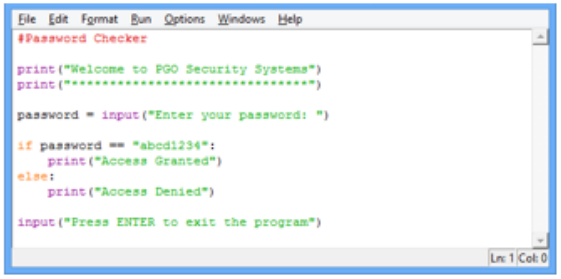
Programming with Python

Programming with Python



Python's Development Environment

Called IDLE – Integrated Development Environment

Two Modes:
Interactive Mode lets you see your results as you type them.
Script Mode lets you save your program and run it again later.

Writing error-free code

When writing **programs**, **code** should be as legible and error free as possible. **Debugging** helps keep **code** free of **errors** and documenting helps keep **code** clear enough to read.

Syntax errors

Syntax is the spelling and grammar of a **programming language**. In **programming**, a **syntax error** occurs when:

- there is a **spelling mistake.**
- there is a **grammatical mistake.**

Data Types

String - holds alphanumeric data as text
Integer - holds whole numbers
Float - holds numbers with a decimal point
Boolean - holds either 'True' or 'False'

Defining Variable Data Types

Python automatically assigns a data type to a variable. You can override this to manually define or change the data type using:

str() , **int()** or **float()**

Selection

When designing **programs**, there are often points where a **decision** must be made. This **decision** is known as **selection** and is implemented in **programming** using **IF statements**.

Operator	Meaning	Example	Evaluates to
==	equal to	7==7	True
!=	not equal to	6!=7	True
>	Greater than	7>6	True
<	Less than	5<8	True
>=	Greater than or equal to	6>=8	False
<=	Less than or equal to	7<=7	True

Procedures

A **procedure** is a small section of a **program** that performs a specific task. **Procedures** can be used repeatedly throughout a **program**. **Procedures** can make **code** shorter, simpler, and easier to write. Writing a **procedure** is extremely simple. Every **procedure** needs:

1. *A name*
2. *The **program** code to perform the task*

Variables

A **variable** is a location in **memory** in which you can temporarily store text or numbers. It is used like an empty box or the Memory function on a calculator. You can choose a name for the box (the "**variable name**") and change its contents in your **program.**

Using a Variable (firstname)

print ("What is your name?")
firstname = input()
print ("Hello,",**firstname**)

Functions

Functions are special keywords that do a specific job. **Functions** appear in purple.

print() and **input()** are examples of functions

```
print ("What is your name?")
firstname = input()
print ("Hello,",firstname)
```

Adding Comments

Comments are useful to help understand your **code.** They will not affect the way a **program** runs. **Comments** appear in red and have a preceding # symbol.

```
#firstname is a variable
print ("What is your name?")
firstname = input()
print ("Hello,",firstname)
```

Iteration

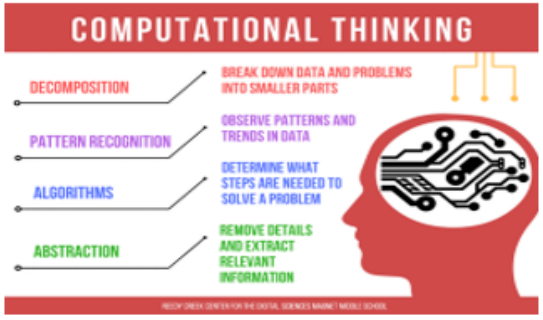
Algorithms consist of steps that are carried out (performed) one after another. Sometimes an **algorithm** needs to **repeat** certain steps until told to stop or until a particular condition has been met. ***Iteration is the process of repeating steps.***

Iteration allows us to **simplify** our **algorithm** by stating that we will **repeat** certain **steps** until told otherwise. **Iteration** is implemented in **programming** using **FOR** and **WHILE** statements.

