

Plan and program a monologue

Computing Science Concepts

- Sequence
- Order can be important in sequences
- Algorithm
- PRIMM
- Four levels of abstraction
- Inputs

National Curriculum Programs of Study

Pupils should be taught to:

design, write and debug programs that accomplish specific goals, including controlling or simulating physical systems; solve problems by decomposing them into smaller parts

use sequence, selection, and repetition in programs; work with variables and **various forms of input and output**

use logical reasoning to explain how some simple algorithms work and to detect and correct errors in algorithms and programs

Research Focus



This planning uses

PRIMM¹ methodology where the code is provided and pupils are encouraged to predict what it will do before investigating it, modifying it and creating their own version.

Computer scientists have **four levels of abstraction**². The ideas level, Planning level (which includes the algorithm), code level, and execution level (testing the code).

Monologue Planning Version C PRIMM

Overview

1. Introduce the concept of sequence through everyday examples

2. Extend the concept of sequence through role play

3. Introduce Scratch if it has not been used before

4. Introduce the challenge by sharing the idea, algorithm and example code for pupils to PRIMM [Predict, Run, Investigate, Modify, Make]

5. Pupils plan their own short monologue

6. Pupils turn their algorithmic planning into their own programmed monologue

7. Pupils fill in the summative assessment form

1. Introduce the concept of sequence through everyday examples

Download **Everyday computing concepts PDF** from <http://code-it.co.uk/knowledge> or directly from <http://code-it.co.uk/wp-content/uploads/2019/04/everydaycomputingconcepts.pdf>

Use the first eight slides to introduce the idea of sequence in our everyday lives. The answer mostly appear in red text. There are some sequences where the order is less important and some where the order is paramount.

Everyday sequences

Teacher instructions

- Pack away
- Stack your chairs
- Get your coats
- Line up ready to go

Picture by KalyanKabin

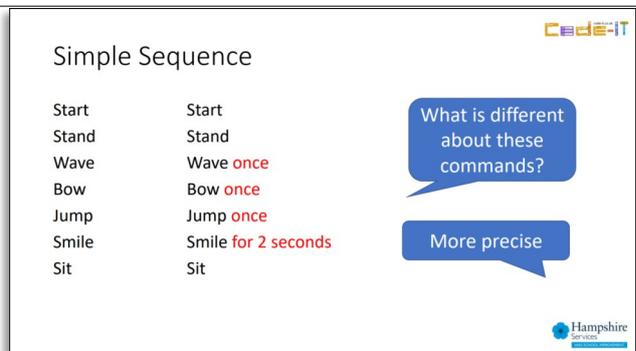
The order does matter for this sequence but it might not always matter for every sequence.

The same will be true for programming sequences. By linking the concept to its everyday use you are linking to known knowledge which means pupils are more likely to assimilate the idea.

2, Extend the concept of sequence through role play

Download Concepts before coding PDF from <http://code-it.co.uk/knowledge> or directly from <http://code-it.co.uk/wp-content/uploads/2019/04/conceptbeforecoding.pdf>

Follow the links in the menu to simple sequence. Use those six slides to roleplay and write simple fun sequences. These slides introduce the idea that the more precise a sequence is the more useful it is. Stop before you get to the dance slide.



Start	Start
Stand	Stand
Wave	Wave once
Bow	Bow once
Jump	Jump once
Smile	Smile for 2 seconds
Sit	Sit

What is different about these commands?

More precise

Ceddit Hampshire services

Formative assessment opportunity

While pupils are writing their own sequences go round and check them all. Is anyone struggling? Have they copied the one on the board exactly? This is often an indication that they are not sure how to create their own or that spelling is an issue. A good supportive activity is to get them to tell you about their own sequence that you scribe for them.

3, Introduce Scratch if it has not been used before

If pupils have never used Scratch before it is worth going over the basics. I have included some videos here Scratch 2.0 <https://youtu.be/bNoyArexVns> or Scratch 3.0 https://youtu.be/gtqMauyKE_w but I recommend that you watch them but introduce it yourself in short bursts.

Repeat until basic Scratch introduced

Show pupils basic feature **Introducing Scratch algorithm**
Pupils have time to try it out

Sprites are pictures that we can give instructions too through dragging blocks and snapping them together in the scripts area.

There are lots of different types of blocks and they are colour coded with similar blocks. Many of these blocks will contain ideas that we won't know how to use until upper KS2 or KS3.

Starting blocks have a curved top and can be found in the events section. Drag out the when this sprite is clicked starting block to show pupils.

