**Python**

**Print**

In Python the print() function is used to tell a computer to talk;

print("There is something at work in my soul, which I do not understand.")

The printed words are referred to as **output**.

Here is another example of a simple command;

print("Hello world!")

**Strings**

Blocks of texts are strings. For example, the previous code blocks are examples of strings.

**Variables**

Programming languages offer a method of storing data for reuse. Perhaps we sant to store a greeting, we can do this in a variable.

message\_string = "Hello there"

# Prints "Hello there"

print(message\_string)

In this instance, we stored a string as a variable. When we run our code, python will print the message\_string variable which is; ‘Hello there’.

* Variables do not have spaces or symbols but they\_can have underscores.
* Variables should not begin with numbers, but they can\_have\_numbers\_after\_the\_first\_letter\_23432432.

Variables can be updated.

# Greeting

message\_string = "Hello there"

print(message\_string)

# Farewell

message\_string = "Hasta la vista"

print(message\_string)

Imagine that Python is working chronologically in a descending order. Initially there first message that will be printed will be ‘Hello there’.

Later, we re-assign the variable; message\_string to say ‘Hasta la vista.’ This will then print ‘Hasta la vista’ henceforth until the variable is updated.

**Errors**

^

Python will highlight errors with this symbol. We call the process of updating the program so that it no longer produces unexpected errors ‘debugging’.

There are two common errors in Python; SyntaxError and NameError.

Syntax Error

This means something is wrong with the way in which the program is written – punctuation that does not belong, a command where it is not expected, or a missing parenthesis can all trigger a SyntaxError.

NameError

A NameError occurs when the Python interpreter sees a word it does not recognise. Code that contains something that looks like a variable but was never defined will throw a NameError.

print('This message has mismatched quote marks!")

print(Abracadabra)

Here are two examples. Firstly we have a SyntaxError. This is because the quotation marks do not match.

Secondly we have a NameError. This is because we do not have a variable assigned to Abracadabra. If we are trying to print Abracadabra on its own, then we need quotation marks!

This code will work:

print('This message has mismatched quote marks!')

print('Abracadabra')

**Numbers**

Computers can also use numbers. There are multiple ways to store numbers.

An integer or int is a whole number. There are no decimal points.

A floating-point number, or a float, is a decimal number. It is used to represent fractional quantities as well as precise measurements.

Numbers can be assigned to variables.

an\_int = 2

a\_float = 2.1

print(an\_int + 3)

# Output: 5

In this instance when we print; the number 3 is a literal. This means it is not a variable, it is ‘literally’ the number 3.

**Calculations**