

# Knowledge Organiser - Python

## Key Terms & Definitions

1	Algorithm	A sequence of steps used by a human or computer to solve a problem or complete a task
2	Program	An algorithm expressed in a programming language
3	Programming language	A set of rules for instructing a computer to perform specific tasks
4	Interpreter	A program which translates high level language code to machine code and executes it
8	Input	Any method of getting data into the computer
9	Output	Any method of getting data out of the computer
10	Variable	A storage location with a name. The data in a variable can be changed after being initially set
11	Assignment	A statement in a programming language used to set or reset the data stored in a storage location identified by a variable name
12	Syntax error	An error that has occurred because the programmer has not followed the rules of the programming language they're using
13	Logical error	When a program does not behave in the way that it should, even though the programmer has followed the rules of the language
14	Arithmetic expression	A mathematical operation, for example, $10+5$
15	Evaluate	In the context of programming, to evaluate an expression means to simplify it. For example, the arithmetic expression $10+5$ evaluates to 15
16	String	A sequence of characters, for example "Hello world"
17	Sequence	One of the three basic programming constructs. Instructions that are carried one after the other in order.
18	Selection	One of the three basic programming constructs. Instructions that can evaluate a Boolean expression and branch off to one or more alternative paths.
19	Iteration	One of the three basic programming constructs. A selection of code that can be repeated either a set number of times (count-controlled) or a variable number of times based on the evaluation of a Boolean expression (condition-controlled).
20	Comparison operator	Used to compare two expressions
22	Subroutine/function/procedure call	Specifying the point at which code in a subroutine is to be run
23	Importing	Specifying that you want to use a particular code library of subroutine contained within the library
25	Condition	A boolean expression being used to make a decision

26	Increment	Increase a number by 1
27	Decrement	Decrease a number by 1
28	Type casting	Converting a value from one data type to another. For example, converting the string "10" to the integer 10.
29	Boolean	A data type which can take two possible values: true or false

## Output

The `print` function is used to write output to the screen. `print` takes one or more arguments (strings or variables between the brackets) and writes the data to the screen.

## Output Examples

```
print("Hello World!")
```

```
print("Hello", name, "nice to meet you")
```

## Variable Assignment

Variable assignments **are not** equations. Variable assignments are instructions for the computer. This means that the data stored in a variable can change throughout the runtime of the program.

## Assignment examples

```
# Example 1
```

```
name = "Sam"
```

```
# Example 2
```

```
friendName = "Jake"
```

```
# Example 3
```

```
total = 20 + 60 + 40
```

```
# Example 4
```

```
area = 3.14 * r * r
```

## Input

The `input` function is used to prompt the user to enter some data using the keyboard. `input` can take a string argument which is used as a prompt to the user to tell them what data the computer is expecting.

## Input Examples

```
# Example 1
```

```
name = input("What is your name?")
```

```
# Example 2
```

```
age = int(input("What is your age?"))
```

# Selection

An `if` statement can be used to implement selection in Python. It is optionally followed by an `elif` and/or an `else` statement.

## Selection Examples

# Example 1

```
if age >= 18:
    print("You can watch the film")
else:
    print("You can't watch the film")
```

# Example 2

```
if age >= 18:
    print("You can watch any film")
elif age >= 15:
    print("You can only watch films with a 15 rating or below")
elif age >= 12:
    print("You can only watch films with a 12 rating or below")
else:
    print("You can only watch PG or U rated films")
```

# Example 3

```
if len(password) < 8:
    print("Password is too short!")
```

# Condition-controlled iteration

A `while` statement can be used to repeat a section of code until a condition becomes false.

## Condition-controlled iteration examples

# Example 1

```
correct = False
while not correct:
    password = input("Enter your password: ")
    if password == correctPw:
        correct = True
        print("Access granted")
    else:
        print("Access denied: incorrect password")
```

# Example 2

```
sum = 0
done = False
while not done:
    price = int(input("Enter the price of the next item of shopping or 0 if you're done entering items: "))
    if price == 0:
        done = True
    else:
        sum += price
```

```
print("Your shopping comes to", sum)
```