There are **two** ways in which **programs** can **iterate** or **'loop'**:

- count-controlled loops
  - Sometimes it is necessary for **steps** to **iterate** a **specific number** of times.
- condition-controlled loops
  - **iteration** continues **while**, or **until**, a **condition** is met.

Each type of **loop** works in a slightly different way and produces different results.

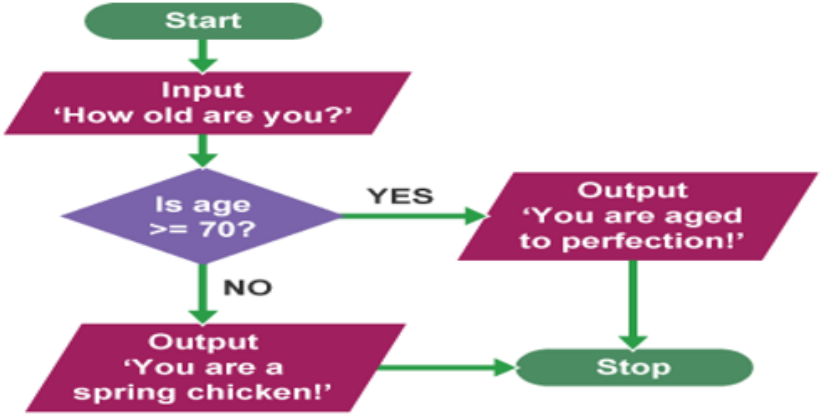


IF Statements

When designing **programs**, there are often points where a **decision** must be made. This **decision** is known as **selection** and is implemented in **programming** using **IF** statements. In **programming**, **selection** is usually represented by the statements **IF** and **ELSE**.
For **selection**, Python uses the statements **if** and **else** (note the lowercase **syntax** that **Python** uses):

Consider the age-related **algorithm** using **Python**. The steps are:

- Ask how old you are
- if you are 70 or older, say "You are aged to perfection!"
- else say "You are a spring chicken!"



The above algorithm would be written in Python (3.x) as:

```
age = int(input("How old are you?"))

if age >= 70:

        print("You are aged to perfection!")

else:

        print("You are a spring chicken!")
```

Arrays

An **array** is a series of **memory** locations – or **'boxes'** – each of which holds a single item of **data**, but with each box sharing the same name. All **data** in an **array** must be of the same **data type**.

Arrays are named like **variables**. The number in brackets determines how many **data** items the **array** can hold. The array **score(9)** would allow ten data items to be stored.

Any **facility** that holds more than one item of **data** is known as a **data structure**. Therefore, an **array** is a **data structure**.

Lists are **data structures** similar to **arrays** that allow **data** of more than one **data type**.

Functions

A **function** is also a small section of a **program** that performs a specific task that can be used repeatedly throughout a **program**, but the task is usually a **calculation**. **Functions** perform the task and return a value to the main **program**.

Every **function** needs:

1. A name
2. The values that it needs to use for calculation
3. The **program** code to perform the task
4. A value to return to the main program

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 (**process**ed) 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.

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
```

e.g.

```
if i == 7:
    print "seven"
elif i == 8:
    print "eight"
else:
    print str(i)
for i in [1, 2, 3, 4]:
    if i == 2: continue
    if i == 3: break
    print i
while True:
    print "infinity"
```

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

```
"""
    Multi-line comment
"""
```

```
# 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

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.

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

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.

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

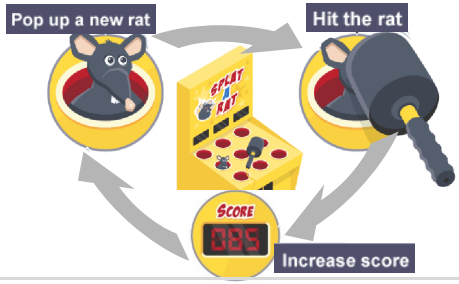
Selection is a decision or question.

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.



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.

Algorithms

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

Pseudocode

How to make a cup of tea

1. Put teabag in cup
2. Fill kettle
3. Boil kettle
4. Pour water into cup
5. Add milk
6. Stir
7. Give cup of tea to person

Flowchart

Entering a room