



HAL
open science

“Literacy from Python”

Lawrence Williams, Beth Mead

► **To cite this version:**

Lawrence Williams, Beth Mead. “Literacy from Python”. Open Conference on Computers in Education (OCCE), Aug 2021, Tampere, Finland. pp.41-53, 10.1007/978-3-030-97986-7_4 . hal-04161452

HAL Id: hal-04161452

<https://inria.hal.science/hal-04161452v1>

Submitted on 1 Aug 2023

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L’archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d’enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



Distributed under a Creative Commons Attribution 4.0 International License



This document is the original author manuscript of a paper submitted to an IFIP conference proceedings or other IFIP publication by Springer Nature. As such, there may be some differences in the official published version of the paper. Such differences, if any, are usually due to reformatting during preparation for publication or minor corrections made by the author(s) during final proofreading of the publication manuscript.

“Literacy from Python” Using Python for a Proposed Cross-curricular Teaching and Learning Model

Lawrence Williams¹ and Beth Mead²

¹ TPEA Specialist Leader in EdTech Education, UK

² Safeguarded school student, Wimbledon, London, UK

lawrencewilliams2060@gmail.com

Abstract. This pilot case study, which outlines an introductory computer programming project, using Python, builds on the international success of the “Literacy from Scratch” teaching model [1-3]. Its aim is to contextualise the teaching of Python through a cross-curricular teaching and learning model, rather than through teaching Python elements separately. It therefore recommends the development of elementary coding skills by pupils, aged from 10 to 14 years, through the creation of very short stories or scripts, using Python strings. Each story chapter of the project further introduces worked examples of additional Python programming elements such as Turtle graphics and times-tables, within the constructionist framework of a creative writing project [4]. The project is supported by a free website for teachers. Our hope is that this creative approach to coding may encourage the enthusiastic adoption of such a cross-curricular method for teaching and learning computer coding with Python in our schools.

Key words: Python, Scratch, story-telling, cross-curricular teaching and learning, creativity, constructionism, contextualisation

1 Introduction

Teachers in England state schools are required to teach two computer programming languages to their students, from age 5 years. One of these languages can be a visual or “block coding” program, such as Scratch, but the other must be text-based, such as Python. In practice, the programs most commonly used in England’s schools, because they are free, are Scratch, frequently for pupils aged 5 to 11 years, followed by Python for ages 11 years upwards. Some schools purchase computer programming packages because they come with prepared lesson plans, and teaching strategies. To support the development of Scratch in our UK schools, Brunel University London, and Charles University in Prague, Czech Republic, with Torino University, Italy, developed the free website, “Literacy from Scratch”. Devised by Lawrence Williams (at Brunel’s Education Department), and quickly shared with other teacher training teams abroad, it

proved to be very popular, and an accompanying text book was subsequently commissioned, and published, by Routledge [5]. The learning model for Python, presented here, is founded on the same constructionist and psychological principles [6].

My previous work in “Literacy from Scratch” at Brunel’s partner schools in west London, resulted in pupil projects being fully completed by an entire Year 8 group (160 students, aged 12–13 years) at Bishop Ramsey School, and later by a Year 1 class (aged 5 years) and a Year 2 class (aged 6 years) at the Swaminarayan Primary School. All of the pupils involved met all the assessment criteria for the project, including story-telling, use of dialogue, animation, and simple art work for creating backgrounds and story characters. The success of this earlier project rested partly on the contextualisation of the learning of coding: secondary school pupils wanted to tell their stories, and the only way to do this was through using accurate Scratch coding, a powerful incentive to learning. At Primary school level (5 years to 11), the teachers already knew how to develop creative writing skills, and, once again, Scratch became the new means by which to deliver creative and animated narrative work. Contextualised learning, in our view, clearly helps students to make connections between the content of their studies and the context in which it is developed. Vygotsky [6] refers to this as the “gap between what is known and what is being learned”. Students draw from their previous experience (in this case, of writing narratives, creating characters and dialogue) to give new meaning to what they are now learning i.e. Scratch and Python computer coding [7].

