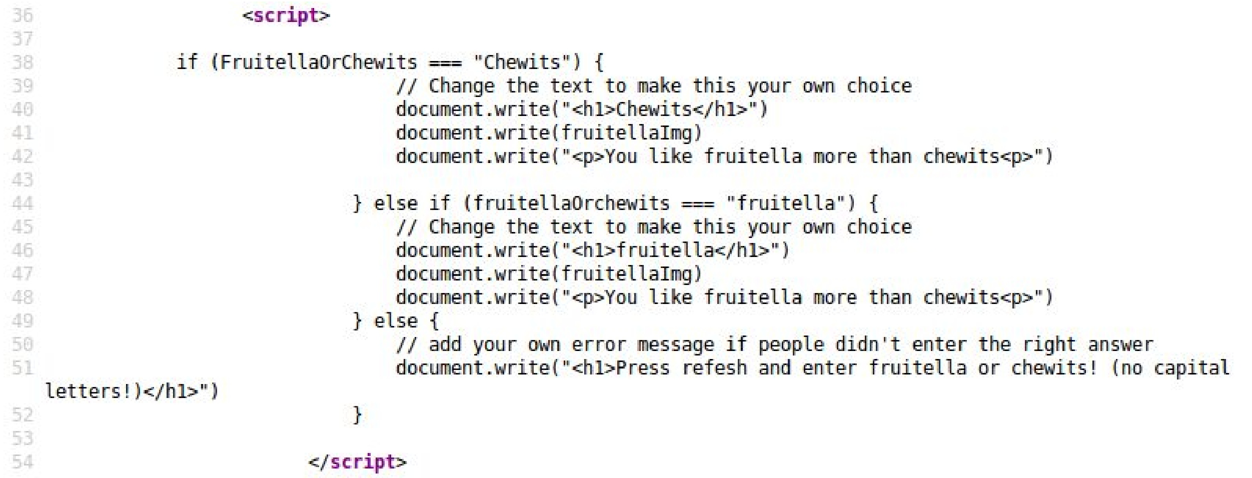
3.2.1 Identifying and addressing areas of misconceptions emerging from assessment

While some students, completed Webmaking tasks correctly on the first attempt, most showed errors of various types. Collecting student work showed areas of common misconceptions. Analysis of work was used to inform debugging sessions in ongoing lessons. Examples of common mistakes are shown in the following illustrations.



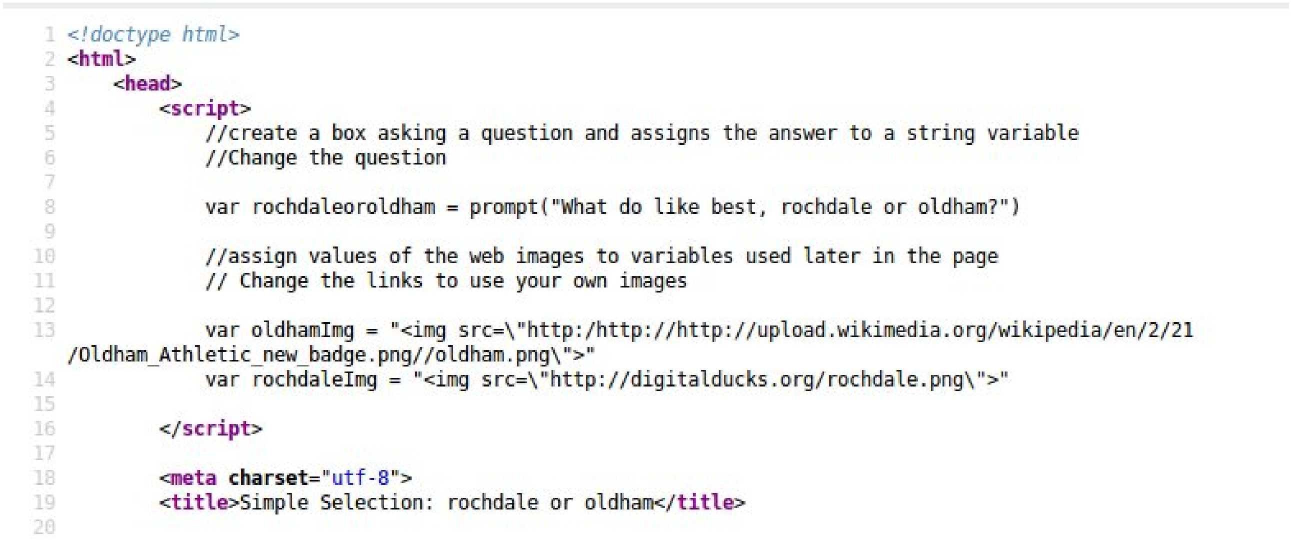
```
<!doctype html>
<html>
   <head>
      <script>
         //create a box asking a question and assigns the answer to a string variable
         //Change the question

         var fruitellaOrchewits = prompt("What do like best, fruitella or chewits?")

         //assign values of the web images to variables used later in the page
         // Change the links to use your own images

         var fruitellaImg = "<Img src=\"http://40.media.tumblr.com/tumblr_lss2ovUQ971qk7v78o1_250.jpg\">"
         var chewitsImg = "<img src=\"data:image/jpeg;base64,/9j/4AAQSkZJRgABAQAAAQABAAD
/2wCEAAkGBxITEBQUEhQUFRUWFxUWFxgVFxUXGBgZFxUXFxcXFhgYHCggGBolHRQXITEiJikrMC4uGB8zODMsNygtLisBCgoKDg0OGxAQGy0kH
yQsLCwsLCwsLC8sLCwsLSwsLCwtLCwsLCwtLCwsLCwsLCwsLCwsLCwtLCwsLCwsLP/AABEIANEA8gMBEQACEQEDEQH
/xAAcAAEAAgMBAQEAAAAAAAAAAAAABAUCAwYBBwj
/xAA9EAACAQIEBAUCAwYGAQUBAAABAgADEQQSITEFIkFRBhNhcYEykRShsQcjQlLB0WJyksLh8DNDgqKy0hX
/xAAbAQEAAwEBAQEAAAAAAAAAAAAAAQIDBAUGB
//EADYRAAIBAgUBBQgBBAIDAQAAAAABAgMRBBIhMUFRBSJhcfATMoGRobHB0eEUI1LxBkIzcrIk
/9oADAMBAAIRAxEAPwD7jAEAQBAEAQBAEAQBAEAQBAEAQBAEAQBAEAQBAEAQBAEAQBAEAQBAEAQBAEAxLiRcl
JnuaSQewBAEAQBAEAQBAEAQBAEAQBAEAQBAEAj1cdTU2LC4lXOKNY0KldIhYjjFMbG/tM5VUdJHNnkn3
/tKe0b2R0f0Vl35WFNnqm7tlH8o0I9+0souXMvMxlOFSC+LNG2JxSp8YmuYmDJ4uGsZiG1v23EpJxj7rj
/ixaSFq2XKOux+B1Me3cVdl49kyqyy0tySfhgArUgwUkgZhY8psZtSqqpHMjy8Rh5UJZJA2iAIAgCAIAgC
AIAgCAIAgGjFYtKY5j8j8dZWUkjSnBRrUrSnRnU91FXVxroofKSmoWMhTWgwB2n4wG+HqlLPrgPacLHOPKmQDPfmv5Ha+dHTmfl6OlN6TmTXnUdbGRSC5kY9sI7f
xVWu92JY9ALm3sJjcfRo4Wc9tRo4WF9tRo/2faP2cBVwhRGLIrmxPqASPUuTPRwtl8Pk8Qi9b/nk9Tp96rCxfH5725KU8Qqk1ZtcfR
```

