

# Variables as Placeholders

## Computing Science Glossary

**Completion** A key part of the code is missing and pupils need to add it into the program before using and modifying.

**Algorithm** Part of planning stage before programming written for another human to read.

**Code** Written for a digital device

**USE MODIFY MAKE** A strategy that promotes using and modifying code before creation

**Variable** consists of a name and assigned value. Read the name get the value.

### Investigate Change Add Create

This has a smaller repeated cycle making it easier to leave examining the last harder example (Good for SEN)

## Placeholder

### Questions & Answers

**How does this fit in with other app programming?** See overview document

**What age is this for?** Y5+ is a good place to use this if pupils have done other work on sequence, repetition and selection first.

**How hard is this to teach?** Very easy as all the code instructions are in the booklets.

**How do we Assess learning?** Pupils use answers provided to mark their own work.

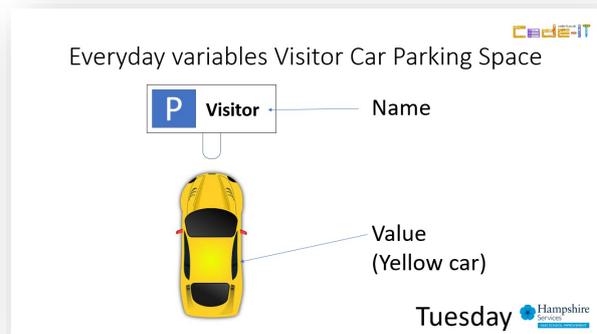
**Is it in line with NC?** Yes see next page.

**Why is there a choice?** All of the methods chosen have good research behind them but we don't know which are best or even if there is a best for all pupils. We do know that it helps pupils to encounter a variety of different types of method so they are continually challenged. If you work your way through all modules I recommend you switch strategies each time to keep the challenge high.

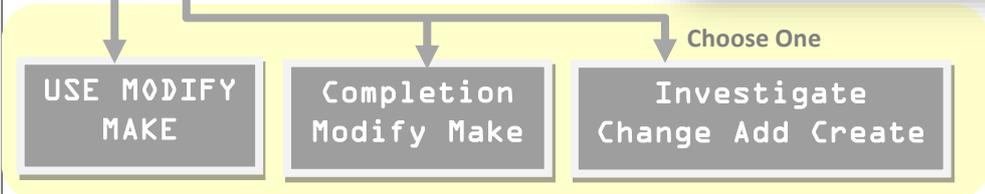
**How creative is this?** This combines the best knowledge we have about how to learn something new with the opportunity at the end for pupils to create something that they want to create that uses their new knowledge.

### Everyday Placeholder Variables

Download **Everyday computing concepts PDF** from <http://code-it.co.uk/wp-content/uploads/2019/04/everydaycomputingconcepts.pdf> Use slides 25-39 (everyday selection) to introduce the idea of variables as placeholders in our everyday lives. By linking the concept to its everyday use you are linking to known knowledge which means pupils are more likely to assimilate the idea into an established mental schema.



Everyday Placeholder Variables



## Booklet Choices

You choose from one of the options above. Each option has its own booklet which guides pupils through the stages, making them think deeply about either the code or the algorithm before modifying it and having a choice of things to make. Pupils are instructed when to work in pairs and when to work alone. Separate the answer sheet (last pages of the booklet) into sections as pupils will need to mark their work as part of their learning process. If you are not sure which to choose download the booklets and look at the differences.

## National Curriculum Programs of Study

(bold text is covered in this module)

**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**

### Before the module

Read the planning and download the PDFs for everyday Variables, Decide which booklet (s) variation you are going to use and download and print it out one per pupil. Remove the answer sheets for pupils and separate them into sections for pupils to access when they need to. Download the code for your version of Scratch 2 or 3 and place it on your network where pupils can access it, or note where it is on the Scratch website if using Scratch online.

### Formative assessment support

If pupils are struggling to predict, investigate/use or modify ask them to read code or algorithm out loud. If pupils are struggling to work together in a meaningful way then encouraging and rewarding positive attitudes to working collaboratively using the communicates stickers shown at the end helps. Lots of misconceptions can be solved by reading the code or algorithm slowly and out loud to their partner.

