

Knowledge Organiser: Programming

Summary

Programming is writing computer code to create a program, in order to solve a problem. Programs consist of a series of instructions to tell a computer exactly what to do and how to do it.

An **algorithm** is a set of instructions that describes how to get something done. It is crucial that the steps in an algorithm are sequenced and performed in the right order - otherwise the algorithm will not work correctly. Algorithms can be designed using **pseudocode** and **flow charts**. They are written using **statements** and **expressions**. There are three basic building blocks (constructs) to use when designing algorithms: **sequencing**, **selection** and **iteration**. We create programs to **implement** algorithms. Algorithms consist of steps, where programs consist of statements.

In programming, iteration is often referred to as '**looping**', because when a program iterates it 'loops' to an earlier step. It is implemented using **FOR** and **WHILE** statements. Selection is implemented in programming using **IF** statements.

Key Vocabulary

Algorithm	A sequence of logical instructions for carrying out a task. In computing, algorithms are needed to design computer programs.
Flowchart	A diagram that shows a process, made up of boxes representing steps, decision, inputs and outputs.
Instruction	A single action that can be performed by a computer processor.
Programming	The process of writing computer software.
Programming language	A language used by a programmer to write a piece of software. There are many programming languages.
Pseudocode	A method of writing up a set of instructions for a computer program using plain English. This is a good way of planning a program before coding.
Variable	In a computer program, this is a memory location where values are stored.

Variable

Computer programs use variables to store information. Variables could be used to store the score in a game, the number of cars in a car park or the cost of items on a till. They work in a similar way to algebra, where a letter in your code can stand for a number.



Sequencing

Sequencing is the specific order in which instructions are performed in an algorithm.

Algorithms consist of instructions that are carried out (performed) one after another.



Selection

Selection is a decision or question.

At some point, a program may need to ask a question because it has reached a step where one or more options are available. Depending on the answer given, the program will follow a certain step and ignore the others.



Iteration

Iteration is the process of repeating steps.

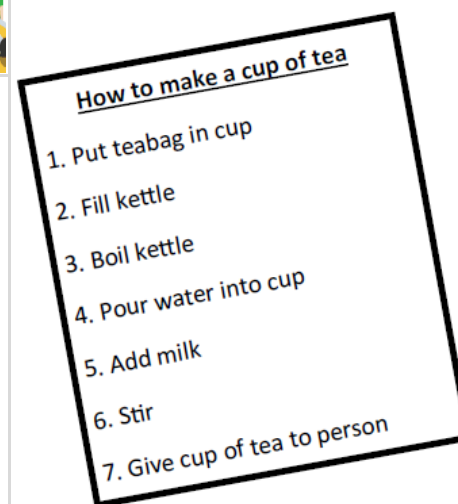
Iteration allows us to simplify our algorithm by stating that we will repeat certain steps until told otherwise. This makes designing algorithms quicker and simpler because they don't have to include lots of unnecessary steps.



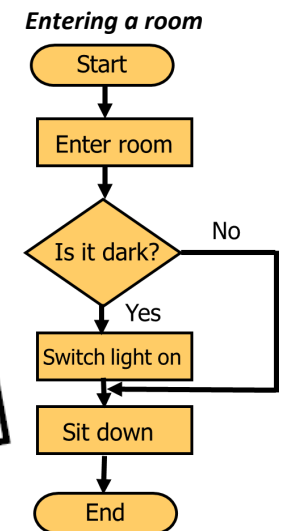
Algorithms

Algorithms can be represented as **pseudocode** or a **flowchart**, and programming is the translation of these into a computer program.

Pseudocode



Flowchart



Knowledge Organiser: Programming

Summary

Sometimes we need computers to remember the information we give it and that it calculates during programs. A **variable** can be thought of as a box that the computer can use to store a value. The value held in that box can change or 'vary'.

A program can use as many variables as it needs it to. **Variables are a key element of programming.** They are used for calculations, for storing values for later use, in decisions and in iteration. It is important to use meaningful names for variables.

Programs require data to be **input**. This data is used (**processed**) by the program, and data (or information) is **output** as a result. Once data has been processed, programs often need to output the data they have generated. In Python, the **'print' statement** is used to output data.

Key Vocabulary

Assignment	Setting the value of a variable in a computer program.
Constant	A value in computer programming that does not change.
Data Type	In computer programming, data is divided up and organised according to type, e.g. numbers, characters and Boolean.
Debug	The process of finding and correcting programming errors.
Execute	To run a computer program.
High-level language	A computer programming language used to write programs. They need to be translated into machine code through a compiler, interpreter or assembler.
Machine code	Also called object-code, this is low-level code that represents how computer hardware and CPUs understand instructions. It is represented by binary numbers.
Runtime	The period when a computer program is executing or running.
Syntax	Rules governing how to write statements in a programming language.

Python Language & Syntax

Control Flow

```

if conditional:
    <body>
elif conditional:
    <body>
else:
    <body>
for value in list:
    <body>
    continue
    break
while conditional:
    <body>
    continue
    break
    
```

Comparisons

```

value1 == value2      "str" == "str" → True
value1 != value2     "str" != "str" → False
value1 < value2      1 < 2 → True
value1 <= value2     2 <= 2 → True
value1 > value2      2 > 3 → False
value1 >= value2     3 >= 3 → True
value is [not] None
value in list        1 in [2,3,4] → False
isinstance(class instance, ClassName)
    
```

Comments

```

"""
# Line Comment
Multi-line comment
"""
    
```

Variable Assignment

```

integer = 1
string = "string"
unicode_string = u"unicode string"
mutli_line_string = """ multi-line
string
"""
tuple = (element1, element2, element3, ...)
list = [ element1, element2, element3, ... ]
dictionary = { key1 : value1, key2 : value2, ... }
dictionary[key] = value
class_instance = ClassName(init_args)
    
```

Basis Arithmetic

```

i = a + b          i = a - b
i = a / b          i = a * b
i = a % b          e.g. 11 % 3 → 2
    
```

Frequently Used Built-in Types

True	False	None
str	unicode	int
float	list	dict

Other than **True**, **False** and **None**, these can also be used as functions to explicitly cast a value to that type

Data types

Different types of data are represented in different ways inside a computer and need varying amounts of memory to store them.

Data type	Example	Size
Integer (whole number)	4, 27, 65535	1 - 8 bytes
Floating point (decimal number)	4.2, 27.4, 5.63	4 - 8 bytes
Character	A, a, 3, \$, £, #	1 byte
String	Abc, hello world	Limited
Boolean	true or false	1 bit