4, Introduce the challenge by sharing the idea, algorithm and example code for pupils to PRIMM [Predict, Run, Investigate, Modify, Make]

Which templates to use

I recommend starting with one type of programming first (say and wait blocks or timed say blocks) as pupils progress make sure they look at an example which uses the other method. The other examples you can save for

when this sprite clicked



```
when this sprite clicked
say All the Britons, for 2 seconds
say dye themselves with woad, for 3 seconds
say which is a bluish colour, for 3 seconds
say This makes them look more terrible appearance in
say They wear their hair long for 3 seconds
say and have every part of their body shaved, for
say except their head and upper lip, for 3 seconds
```

Julius Caesar on the Britons

What challenge to use

This module comes with four pre-made challenges.

Tables –3 times tables (say and wait blocks)

Victorian –Victorian working conditions (say and wait blocks)

Julius C –His account of the Britons when he invaded (timed say blocks)

Churchill –His speech in 1940 (Timed say blocks)

You can of course make your own really easily and there is a blank word template to create the idea and algorithm. Scratch supports copy and paste.

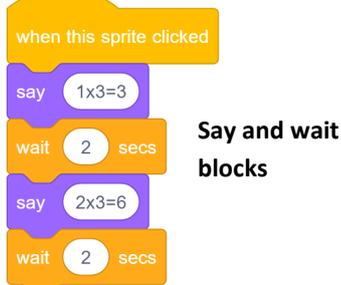
Idea and algorithm planners

If pupils are only writing a monologue use the A4 monologue planner.

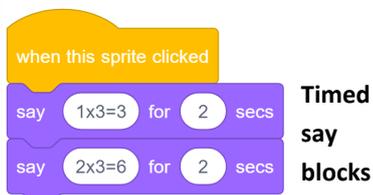
monologuealgorithmplannerA4
ZIP

If pupils are going to complete the stage and sound module or animation modules then use the left half of the A3 monologue planner.

monologuealgorithmplannerA3
ZIP



Say and wait blocks



Timed say blocks

pupils who complete faster.

Open a template file

Direct pupils to open the template file. This can be found either on the Scratch website at <https://scratch.mit.edu/projects/303987509/>

Or as a downloadable file to put on your network

Scratch 2.0 **monologueVCscratch2** ZIP

Scratch 3.0 **monologueVCscratch3** ZIP

You can adapt this planning to read the text out loud or translate it into another language but this will increase the complexity of the project.

Predict and run

Ask pupils to predict what they think the code will do before they run it. Once they have predicted to their programming partner and run the code give pupils the idea and algorithm that goes with their example **monologuealgorithm PDF**. How close was their prediction to the authors idea?

Investigate & Modify

There are some slides **PRIMM Monologue Module C investigates slides PDF** with some focus questions for pupils to answer with the answers. This aspect can be good to work in similar ability pairs as it give space to discuss possible answers. Put up the first slide and give pupils a chance to explore and find answers before running through the answers with the class.

5. Pupils plan their own short monologue

Give out the idea and algorithm planner. You can simplify this by giving pupils an idea or you could let them choose their own idea. This could have any cross curricular theme.

Formative assessment



Check pupils algorithms as they are creating. Do they make sense? Have they kept to their idea? Have they included punctuation? Have they included timings? Do they know what type of blocks their algorithm can easily convert into? Although it is easy to adjust an algorithm to output in either say and wait or timed say blocks.

Check pupils algorithms before they turn them into code.

6. Pupils turn their algorithmic planning into their own programmed monologue

Give pupils time to do this. Those that finish earlier than others can create a different type of monologue.

7. Pupils fill in the summative assessment form

You can find a summative assessment quick Kahoot Quiz linked at <http://code-it.co.uk/gold/>

Whilst Kahoot is a limited assessment tool it is free and it is easy for teachers to pass the results back to code-it via phil.bagge@code-it.me so I can look at which method provides best short test results. Not conclusive but useful.

If you pass the results back please

- 1, Anonymise the results by removing the names
- 2, Ask the head teacher for permission
- 3, In the email title say which module you are doing (ie Animation D)

Research References



¹ PRIMM Sentence

<https://blogs.kcl.ac.uk/cser/2017/09/01/primm-a-structured-approach-to-teaching-programming/>

² Four levels of abstraction

This article includes an example of the four levels of abstraction and sign posts the work of Waite and Armoni in using them with school level pupils.

<http://code-it.co.uk/algprogdiff/>



Inspired by Behaviour Rubric created with @MarkDorling and linked at <http://code-it.co.uk/attitudes/>