Illustration 15: Correct and incorrect referencing of binary data value in image URL variables



```
             <script>

         if (FruitellaOrChewits === "Chewits") {
                        // Change the text to make this your own choice
                        document.write("<h1>Chewits</h1>")
                        document.write(fruitellaImg)
                        document.write("<p>You like fruitella more than chewits<p>")

                     } else if (fruitellaOrchewits === "fruitella") {
                        // Change the text to make this your own choice
                        document.write("<h1>fruitella</h1>")
                        document.write(fruitellaImg)
                        document.write("<p>You like fruitella more than chewits<p>")
                     } else {
                        // add your own error message if people didn't enter the right answer
                        document.write("<h1>Press refesh and enter fruitella or chewits! (no capital
letters!)</h1>")
                     }

                  </script>
```

Illustration 16: Case error in selection values & Logic errors in selection code blocks

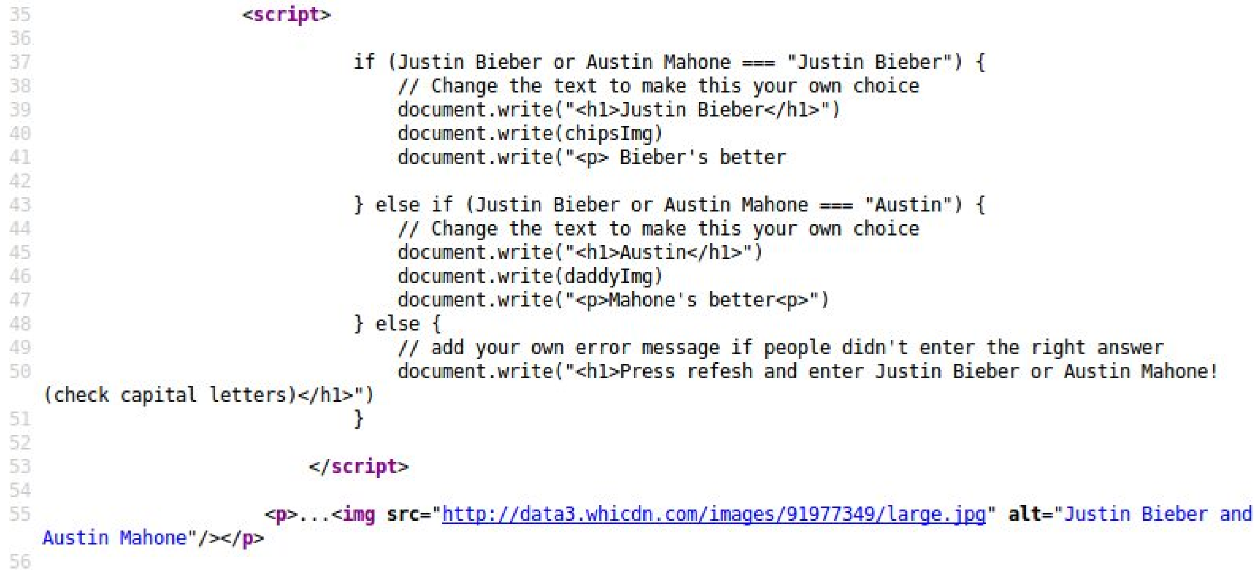


```
<!doctype html>
<html>
    <head>
        <script>
            //create a box asking a question and assigns the answer to a string variable
            //Change the question

            var rochdaleoroldham = prompt("What do like best, rochdale or oldham?")

            //assign values of the web images to variables used later in the page
            // Change the links to use your own images

            var oldhamImg = "<img src=\"http:/http://http://upload.wikimedia.org/wikipedia/en/2/21
/Oldham_Athletic_new_badge.png//oldham.png\">"
            var rochdaleImg = "<img src=\"http://digitalducks.org/rochdale.png\">"

        </script>

        <meta charset="utf-8">
        <title>Simple Selection: rochdale or oldham</title>

```

Illustration 17: Misconception and syntax errors; images URLs are incorrect

In Illustration 16 we can see some good use of selection syntax and comparision operators but some logic errors and inconsistent use of variable values.

In Illustration 17 we see a lack of care or understanding about how image URLs are formed.

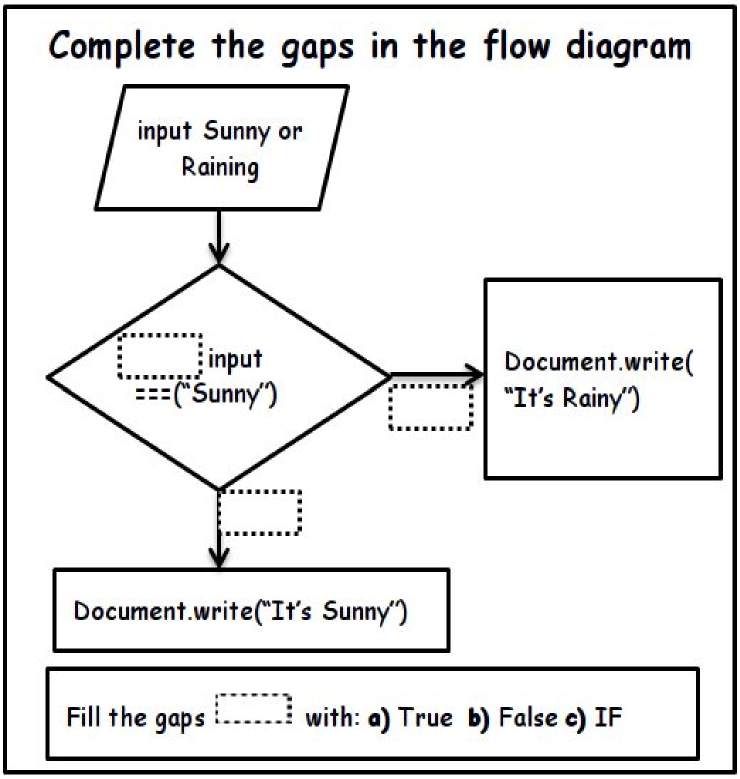


```
                  <script>

                            if (Justin Bieber or Austin Mahone === "Justin Bieber") {
                                // Change the text to make this your own choice
                                document.write("<h1>Justin Bieber</h1>")
                                document.write(chipsImg)
                                document.write("<p> Bieber's better

                            } else if (Justin Bieber or Austin Mahone === "Austin") {
                                // Change the text to make this your own choice
                                document.write("<h1>Austin</h1>")
                                document.write(daddyImg)
                                document.write("<p>Mahone's better<p>")
                            } else {
                                // add your own error message if people didn't enter the right answer
                                document.write("<h1>Press refesh and enter Justin Bieber or Austin Mahone!
(check capital letters)</h1>")
                            }

                        </script>

                    <p>...<img src="http://data3.whicdn.com/images/91977349/large.jpg" alt="Justin Bieber and
Austin Mahone"/></p>

```

Illustration 18: Syntax error with spaces in variable name

In Illustration 18 we can see that variable names containing spaces will stop the code from executing correctly.

Feedback from RAG quizzes showed that a significant section of the class had not fully understood initial exercises on if / else selection statements. To address this I structured a follow up worksheet. Students were asked to perform different activities based on their previous understanding. Weaker students undertook a revision task shown in Illustration 19.