The second requirement of using a text-based computer language, however, is more difficult for UK teachers, currently still largely untrained in programming. Following the classroom success of the “Literacy from Scratch” teaching and learning project, we (the paper authors) are starting to develop “Literacy from Python”, based on this earlier curriculum model. It aims to support the following elements of the Programmes of Study, Key Stage 3, England National Curriculum¹:

“Pupils should be taught to:

- “use 2 or more programming languages, **at least one of which is textual, to solve a variety of computational problems; make appropriate use of data structures [for example, lists, tables or arrays];** design and develop modular programs that use procedures or functions
- “understand how instructions are stored and executed within a computer system; **understand how data of various types (including text, sounds and pictures) can be represented and manipulated digitally, in the form of binary digits**
- “**undertake creative projects that involve selecting, using, and combining multiple applications, preferably across a range of devices, to achieve challenging goals, including collecting and analysing data and meeting the needs of known users**
- “**create, reuse, revise and repurpose digital artefacts for a given audience, with attention to trustworthiness, design and usability**”².

¹ Statutory Guidance, announced in February 2012, National Curriculum in England, Computing Programmes of Study, Key Stage 3, for implementation (2013)

² The UK National Curriculum in Computing elements, highlighted in bold, are fulfilled by the “Literacy from Python” project.

The first steps taken (called Chapter 1 on the website) were originally, however, simply an experimental exploration of what might be done to contextualise the learning of Python, as we had done earlier for Scratch. Once the model had been conceptualised by the paper’s authors as a possible cross-curricular working model, however, Beth (Beth Mead, a primary school pupil, then aged 11 years) and I set about developing further narrative scripts. The model was developed based specifically on the “Literacy from Scratch” model, and all of the Python coding was written by Beth. Largely self-taught, she used coding books [8], and online resources, such as the modified Turtle graphics example given in Chapter 3, shown later. Each Chapter added more Python elements as part of Beth’s narrative scripts, and we created a website to share these thoughts (and the coding elements) with teachers. This thinking crystallised into the following set of project aims for the “Literacy from Python” project:

- To develop a cross-curricular project including simple story-writing skills and computer coding
- To develop elementary computer programming skills, using Python elements, such as strings and lists
- To provide support for UK Key Stage Computing courses (for ages 11 to 14 years) and GCSE Computer Science (for ages 15 to 16 years), and possibly for postgraduate teacher training courses
- To develop contextualised learning as a framework for teaching Python
- To create an educational support site, where teachers can download suitable, free classroom resources based on this teaching and learning model

The “Literacy from Python” project evolved, then, from an experiment with storytelling, which started in early 2018. Beth wrote the first short story using Python strings, set in a classroom, and involving her real-life pet dog, Maisy, and her Aunt Justine’s cat, Malika. This project was itself the development of a series of short stories, “Stories for Children 2” that she had already written and published on the web, using Microsoft (MS) PowerPoint, to exemplify some basic computer coding concepts such as Algorithms, Abstraction, Decomposition, Generalisation, Logical Thinking, and If, Then, Else. These MS PowerPoint coding stories are posted on the “World Ecitizens” website as a free resource for teachers. There was then a delay, throughout 2018, while Beth and I worked together on some new projects using Scratch 2 and then Scratch 3, incorporating Sibelius music files with Lloyd Mead, at Lambeth College [9]. We returned to look at Python in 2019, to develop this storytelling idea further. Frustratingly, while positive discussions are currently underway with local schools and colleges, the classroom development of this work has, owing to severely restricted access to schools, been slowed by Covid³.

While there is a useful (single) introductory lesson on the MIT Scratch website, showing how to download Python, there is, as yet, no material for developing storytelling using Python in the cross-curricular way suggested in this paper. The Python

³ This teaching model was first accepted for presentation at ICET 2020 (Vancouver), but was later withdrawn because of health concerns caused by the Covid pandemic. Presentation of the paper was postponed, and later delivered by remote video link, at OCCE-IFIP 2021, for the TPEA.

websites have methods for generating storytelling with Pandas ('The library called **pandas**' is a Python package that provides data structures and is useful for structured and time-series data), but, once again, this is quite different from the creative, cross-curricular, constructionist, contextualised method of teaching and learning that we are advocating.

The accompanying Python website⁴, developed to support this curriculum initiative, outlines several Python coding skills developed through creative narrative work, divided into very short Chapters or scripts. More elements are planned (see Figure 10).