### Classroom Organisation

In some sections pupils are asked to work with a partner of similar programming ability. If you are not sure what programming ability they are go with Maths skills as a starting place. You can move partners around between modules so that pupils benefit from different interactions.

### Assessing which version helps

Get pupils to mark their booklets, collect in all the marks by sections. Collect the marks from the modify section and calculate a mean average for the whole class. Email or Tweet this to phil.bagge@code-it.me or @baggiepr stating clearly what age, module and version your class did. For example Y5 9-10 Years old Diving Beetle PRIMM Algorithm Mean average 5.2/9 for 32 pupils. Any other feedback, verbal or written, is also welcome.

## Resources

### Everyday Variables

[PDF Download](#) Slides 25-39  
(20 mins)

### Completion Modify Make

Pupil booklets  
Scratch 2 & 3 Code to download  
Scratch 3 code on Scratch website

### Investigate Change Add

Pupil booklets  
Scratch 2 & 3 Code to download  
Scratch 3 code on Scratch website

### USE MODIFY MAKE

Pupil booklets  
Scratch 2 & 3 Code to download  
Scratch 3 code on Scratch website

### All Resources at

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

## Further Research Reading



### Use Modify Create

Irene Lee et al Computational thinking for Youth in practice (2011)

### PRIMM Sentence

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

### Four levels of abstraction

This article includes an example of the four levels of abstraction and sign posts the work of Waite and Armoni

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

Placeholder Variables Learning Path (Trajectory)

EverydayVariables Read the name get the value Car Parks & Weddings

Which version has teacher chosen for the whole class, groups of pupils or each pair?

Everyday variables Visitor Car Parking Space

Tuesday

COMPLETION VERSION  
Pupils add the variables into one of the example programs using the planning to help them

```
set user_name to Dolly
say user_name for 2 seconds
say user_name for 2 seconds
say user_name for 2 seconds
set user_name to daniel
```

```
when this sprite clicked
hide list jobs
search for keyword to spotlight stage
say Klaus for 2 seconds
say Original music by Louis Armstrong for 3 seconds
say Singalong for 2 seconds
start sound heliodivy.mp3
say yes Ch. hello for 0.5 seconds
```

Pupils use marksheet to check if their version is correct

INVESTIGATE CHANGE ADD DESIGN MAKE

Pupils work with one code sample

INVESTIGATE Pupils examine and run code and answer questions about it

What is the value of the user\_name variable at the start of the program?

CHANGE Pupils modify code in response to a series of written challenges. They record their answers

Change the value of the user\_name variable

ADD Pupils add things to the code responding to a challenge

Add a question to the bottom of the quiz

Pupils use marksheet to check answers

Is pupil pair getting behind?

Have you worked through two code samples?

Have you worked through three code samples?

MAKE Pupils chose what they want to make from a selection of challenges

Use a variable to personalise a previously created program

USE MODIFY CREATE VERSION

USE Pupils examine and run code and answer questions about it

Pupils use marksheet to check answers

MODIFY Pupils modify code in response to a series of written challenges. They record their answers

Pupils use marksheet to check answers

CREATE Pupils chose what they want to make from a selection of challenges

Colour Key

- Whole Class
- Same Ability Pair
- Coding individually supported by previous partner

*I recognise there is more than one way to solve/describe a problem*

*I don't just accept the first solution*

*I look for a range of solution to the same problem*

*I can evaluate my solutions against a set criteria*



*I look for how a project can be extended*

*I can design criteria to evaluate my creations*



Insert picture of your students here



*I can break complex problems into parts*

*I can contribute useful ideas to a partner or group*

*I can discover / concentrate on the most important part of a problem*

*I can encourage others to share their ideas*



*I can identify patterns in problems & solutions*

*I can adapt existing ideas to solve new problems*

*I lead using all the people talent in my group*

Communicates

Adapts

*I can develop, test and debug until a product is refined*

*I learn from setbacks and don't let them put me off*



Investigates



*I make predictions about what will happen*

*I can persevere even if the solution is not obvious*

Perseveres

*I repeatedly experiment through predicting, making, testing & debugging*

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