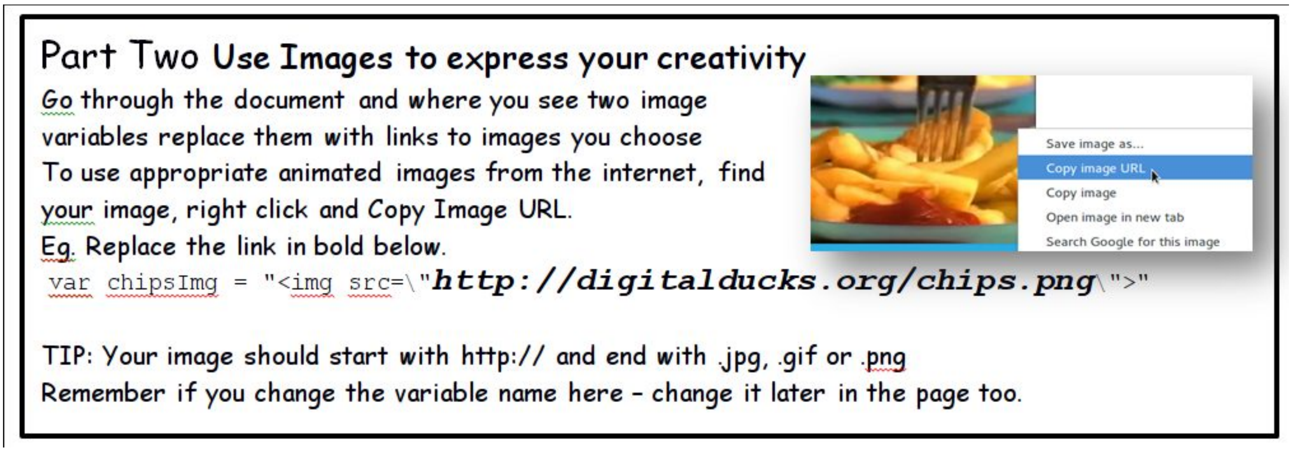
Complete the gaps in the flow diagram

Fill the gaps with: a) True b) False c) IF

Illustration 19: Differentiated reinforcement activity on Selection for worksheet

I tried to design activities with a focus on Javascript so that students did not need to know HTML. However, it was clear from many of these errors and from student feedback, that in order to work with images, some teaching of HTML was needed to avoid excessive syntax errors.

To help correct common HTML mistakes, I took screenshots of the various types of errors committed by students and included them in presentations. Solutions to common errors were included in worksheets as a handy reference (as shown in Illustration 20).



Part Two Use Images to express your creativity

Go through the document and where you see two image variables replace them with links to images you choose

To use appropriate animated images from the internet, find your image, right click and Copy Image URL.

Eg. Replace the link in bold below.

```
var chipsImg = "<img src=\"http://digitalducks.org/chips.png\">"
```

TIP: Your image should start with http:// and end with .jpg, .gif or .png

Remember if you change the variable name here - change it later in the page too.

Illustration 20: Worksheet extract with aim to address misconceptions

3.2.2 Changing the approach of the main learning tasks

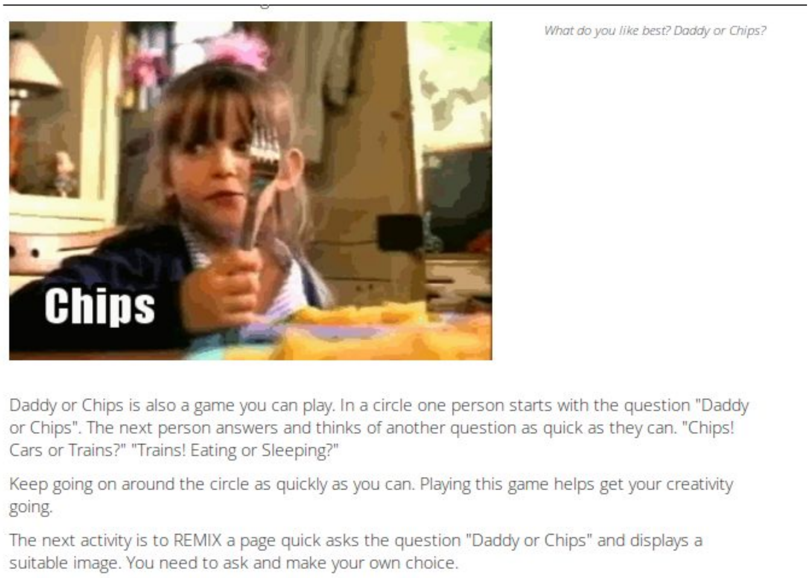
As the course progressed, there was a divergence between the original, imagined approach and the one adopted for classroom delivery. Despite feedback to students that making mistakes shows progress there were on-going frustrations from the class. Due to this, I adopted a more instruction-based approach for activities than the more exploratory approach of ill structured problems (Jonassen, 1997).

This shift from a constructive approach to an associative one yielded immediate results. A much greater percentage of students made progress against the specific outcomes of each lesson. However, my concern is that the process of atomising the material and emphasising prescriptive progress over more student driven tasks may negatively effect long-term student engagement and reduce the potential for higher order learning. An alternative strategy is explored in Section 3.3.1.

3.2.3 Use of drama games, computer games, videos and popular culture to create an engaging environment

The use of the games (both drama and computer based), videos and popular culture proved to be good ways to increase student engagement. They also helped to generate student material designed to facilitate creative work.

One example of this is the Daddy or Chips video and generative word game which is outlined in Appendix 1 in the lesson on Selection.



What do you like best? Daddy or Chips?

Daddy or Chips is also a game you can play. In a circle one person starts with the question "Daddy or Chips". The next person answers and thinks of another question as quick as they can. "Chips! Cars or Trains?" "Trains! Eating or Sleeping?"

Keep going on around the circle as quickly as you can. Playing this game helps get your creativity going.

The next activity is to REMIX a page quick asks the question "Daddy or Chips" and displays a suitable image. You need to ask and make your own choice.

Illustration 21: Building engagement using video, games and pop culture

This success lead me to create a section on other appropriate drama / word games which could be used by other facilitators. Rather than embed them into each lesson, they are included in the teacher's notes section on the index page of the curriculum in Appendix 1.

Hacking the Cookie Clicker game was one of the most immediately appealing activities for disengaged students. It featured highly in student plenary responses about what had been learned. Cookie Clicker is an online game that was known to nearly all students.

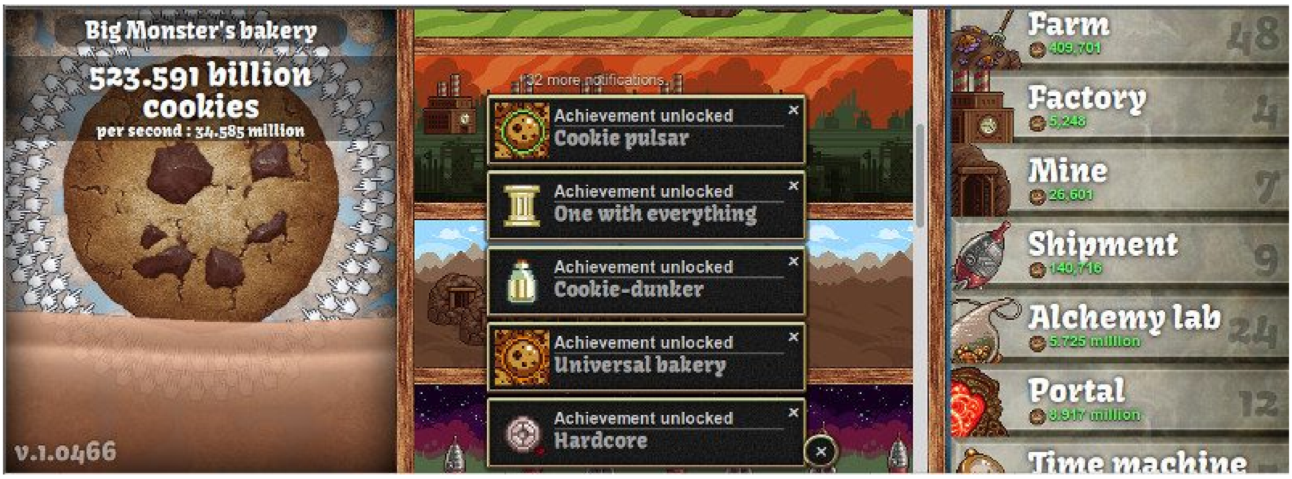
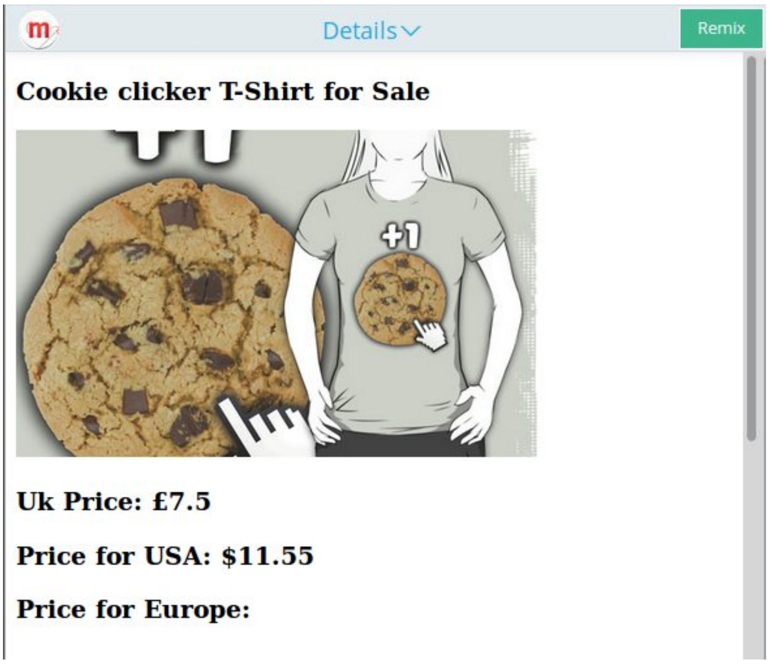