2 Method

In English lessons, or as an after-school homework, students aged 10 years and upward, will (post-Covid) be encouraged to create characters, and write a very short story which includes some dialogue - the "literacy" aspect of the project. This story can then be taken to the computer room, where, using Python, it can be reworked in the IDLE programming window, and printed out in the Shell window as an elementary Python program. This model therefore adds creative and cross-curricular elements to the computer coding process. In Figure 1, the programming example can be seen, faintly, on the right of the illustration, with the instruction to print (in pink in Python – and on the website), with blank lines added as brackets () and the "table" imported, so that when the story is saved and "Run" as a program, the reader can choose a times-table which prints out within the story (in blue text in Python) in the left, shell, window. The "Literacy from Python" website provides a clearer, more colourful view.

The image shows two windows from a Python IDE. The left window, titled 'Python 3.8.3 Shell', displays a story about a character named Maisy who is learning about Python. The story includes dialogue and actions, such as Maisy asking questions and Malika explaining concepts. The right window, titled 'Python 3.8.3 Shell', shows the corresponding Python code. The code uses `print()` statements to output the story text. A function named `times_table` is defined, which takes a number `n` as input and prints a multiplication table for `n` from 1 to 12. The code also includes a `main` function that prompts the user for a number and calls the `times_table` function.

```

Python 3.8.3 Shell
File Edit Shell Debug Options Window Help
Maisy LEARNS TO CODE WITH 'PYTHON'

(Oo, how to use 'Python' for story-telling)

Maisy, the Spanish, was walking past the school.

Hearing the word 'Python', she wondered why the pupils were talking about snakes.

She also heard the words 'Times Tables', and wanted to know more.

So, she went to see her wise friend Malika, the cat.

Malika explained: "It's not a python, it's 'Python', a computer program. It is named after a famous TV comedy show!"

Maisy wanted to learn about Python, so Malika showed her how to download the free computer program into her computer.

Maisy noticed that different words were different colours depending on what word a user used.

Maisy went back to Malika and asked what had happened.

Malika told her that each colour meant something different: 'print' is pink/purple because it is a function using its brackets '()' and inverted comma'' are green because they are strings.

Even though Malika is very wise, she didn't know what a function was.

Maisy knew that a function is a task that the computer carries out, such as print, or run, and shared her knowledge with Malika.

Maisy was playing about and found that Python could also calculate numbers. Maisy remembered that the pupils were talking about Times Tables.

She went back to Malika to ask if Python could do Times Tables. Malika showed Maisy how to make them.

The 7 times table is one of the hardest. Let's try to make the 7 times table, and see what happens! Put the number 7 in and press Enter...?

1 x 7 = 7
2 x 7 = 14
3 x 7 = 21
4 x 7 = 28
5 x 7 = 35
6 x 7 = 42
7 x 7 = 49
8 x 7 = 56
9 x 7 = 63

Python 3.8.3 Shell
File Edit Shell Debug Options Window Help
#Programmer and story: Beth Wood
#Date: Monday 28th February 2018, 15:58

print("Maisy LEARNS TO CODE WITH 'PYTHON'")
print()
print("(Oo, how to use 'Python' for story-telling)")
print()
print("Maisy, the Spanish, was walking past the school.")
print()
print("Hearing the word 'Python', she wondered why the pupils were talking about snakes.")
print("She also heard the words 'Times Tables', and wanted to know more.")
print("So, she went to see her wise friend Malika, the cat.")
print("Malika explained: 'It's not a python, it's 'Python', a computer program.")
print("Maisy wanted to learn about Python, so Malika showed her how to download the free computer program into her computer.")
print("Maisy noticed that different words were different colours depending on what word a user used.")
print("Maisy went back to Malika and asked what had happened.")
print("Malika told her that each colour meant something different: 'print' is pink/purple because it is a function using its brackets '()' and inverted comma'' are green because they are strings.")
print("Even though Malika is very wise, she didn't know what a function was.")
print("Maisy knew that a function is a task that the computer carries out, such as print, or run, and shared her knowledge with Malika.")
print()
print("Maisy was playing about and found that Python could also calculate numbers.")
print("Maisy remembered that the pupils were talking about Times Tables. Malika showed Maisy how to make them.")

table = int(input("The 7 times table is one of the hardest. Let's try to make a 7 times table. Put the number 7 in and press Enter...?"))
print("M", "x", table, "=", x*table)

print()
print("Maisy wondered why you put 18 for 7 x 12. Malika didn't know, but said 'I'll try to help you.'")
print("Maisy must wonder if Python could help her with a guessing game.")
print("Can you guess how many pupils there are in the class?" asked Maisy.)
print("I will guess there are 28 pupils.")
print()
print("28? No. A couple more than that.")
print()
print("29? No. A couple more than that.")
print()
print("Maisy said, 'How I know what the pupils were talking about when I passed...')
print("Maisy said, 'How I know what the pupils were talking about when I passed...')