Illustration 22: Use of a familiar online games to increase student engagement

The second lesson was structured to use a key Javascript concept to obtain a high score on the game by cheating. This engagement factor was so strong that I changed the material for the main Webmaker task of the lesson to include the cookie element. Students were asked to remix a page to create a calculation to provide the price of a Cookie Clicker T-shirt in Euros.



Cookie clicker T-Shirt for Sale

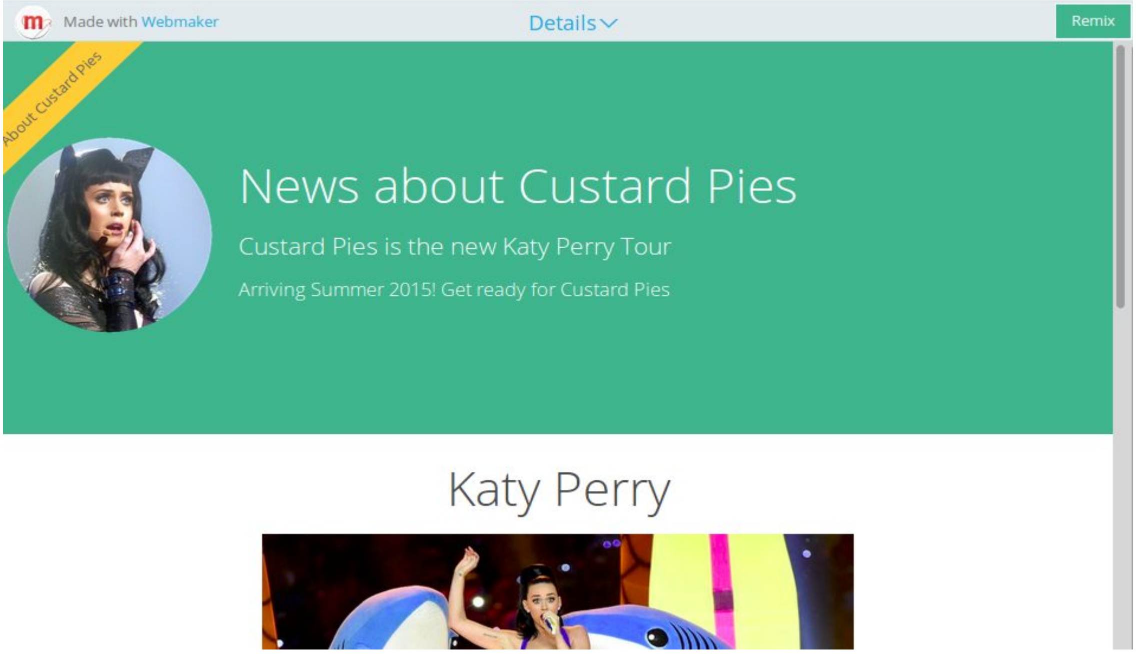
Uk Price: £7.5

Price for USA: $11.55

Price for Europe:

Illustration 23: Adapting activity subject matter to increase student engagement

Katy Perry, a singer whose live shows are diverse, colourful and a bit daft was used as a way to encourage creativity in creating names for a tour, as part of a Remix activity on variable values. Students who didn't like Katy Perry had a stronger incentive to remix the page and replace it with their own choice of music and bands, thus encouraging a different kind of creativity.



News about Custard Pies

Custard Pies is the new Katy Perry Tour

Arriving Summer 2015! Get ready for Custard Pies

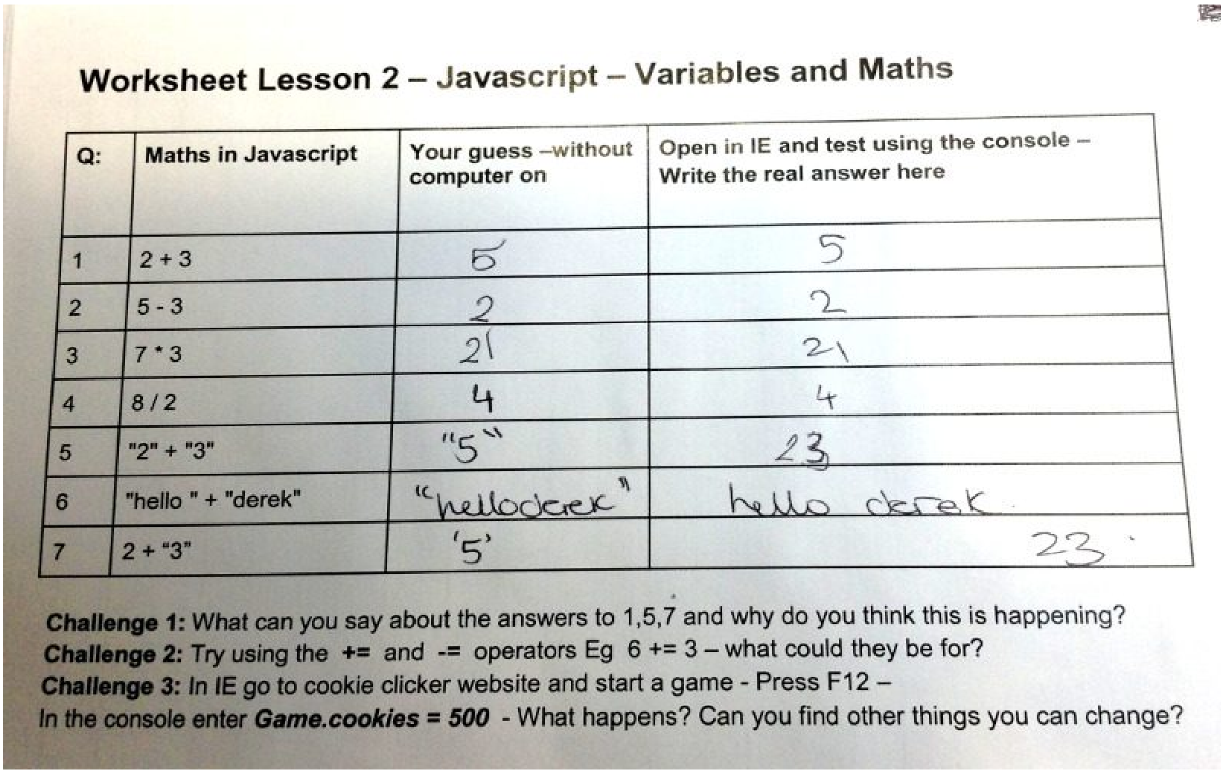
Katy Perry

Illustration 24: Remix Katy Perry's Tour Page Activity

There is a danger that in-jokes and culturally specific references included to increase learner engagement become a barrier to inclusion or distract from the true focus of the learning at hand. However, resources can be adapted to suit each group. In general, where attempted, my experience is that these techniques to increase student engagement were effective rather than a distraction. The students made comments, asked questions and showed greater focus after these elements had been introduced.

3.2.4 Instruction based activities with quick outcomes

While the goal of the unit is to build student autonomy by engaging with problem-based tasks, the use of quick associative activities to build knowledge and confidence are also vital.



Worksheet Lesson 2 – Javascript – Variables and Maths

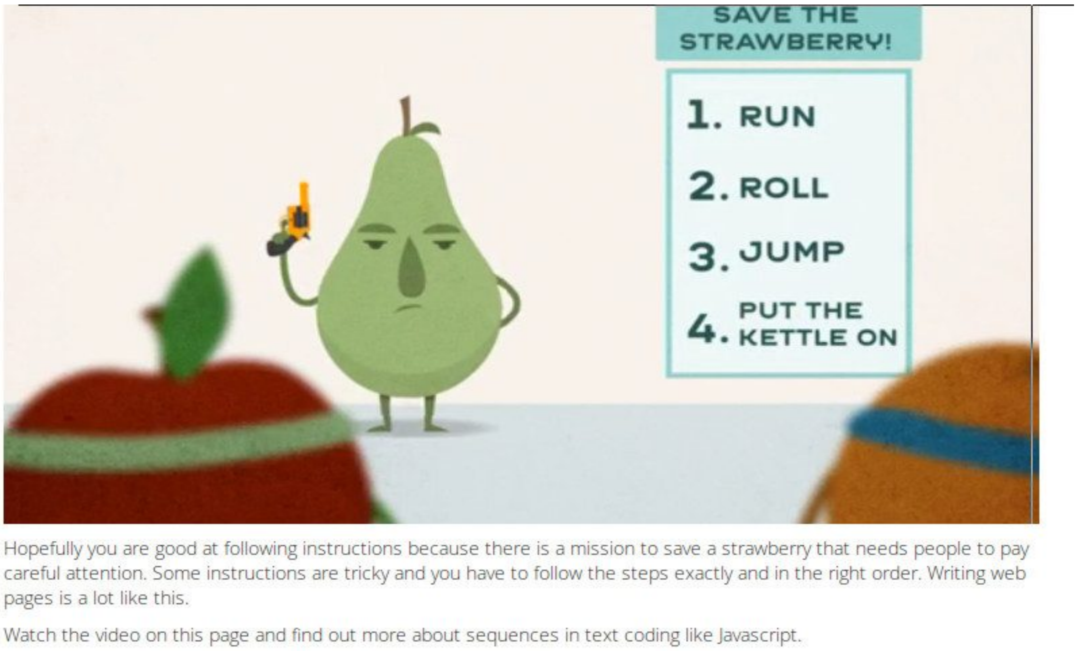
Q:	Maths in Javascript	Your guess –without computer on	Open in IE and test using the console – Write the real answer here
1	2 + 3	5	5
2	5 - 3	2	2
3	7 * 3	21	21
4	8 / 2	4	4
5	"2" + "3"	"5"	23
6	"hello " + "derek"	"hellodcrek"	hello derek
7	2 + "3"	'5'	23

Challenge 1: What can you say about the answers to 1,5,7 and why do you think this is happening?
Challenge 2: Try using the **+=** and **-=** operators Eg 6 += 3 – what could they be for?
Challenge 3: In IE go to cookie clicker website and start a game - Press F12 –
In the console enter ***Game.cookies = 500*** - What happens? Can you find other things you can change?

Illustration 25: Example of a quick instruction-based task with exploring elements

These activities helped with formative assessment and the rigours of providing data for school-based assessment. A key example of a successful task was 'Hacking Cookie Clicker' where students used the console of their internet browser to interact, to undertake a quick theory exercise and to then use this knowledge to 'hack' the score of a simple web based game. The sheet used to introduce this activity is shown in Illustration 25.

Positive student response led me to research and incorporate other associative but interactive activities which introduced, revised or supported more constructive activities.



Hopefully you are good at following instructions because there is a mission to save a strawberry that needs people to pay careful attention. Some instructions are tricky and you have to follow the steps exactly and in the right order. Writing web pages is a lot like this.

Watch the video on this page and find out more about sequences in text coding like Javascript.

Illustration 26: Incorporating BBC Bitesize videos and quizzes on sequencing, selection and iteration

Recent resources from BBC Bitesize on algorithms covering the sorting, selection and iteration featuring videos and quizzes have been incorporated into the curriculum as shown in Illustration 26. These resources use strong, engaging imagery of fruits with funny characters to illustrate concepts.

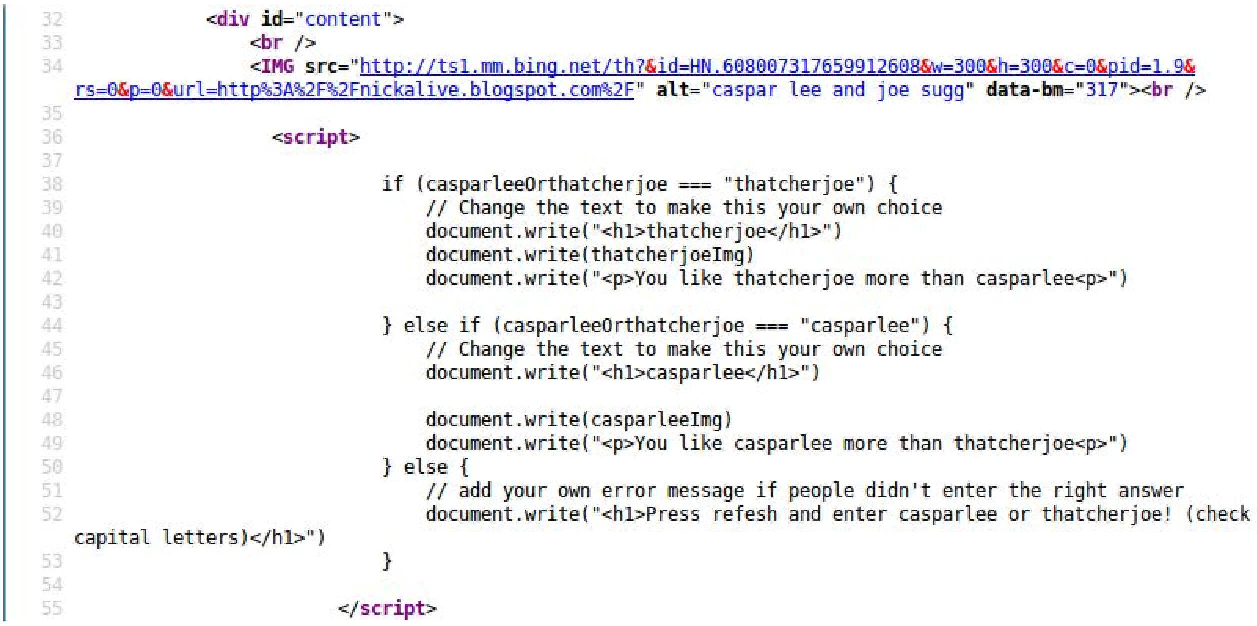
It is worth re-stating that these associative tasks do not counteract the aims of constructivist tasks. In effect, they are important for giving students key information and helping them to build confidence in their knowledge.

3.2.5 Reflections on the use of Dreamweaver

Because Dreamweaver had been used in a previous unit, a clear advantage was the familiarity students had with the tool. Disadvantages of the use of Dreamweaver included the potential complexities of the application and the use of the WYSWIG editor which divorces the user from the underlying code. The tendency is for users to use the GUI to do and undo formatting changes with no reference to the increasingly convoluted and contradictory HTML code being generated.