```

Fig.1. Simple Python programming for storytelling and times tables

⁴ The websites referenced in this paper can all be found using Google searches.

This process therefore combines storytelling, simple arithmetic (addition and times-tables), as well as computer programming. It is explained in fuller detail on the “Literacy from Python” website, together with the downloadable Python story files, for students and teachers to explore and develop in their own classrooms.



Fig. 2. Maisy, a real-life Spaniel

Maisy (Figure 2) is one of the characters in the Python stories. She also appears in the 2017 storybook series that accompanies this ongoing project. Each story, originally in MS PowerPoint format, and free for use in the classroom, looks at a different element of computational thinking (“World Ecitizens” website). Some of these concepts are detailed below.

Computational Thinking

There is no single, agreed definition of computational thinking. It depends upon whom you ask - programmer, logician, or mathematician. There are, however, several generally agreed elements, and these are explored in these stories for children. For information on the choice of computational thinking concepts, these are shown in the resources developed by Computing at School, and include:

- Algorithmic thinking (following a set of rules; sequencing skills)
- Debugging (detecting and correcting errors)
- Decomposition (breaking a complex problem down into simpler parts)
- Logical thinking (reasoning skills)
- Generalisation (recognising patterns, and using them to find solutions)
- Abstraction (sorting relevant information from unnecessary detail)

These concepts [8] are presented individually through these Maisy stories, and can be shown to pupils on an interactive whiteboard, or printed out for use in the classroom (see Figure 3). They can then be discussed in class and incorporated into the narrative work before embarking on Python coding.

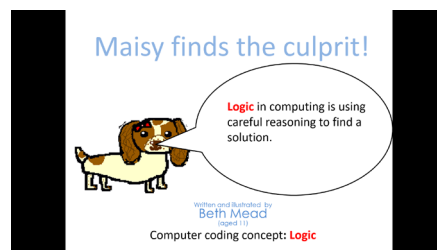


Fig. 3. A sample title page (one of six stories) exemplifying a coding concept

This teaching method has inspired us (Beth and myself) to develop the idea further, in addition to the MS PowerPoint method, above, incorporating more and more Python programming elements, as well as developing the storybook characters along the way [7]. “Literacy from Python” can be seen, therefore, as the latest development of a series of storytelling projects, which can be found set out in historical sequence, on the “World Ecitizens” web site (see the homepage and scroll down: <http://www.worldcitizens.net/>).

The five-chapter story series, together with the Python elements is given below.

Chapter 1: Addition and Times Tables

Python Elements: Text Entry plus Importation of Times-tables

Using Python 3.6.5 originally (see the small text in the top bar of the Figure 4 file), Beth created a children’s story which introduced her animal characters, Maisy and Malika, to Addition and Time Tables within an imaginary classroom scenario. This teaching and learning process is further explained on the website, with practical hints for teachers (see Introduction and Teachers’ Notes).

It is a creative, cross-curricular project that so far involves:

- English (writing short stories, including very careful punctuation)
- Mathematics (addition, times-tables, geometry – and we hope to add graphs later)
- Geography
- Art
- Computing (using Python coding, as a second coding language after Scratch)
- ICT (file management as part of computing)

Beth used the print text facility in Python to make the following short story chapters. It shows how Python sub-routines (a table, table = int) can be imported and used within a story-telling context. The text appears within the brackets in green, and “prints out” as blue:

```
table = int(input("The 7 times table is one of the hardest. Let's try to make the 7 times table, and see what happens! Put the number 7 in and press Enter..."))
```

```
for x in range(1,13):
```

```
    print(x, "x", table, "=", x*table)
```

This is found to be much easier than it looks.