I attempted to side step this disadvantage by instructing students to use the split view of Dreamweaver, asking students to interact only with the code and using a Live view to preview changes.

The following code example shows where a student has used the WSYIWYG editor to insert an image outside of the selection code blocks.



```
            <div id="content">
                <br />
                <IMG src="http://ts1.mm.bing.net/th?&id=HN.608007317659912608&w=300&h=300&c=0&pid=1.9&
rs=0&p=0&url=http%3A%2F%2Fnickalive.blogspot.com%2F" alt="caspar lee and joe sugg" data-bm="317"><br />

                  <script>

                            if (casparleeOrthatcherjoe === "thatcherjoe") {
                                // Change the text to make this your own choice
                                document.write("<h1>thatcherjoe</h1>")
                                document.write(thatcherjoeImg)
                                document.write("<p>You like thatcherjoe more than casparlee<p>")

                            } else if (casparleeOrthatcherjoe === "casparlee") {
                                // Change the text to make this your own choice
                                document.write("<h1>casparlee</h1>")

                                document.write(casparleeImg)
                                document.write("<p>You like casparlee more than thatcherjoe<p>")
                            } else {
                                // add your own error message if people didn't enter the right answer
                                document.write("<h1>Press refesh and enter casparlee or thatcherjoe! (check
capital letters)</h1>")
                            }

                        </script>
```

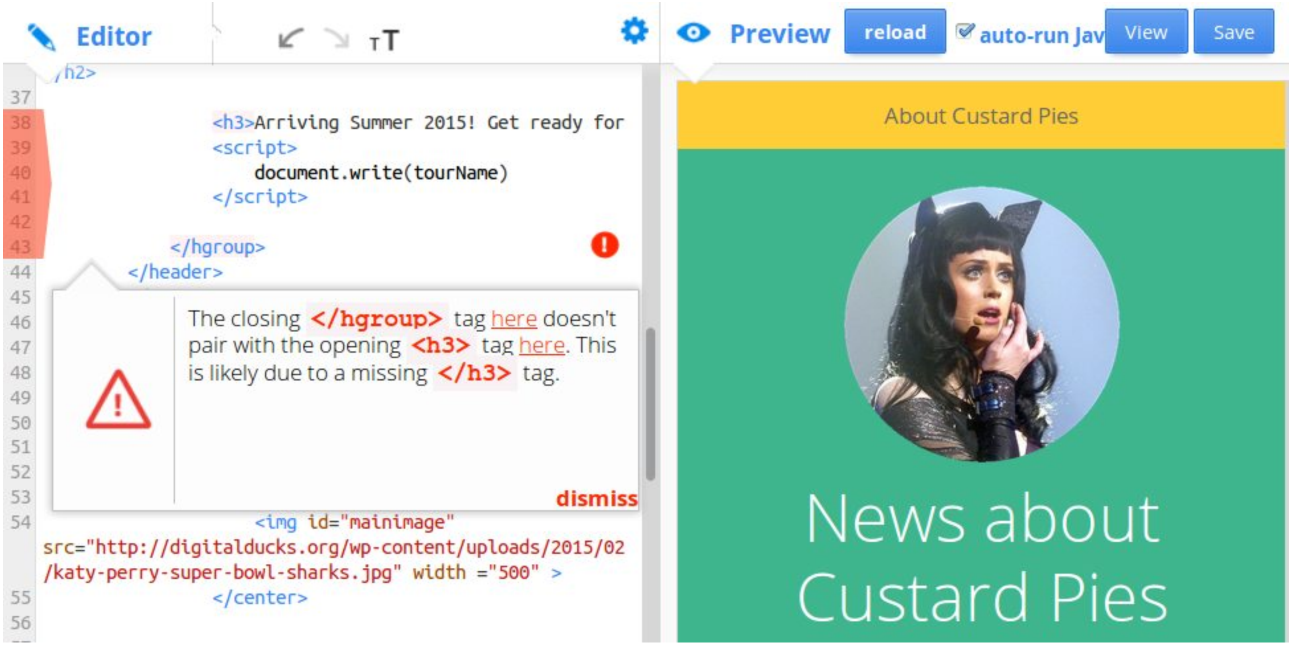
Illustration 27: Images added outside code blocks

While, this mistake showed good initiative and this example provided a way of revisiting the concept of code blocks during the presentation phase of the next lesson, it does show one of the pitfalls of using Dreamweaver in this situation.

3.2.6 Reflections on the use of Thimble

In school, the use of the tool was complicated hugely by poor internet connection during several lessons. If the teaching environment has a slow or unreliable network this tool may not be suitable.

The online editor Thimble was very effective in introducing students to the mechanics of web pages in a quick and engaging way. This may, in part, be due to its novelty and attractive layout. In addition, the lack of WSYIWYG keeps students focused on the code and the side by side code and preview window is clear.



```
/h2>

            <h3>Arriving Summer 2015! Get ready for
            <script>
                document.write(tourName)
            </script>

        </hgroup>
    </header>
```

The closing `</hgroup>` tag here doesn't pair with the opening `<h3>` tag here. This is likely due to a missing `</h3>` tag.

```
                <img id="mainimage"
src="http://digitalducks.org/wp-content/uploads/2015/02
/katy-perry-super-bowl-sharks.jpg" width ="500" >
            </center>
```

About Custard Pies

News about Custard Pies

Illustration 28: Thimble with side by side code and preview and code syntax help

The live updating of pages when Javascript variables are altered provides immediate feedback, thus visualising the effect of changing the values of variables. Syntax errors such as unclosed tags are highlighted and sometimes solutions are suggested as shown in Illustration 28.

Such technical scaffolding and the advantage of seeing the immediate effect of work done, fulfils some of the key characteristics of an effective, constructivist environment (Resnick and Silverman, 2005).

The designers of Thimble do seem to have thought carefully about what 'black boxes' are used to simplify the use of the tool. There is no automatic restriction of which HTML tags which can be used. However the tool does automatically update the preview window without having to click 'save'. It also simplifies the process of creating an account and publishing web pages when users do click on 'save'.

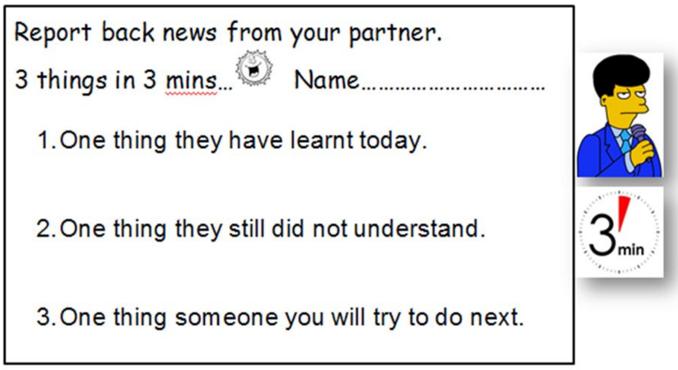
Let's look at the characteristics of 'low floor and wide walls' and 'a little coding goes a long way'. In these respects, Webmaking (with or without Thimble) holds up very well. A very limited amount of HTML and Javascript can open the door to almost limitless creative potential. The web has millions of images and memes to be explored and remixed. The online, public nature of the production provides advantages outlined in the next section.

Because of the open nature of the Thimble and the ability to input any text code, there is also massive potential to make mistakes. Many times learners were faced with blank pages and forced to check their pages for bugs. As such, the effective use of this tool goes hand in hand with a curriculum that introduces limited HTML tags and teaches their use effectively. On reflection, I would in future prepare a glossary of selected tags to learn and adapt suitable preparatory activities working with HTML before tackling Javascript.

There is also a large semantic jump needed to move from being able to create of web pages from scratch rather than altering them. This is was especially true with adding images and CSS formating. The tension between remix and original work has been explored in research by Hill and Monroy-Hernandez (2013).