So, she wrote the following story, called “Finally” on the website (Finally.py) while still in primary school, Year 6, aged 11 years. This later became “Chapter 1” of the current larger computer coding project, “Literacy from Python”, i.e. writing short stories created by using Python.


```

Finally.py - C:\Users\lowner\AppData\Local\Programs\Python\Python36-32\Finally.py (3.6.5)
File Edit Format Run Options Window Help
#Programmer and story: Beth Mead
#Date: Monday 26th February 2018, 15:56

print("MAISY LEARNS TO CODE WITH 'PYTHON'")
print()
print("Or, how to use 'Python' for story-telling.")
print()
print("Maisy, the Spaniel, was walking past the school.")
print()
print("Hearing the word 'Python', she wondered why the pupils were talking about snakes!")
print("She also heard the words 'Times Tables', and wanted to know more.")
print("So, she went to see her wise friend Malika, the cat.")
print("Malika explained: 'It's not a python, it's 'Python', a computer program. It is named after a famous TV comedy show'")
print("Maisy wanted to learn about Python, so Malika showed her how to download the free computer program into her computer.")
print("Maisy noticed that different words went different colours depending on what words were used.")
print("Maisy went back to Malika and asked what had happened.")
print("Malika told her that each colour meant something different: 'print' is pinky/purple because it is a function; words in brackets '()' and inverted commas '' are gr
print("Even though Malika is very wise, she didn't know what a function was.")
print("Maisy knew that a function is a task that the computer carries out, such as print, or run, and shared her knowledge with Malika.")
print()
print("Maisy was playing about and found that Python could also calculate numbers. Maisy remembered that the pupils were talking about Times Tables.")
print("She went back to Malika to ask if Python could do Times Tables. Malika showed Maisy how to make them.")
print()
table = int(input("The 7 times table is one of the hardest. Let's try to make the 7 times table, and see what happens! Put the number 7 in and press Enter..."))
for x in range(1,13):
    print(x, "x", table, "=", x*table)
print()
print("Maisy wondered why you put 13 for _ x 12. Malika didn't know, but said she would find out later. A sort of traffic light before a road junction")
print()
print("Maisy next wondered if Python could help her with a guessing game.")
print("Can you guess how many pupils there are in the class?" asked Maisy.)
print("I will guess there are 28 pupils.")
print()
print("28? No. A couple more than this.")
print()
print(28 + 2)
print()
print("Well done!")
print()
print("Maisy said, 'Now I know what the pupils were talking about when I passed the school. Python is a really useful snake! I mean, computer program!'")

```

Fig. 4. With table included

Chapter 2: A Geometry lesson

Python Elements: Text Entry plus Turtle Module

This Chapter looks at Python for making geometrical shapes. (A year on from Chapter 1, we are now using version 3.7.2.) In the story, the animal characters learn to make a circle and a square, by importing turtle graphics.

Here is an excerpt from the story:

print("Malika said, "My friend Maisy and I have been working on Python.
We have been learning how to draw geometrical shapes.

Maisy, would you like to remind everyone how these look? How about a
circle and a square?")

```

print()
import turtle
window = turtle.Screen()
turtle.bgcolor("mintcream") #background colour of the window
turtle.color("olive") #stroke colour
turtle.begin_fill()
turtle.circle(80)
turtle.end_fill
from turtle import*
color("mintcream")
forward(50)
color("skyblue")
shape("turtle")
pensize(4)

```

```

forward(50)
right(90)
forward(50)
right(90)
forward(50)
right(90)
forward(50)
right(90)

```

The result of the above coding:

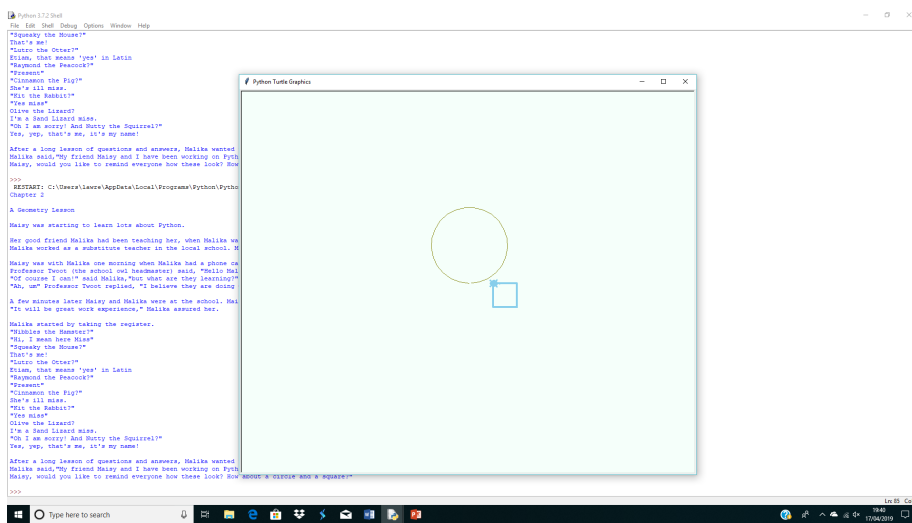


Fig. 5. Drawing a circle and square in Python, with the story and dialogue on the left

Chapter 3: A Biology Lesson

Python Elements: Text Entry plus Turtle Graphics

This Chapter was demanding. It is a biology lesson, with graphics. The story has a decorative flower at the end, thanks to inspiration from Aibek Toroev (a software engineer from Kyrgyzstan). There is a useful YouTube video created by Aibek Toroev, and with his coding later adapted by Beth, at: <https://www.youtube.com/watch?v=ZsnKkyRCWR4>.

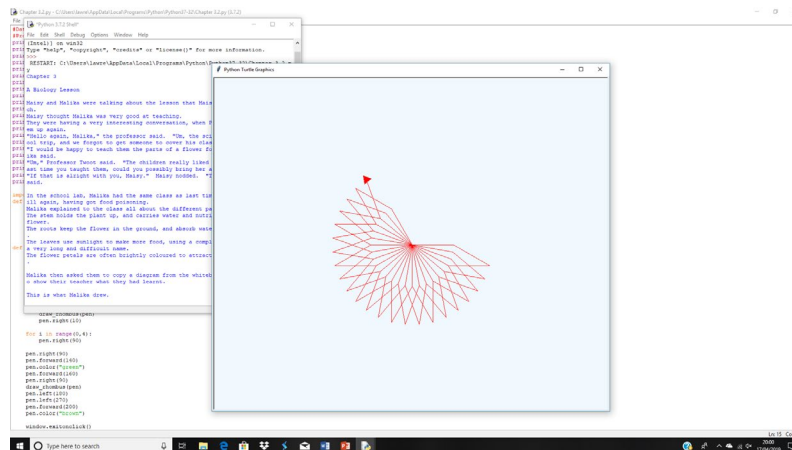


Fig. 6. The flower in progress (see the website file for the finished bloom)

As noted above, the England National Curriculum in Computing requires pupils to “create, re-use, revise and re-purpose” digital information, and Figure 6 is an example of this requirement.

Chapter 4: A Geography Lesson (Quiz)

Python Elements: Text Entry plus If, Or, Else

The curriculum focus of this Chapter is geography, with a classroom quiz based on If, Or, and Else in Python coding (see Figure 7).

```

Monday 1st April 2019
#Programmer Beth Mead
#Thanks to Ethan and Kate (testers)
print("A Geography Lesson")
print()
print("Malika is a very kind cat. She has always liked helping others in need, and so she is friendly with almost all of the teachers at the school.")
print("One morning the phone rang. It was Miss Bindi, the Maths teacher of Geography, who came from Australia.")
print("'''O' day Malika," Miss Bindi said. "Would ya mind coverin' my lesson t'day? It's the last day of school, and I'm goin' back t' Oz for the holidays.'''")
print("Malika, of course, said she would. She then called Maisey, who said she wanted to come, too.")
print("Maisey and Malika met outside the school gates.")
print("'''So, what are we teaching today?' Maisey asked.")
print("'''We are teaching the same class as usual, but we are teaching them Geography.'")
print("'''Maisey and Malika knew the class so well now they didn't even need to call the register. They just did it themselves.'''")
print("'''They taught the class some general geographical knowledge. The class enjoyed the lesson very much.'")
print("'''At the end of the lesson, Malika spoke to the class.'")
print("'''This is a quiz on what we have learnt," Malika said. "Remember you must NOT put Full Stops at the end, as the computer will mark it as wrong.'''")
print()
print("What is the capital city of France?")
ans=input()
if ans=="Paris" or ans=="paris":
    print("You are correct!")
else:
    print("It's called Paris.")
print()
print("What is the longest river in the UK?")
ans=input()
if ans=="Severn" or ans=="severn":
    print("Well done!")
    print("Close, but it is spelt Severn.")
else:
    print("It's called The River Severn.")
print()
print("What are the four MAIN compass points?(Remember capital letters!)")
print("NOTE: put it in NOME order.")
ans=input()
if ans=="North South East West" or ans=="North South East and West" or ans=="North, South, East and West":
    print("Correct!")
else:
    print("North, South, East and West are known as compass points.")
print()
print("Is the North or South pole colder?")
ans=input()
if ans=="North" or ans=="North" or ans=="North pole" or ans=="North pole" or ans=="North Pole":
    print("It's the south that is colder.")
if ans=="South" or ans=="South" or ans=="South pole" or ans=="South pole" or ans=="South Pole":
    print("Well done! Not many people know that.")
print()
print("Well done! You have completed the quiz!")

```

Fig. 7. If, Or, and Else in Python coding

Chapter 5: A History Lesson - Illuminated Manuscripts

Python Elements: Text Entry plus Turtle Graphics

This was found to be harder than it looked. Angles were a challenge, but we were pleased with the result (see Figure 8).

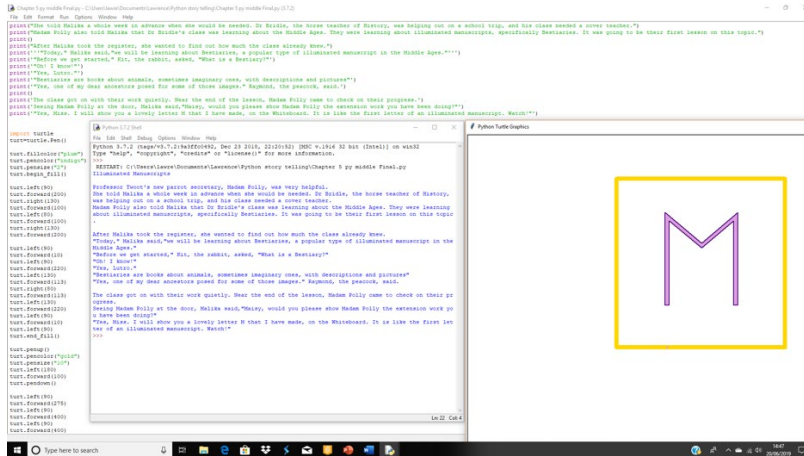


Fig. 8. Coding input with story output, and an “illuminated” letter M

This ended up as a Chapter about a history lesson. However, Beth and I had some discussion about it possibly being part of an English punctuation lesson (on capital letters) or possibly an art lesson (on calligraphy), or mathematics (on angles).

3 Outcomes

The “Literacy from Python” project is steadily meeting its objectives. The teachers’ website (see Figure 9) now has a series of five completed exemplar stories in short “Chapters” in which the animal characters, using Python, explore simple arithmetical concepts such as addition, and times-tables; a lesson in geometry; a lesson in biology; a lesson with a geography quiz (if, or, else); and using Python to create lettering (graphics). There are background and introductory notes for teachers, a MS PowerPoint file indicating the general teaching method, and links to related websites, including one on constructionism, on which the project is based. The project also supports the Programmes of Study for the England National Curriculum in Computing for Key Stage 3. So far (October 2021), the Python website has had over 1,000 visitors. How far the ideas posted there have been applied in schools remains to be seen: we have added a Contact Form to the website to support this evidence gathering.