3.2.7 The Potential of social activities and real life materials

Reporting activities, and other social activities were incorporated into the planning of the unit. An example is shown in Illustration 29.



Report back news from your partner.

3 things in 3 mins... Name...............................

1. One thing they have learnt today.
2. One thing they still did not understand.
3. One thing someone you will try to do next.

Illustration 29: Social reporting activity

On of the advantages of Thimble activities is that student work can be viewed online by other students and is therefore suitable for peer feedback. The design of and potential to feedback on public artefacts fits constructivist aims well (Kafai and Resnick, 1996; Papert, 1993, 1980; Resnick and Silverman, 2005). Public sharing was not used for school implementation due to the complications of students setting up user accounts. Mozilla plans to provide bulk creative of student accounts in the future.

Co-operative learning was also hampered in my classroom due partly to the set up of the room in to tables grouped by ability. Such grouping by ability should be done sparingly according to (Marzano et al., 2001, p. 88), as it may reduce the opportunities for co-operative learning and adversely effect the performance of lower ability students.

In terms of working towards a situative approach, tasks created for the curriculum were tailored to be relevant to the target group. As the lessons progressed tasks became progressively more geared to real-life situations. These included the use of web pages to promote a music event, to sell T-shirts and to advertise opening times.

It was not possible in the classroom implementation to make links with real organisations that would benefit from web design or promotion of events. However it is easy to see that there are many potential social and situative tasks that could be the goal of Webmaking assignments given different circumstances and more time arrange them.

This is an issue that is being addressed by the use of Mozilla Webmaker clubs. One of their goals is to link up existing small learning clubs with real life communication tasks, providing a variety of ways to interact (Mozilla Foundation, 2015b).

3.3 Recommendations emerging from classroom implementation and areas for professional development

3.3.1 A flexible toolkit of engaging activities and formative assessment techniques

A choice of adaptable activities allows the material created for this curriculum to be utilised beyond the confines of schools. Teachers and facilitators can pick and mix tasks based on their knowledge of their students. There is plenty of scope for improvement. Because most activities are published as Thimble activities, this encourages others to adapt them by clicking on 'remix'. A forum for teachers and students encourages such remixes to be fed back into the course.

While this study focused on the creation of simple memes, other webmaking tools and activities exist which de-construct and apply digital story telling techniques. This creates a lot of potential for cross curriculum working and advanced digital literacy skills.

During the delivery of this curriculum I used a variety of different formative assessment techniques. These included RAG activities, thumbs up, learning triangles, differentiated questioning and peer assessment. Such an approach is key to addressing misunderstandings effectively and facilitating informed differentiation.

I would like to do more research into this area, build up a flexible personal toolkit of formative assessment techniques and become a confident practitioner of them. While every teacher has their own techniques, where appropriate I would also integrate suggestions for formative assessment into this online curriculum.

The limits of network connection in this placement created a conservative approach to the use of online tools. However, there are tools which would be easy to integrate. For example, a Socrative quiz for each lesson could be a useful aide for many teachers.

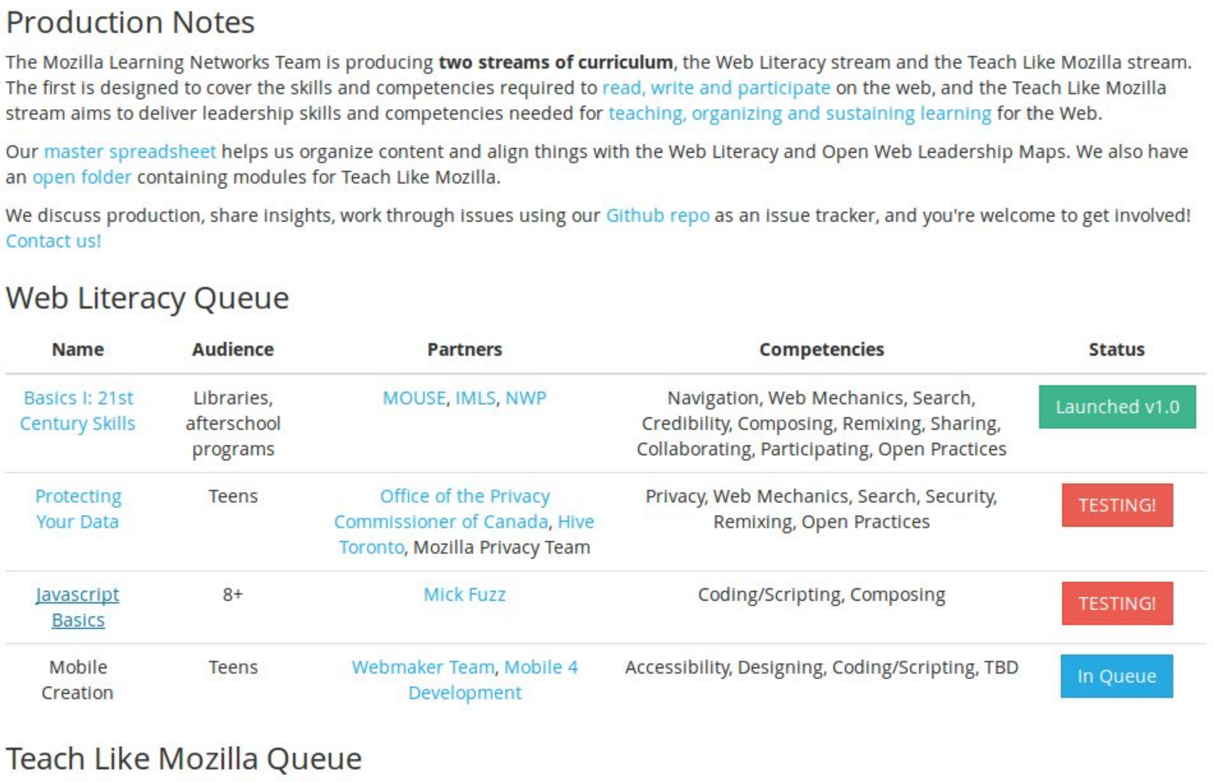
3.3.2 Ideas for extending the curriculum

During the course of this project, I have thought of areas to extend or create a follow up course. The use of Boolean Logic was not covered in this course. It could be integrated with an extension activity or a lesson following on from selection.

Using Canvas to create Javascript animations or games is also a very interesting area which could increase engagement. It has huge potential to link to the students' experiences of playing online games (Chris Mills, 2014). This may be beyond the remit of KS3 but more appropriate to KS4 or 5.

While there are some conceptual complexities to working with existing HTML elements and the DOM to manipulate page elements, this is key knowledge for real web development. As such, this represents an appropriate next step to tackle in an accessible way.

One positive step for my professional development is that this course has been accepted into the testing phase of Mozilla's Webmaker curriculum as evidenced in Illustration 30 (Mozilla Foundation, 2015c).



Production Notes

The Mozilla Learning Networks Team is producing **two streams of curriculum**, the Web Literacy stream and the Teach Like Mozilla stream. The first is designed to cover the skills and competencies required to read, write and participate on the web, and the Teach Like Mozilla stream aims to deliver leadership skills and competencies needed for teaching, organizing and sustaining learning for the Web.

Our master spreadsheet helps us organize content and align things with the Web Literacy and Open Web Leadership Maps. We also have an open folder containing modules for Teach Like Mozilla.

We discuss production, share insights, work through issues using our Github repo as an issue tracker, and you're welcome to get involved! Contact us!