Fig. 9. Literacy from Python website

This constructionist project therefore brings together many elements of the England school computing curriculum (including ICT skills, computer programming, and computational thinking), together with creative writing, elements of simple mathematics, geography, biology, and other curriculum subjects, all presented through storytelling (see Figure 10).

	Literacy elements	Curriculum focus	Python coding elements
Chapter 1	Narrative, description, dialogue	Mathematics	Text entry; times tables
Chapter 2	Narrative, description, dialogue	Geometry	Text entry; Turtle graphics
Chapter 3	Narrative, description, dialogue	Biology	Text entry; Turtle graphics
Chapter 4	Narrative, description, dialogue	Geography	Text entry; If, Or, Else
Chapter 5	Narrative, description, dialogue	History	Text entry; Turtle graphics
Chapter 6 (pending)	Narrative, description, dialogue	TBC	Lists, loops, libraries, iteration

Fig. 10. Project elements and future plans

4 Future Plans

Further work is currently under discussion with local schools and colleges, with a view, post-Covid, to widening the number of students participating in the project, and to developing the number of computer coding concepts, such as loops or lists, within it. The initial project aims and objectives have, however, all been met: the constructionist/contextualised method is now clearly established through the exemplar stories; the supporting website has been started; and the dissemination of these ideas (through IFIP TC3 OCCE 2021 Tampere) has begun. We need, next, to ascertain how many, and how easily, pupils can develop their own narrative script; create lists, loops, iteration and so on; and then integrate these elements successfully into their narrative scripts, thus contextualising their learning [8]. There are two curriculum levels possible for this development in England schools: at Key Stage 3 (ages 11 to 14 years), and within Key Stage 4 GCSE Computer Studies courses (ages 14 to 16 years), where a coursework element in coding is a requirement. It may also be a useful model for postgraduate teacher training courses in computer science. In this way, local school and college feedback can, in

the future, be incorporated into the project and its website, and greater quantitative data about uptake of the model will thereby be generated to assist in evaluating the success, in wider practice, of the teaching approach suggested in this paper.

The international success of the cross-curricular coding project “Literacy from Scratch”, encourages our development of “Literacy from Python”, despite current progress being severely slowed by the Covid pandemic.

We hope that the free teaching materials on the now-integrated websites (“World Ecitizens”, “Literacy from Scratch” and “Literacy from Python”) may be of practical use in the classroom.

References

1. Passey, D., Mavengere, N. (Eds.): Literacy from Scratch. In Proceedings of the 10th IFIP World Conference on Computers in Education, Torun, Poland (July 2-5 2013) pp 17-27, Springer, Heidelberg, Germany (2013)
2. Demo, G.B., Williams, L.: A Working Model for Teacher Training in Computing through the “Literacy from Scratch Project” - The Many Facets of Scratch”. In Proceedings of the ISSEP Conference 2014, September, Istanbul, Turkey. Springer, Heidelberg, Germany (2014)
3. Passey, D., Tatnall, A. (Eds.): An International Working Model for Teacher Training in Computing: Key Competencies for Educating ICT Professionals”. Proceedings of the 10th IFIP World Conference on Computers in Education, Potsdam, Germany, pp 25-33, Springer, Heidelberg, Germany (2014)
4. Kalaš I., Mikolajová, K., Tomcsány, K.: “Konštrukcionizmus. Od Piageta po školu v digitálnom. DidInfo 2011, Slovakia (2011)
5. Williams, L. (Ed.): Introducing Computing: a guide for teachers. Routledge, London (2014)
6. Vygotsky, L. S.: Mind in Society: The Development of Higher Psychological Processes. Harvard University Press, Cambridge, MA (1978)
7. Williams, L., Černochová, M., Dorling, M.: Developing Computational Thinking Skills through the Literacy from Scratch project: An International Collaboration. 11th IFIP Conference, July 2015, Vilnius, Lithuania (unpublished paper) (2015)
8. Stowell, L.: Coding for Beginners: Using Python. Usborne, Tulsa, TX (2018)
9. Luckin, R. (Ed.): Enhancing Teaching and Learning with Technology - What the Research Says. UCL Press, London (2018)
10. Younie, S. (Ed.): Debates in Computing and ICT Education (Debates in Subject Teaching) pp 97 – 114, Routledge, London (2017)