Web Literacy Queue

Name	Audience	Partners	Competencies	Status
Basics I: 21st Century Skills	Libraries, afterschool programs	MOUSE, IMLS, NWP	Navigation, Web Mechanics, Search, Credibility, Composing, Remixing, Sharing, Collaborating, Participating, Open Practices	Launched v1.0
Protecting Your Data	Teens	Office of the Privacy Commissioner of Canada, Hive Toronto, Mozilla Privacy Team	Privacy, Web Mechanics, Search, Security, Remixing, Open Practices	TESTING!
Javascript Basics	8+	Mick Fuzz	Coding/Scripting, Composing	TESTING!
Mobile Creation	Teens	Webmaker Team, Mobile 4 Development	Accessibility, Designing, Coding/Scripting, TBD	In Queue

Teach Like Mozilla Queue

Illustration 30: Screenshot of Mozilla curriculum from - http://mozilla.github.io/webmaker-curriculum/

4 Conclusion

The scope of this study has obvious limits. Only six lessons were delivered in a school environment. Network issues hindered the use of online tools and the student group had low investment in the subject area. Initial aims to use Webmaking tools to solve open ended challenges were modified to give greater structure for most activities.

This change of approach of classroom implementation was partly necessitated by the short length of time of the course. In future, I would allocate at least 20 hours, including time spent learning HTML to provide a more suitable pace to set up the constructivist tasks as outlined in the activities in the learning resources in Appendix 1.

In spite of this, the work collected, progress shown and informal observation of student engagement has lead me to a conclusion that the Webmaking approach has significant potential to teach elements of the KS3 computing curriculum. While this study only engaged with the limited use of social activities and sharing of created work, I feel that this experience shows that Thimble and Webmaker activities are suitable engaging, constructivist, learning environments.

5 References

Beaumont, C., O’Doherty, M., Shannon, L., 2011. Reconceptualising assessment feedback: a key to improving student learning? Stud. High. Educ. 36, 671–687. doi:10.1080/03075071003731135

Beere, J., Gilbert, I., 2012. The perfect Ofsted lesson. Independent Thinking Press, Bancyfelin, Carmarthen, Wales.

Beetham, H. (Ed.), 2013. Rethinking pedagogy for a digital age: designing for 21st century learning, Second edition. ed. Routledge, New York.

Black, P., Wiliam, D., 1998. Assessment and Classroom Learning. Assess. Educ. Princ. Policy Pract. 5, 7–74. doi:10.1080/0969595980050102

Boekaerts, M., Corno, L., 2005. Self-regulation in the classroom: A perspective on assessment and intervention. Appl. Psychol. 54, 199–231.

Chris Mills, 2014. Introduction to programming, Part 5 [WWW Document]. URL http://people.mozilla.org/~cmills/coding-beginners/lesson5.html (accessed 5.10.15).

Computing at School, 2010. Computing at School white paper. [WWW Document]. URL http://www.computingatschool.org.uk/data/uploads/Computing_at_School.pdf (accessed 10.12.14).

Computing at School Working Group, 2012. A curriculum framework for Computer Science and Information Technology [WWW Document]. URL http://www.computingatschool.org.uk/data/uploads/Curriculum%20Framework%20for%20CS%20and%20IT.pdf (accessed 12.18.14).

Department for Education, 2015. Subject content and assessment objectives for GCSE in computer science for teaching from 2016. [WWW Document]. URL https://www.gov.uk/government/publications/gcse-computer-science (accessed 5.6.15).

Fields, D.A., Giang, M., Kafai, Y.B., 2013. Understanding collaborative practices in the Scratch online community: Patterns of participation among youth designers, in: To See the World and a Grain of Sand: Learning across Levels of Space, Time, and Scale: CSCL 2013 Conference Proceedings. International Society of the Learning Sciences, pp. 200–207.

Gilby, N., Great Britain, Department for Education and Skills, 2008. National survey of parents and children: family life, aspirations and engagement with learning, 2008. Dept. for Children, Schools and Families, [London].

Greeno, J.G., Eckert, P., Stucky, S.U., Sachs, P., Wenger, E., 1999. Learning in and for participation in work and society. Wash. DC US Dep. Educ. Retrieved April 21, 2010.

Hill, B.M., Monroy-Hernandez, A., 2013. The Remixing Dilemma: The Trade-Off Between Generativity and Originality. Am. Behav. Sci. 57, 643–663. doi:10.1177/0002764212469359

Illich, I., 1971. Deschooling society, 1st ed. ed, World perspectives, v. 44. Harper & Row, New York.

Jonassen, D.H., 1997. Instructional design models for well-structured and III-structured problem-solving learning outcomes. Educ. Technol. Res. Dev. 45, 65–94.

Kafai, Y.B., 1995. Minds in play: computer game design as a context for children’s learning. L. Erlbaum Associates, Hillsdale, N.J.

Kafai, Y.B., Resnick, M. (Eds.), 1996. Constructionism in practice: designing, thinking, and learning in a digital world. Lawrence Erlbaum Associates, Mahwah, N.J.

Marzano, R.J., Pickering, D., Pollock, J.E., 2001. Classroom instruction that works: Research-based strategies for increasing student achievement. Ascd.

Mayes, T., De Freitas, S. (Eds.), 2013. Rethinking pedagogy for a digital age: designing for 21st century learning, Second edition. ed. Routledge, New York.

Mozilla Foundation, 2015a. The Mozilla Manifesto [WWW Document]. URL https://www.mozilla.org/en-US/about/manifesto/ (accessed 5.9.15).

Mozilla Foundation, 2015b. Webmaker/2015/Mentors/Clubs [WWW Document]. URL https://wiki.mozilla.org/Webmaker/2015/Mentors/Clubs#Local_groups_globally_networked. (accessed 5.6.15).

Mozilla Foundation, 2015c. Mozilla Learning Curriculum [WWW Document]. URL http://mozilla.github.io/webmaker-curriculum/ (accessed 5.10.15).

Mozilla Foundation, 2014. Webmaker Whitepaper [WWW Document]. URL https://wiki.mozilla.org/Webmaker/Whitepaper (accessed 5.9.15).

Oftsed, 2011. The impact of the "Assessing pupils" progress' initiative [WWW Document]. URL https://www.gov.uk/government/uploads/system/uploads/attachment_data/file/379101/The_20impact_20of_20the_20Assessing_20pupils_20progress_20initiative.pdf (accessed 5.10.15).

Papert, S., 1993. The children's machine: rethinking school in the age of the computer. BasicBooks, New York.

Papert, S., 1980. Mindstorms: children, computers, and powerful ideas, 2nd ed. ed. Basic Books, New York.

Piaget, J., 1952. The origins of intelligence in children. W. W. Norton, New York.

Price, D., 2013. Open: how we'll work, live and learn in the future. Crux, [England].

Resnick, M., 2012. Mother's Day, Warrior Cats, and Digital Fluency: Stories from the Scratch Online Community, in: Proceedings of the Constructionism 2012 Conference. Athens, Greece.

Resnick, M., Silverman, B., 2005. Some reflections on designing construction kits for kids, in: Proceedings of the 2005 Conference on Interaction Design and Children. ACM, pp. 117–122.

Ross, A., 2009. Disengagement from Education among 14-16 year olds [WWW Document]. URL http://dera.ioe.ac.uk/10620/1/dcsf-rr178.pdf (accessed 3.17.15).

Rushkoff, D., 2010. Program or be programmed: ten commands for a digital age. OR Books, New York.

The Royal Society, 2012. Shut down or restart? The way forward for computing in UK schools. [WWW Document]. URL https://royalsociety.org/~/media/education/computing-in-schools/2012-01-12-computing-in-schools.pdf (accessed 10.12.14).

Vossoughi, S., Bevan, B., 2015. Making and Tinkering: A Review of the Literature.

Wiliam, D., 2011. Embedded formative assessment. Solution Tree Press, Bloomington, IN.

Wiliam, D., 2006. Assessment for Learning: why, what and how [WWW Document]. URL http://www.dylanwiliam.org/Dylan_Wiliams_website/Papers_files/Cambridge%20AfL%20keynote.doc (accessed 5.10.15).

Young, E., 2005. Assessment for Learning: Embedding and extending. Retrieved May 20, 2008.