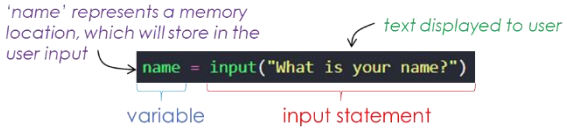
Advanced Python

Inputs & Variables

When we code, we often need our programs to receive and store inputs from the user. In python, this is achieved using the input() script, assigned to a variable.

The input() script is set up like this:



- When it is executed (run), the text inside the brackets will display on the screen and the program will pause for a 'user input'.
- Once the user has entered an input, it will be stored in the variable assigned to the input statement (e.g. name).

Outputs

When we wish to output information (strings/numbers) onto the screen, we use the print() statement.

The print() script is set up like this:

- When it is executed (run), the text inside the brackets will display on the screen.
- Unlike the input()script, there will be no pause and the program will immediately execute the next line of code in the program.
- If the print statement contains text with quotes, it will display the text inside the quotes.
- However, we can also display the contents of variables using a print() statement.
- To do this, we add the variable's label between the brackets, without the quotes.

```
1 text = "Hello World"
2 print(text)
```

```
Hello World
```

Data Types & Maths

In python, when an input statement is executed, the input is always stored by the system as a string data type. Which is fine when we enter words and sentences. However, if we are requesting that the user enters whole numbers that will then be calculated upon in our program, we must inform the system that the entered data is in fact an integer, otherwise the program will not carry out the calculations correctly.

We **cast** data to an **integer type** using the int() function.

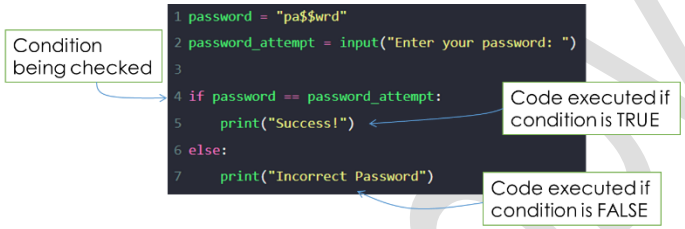
```
not_a_number = input("Enter a number)
now_a_number = int(not_a_number)
```

When the number is entered, it is first stored as a string. We use the int() function to cast the inputted value into an integer data type.

Selection & The IF-ELSE statement

Selection is a programming construct which allows programs to take different pathways (execute different lines of code), depending on a condition. In other words, it allows programs to make decisions.

This is achieved using IF statements, which in python are set up like this:



In the program above, the contents of the two variables (password and password_attempt) are being compared.
- If they match, the program will run the code under the if statement.
- If they do not match, the program will run the code under the else statement.

Iterations & The FOR loop

Iterations are a programming construct which enables the repeated execution of lines of code. The FOR loop is a type of iteration which is count controlled. This means that it loops for a set number of iterations.

In python, FOR loops are set up like this:

The **x** is simply a variable. It could have any name.

It is however a special kind of variable known as a 'stepper'

```
for x in range (m,n,o):
```

We must finish the statement with a colon

The ***m*** is representing the number that the loop will begin counting from
The ***n*** is representing the number that the loop will count up to. It will stop when it reaches this number.
The ***o*** is representing the steps that the loop is counting up in.

If the bracket contained (0,10, 2), the loop would count from 0 to 10, counting up in 2s, therefore the loop would count 0, 2, 4, 6, 8 (stopping at 10) therefore executing 5 times.

Example:

Below is a practical example of the above explanation. Notice the value of x during each loop and how many iterations there are.

```
1 for x in range(0,10,2):
2     print("variable x in this loop contains: ", x)
3
4
5
```

```
variable x in this loop contains:  0
variable x in this loop contains:  2
variable x in this loop contains:  4
variable x in this loop contains:  6
variable x in this loop contains:  8
```

Key Vocabulary

Key Word	*Definition*
Input	Values which get sent from the user into the computer
Variable	The place where inputs get stored by the program
Output	The values which get sent from the computer to the user
Decision	Deciding what to do depending on certain conditions.
IF Statement	A programming construct which enables a program to take different pathways depending on particular conditions.
Data Type	The type of data being used by the program
Variable	The place where inputs get stored by the program
Integer	"Whole Number" data type
Iteration	Another name for a 'loop'

Iterations & The WHILE loop

Iterations are a programming construct which enables the repeated execution of lines of code.

The WHILE loop is a type of iteration which is condition controlled. This means that its ability to loop will depend on a condition. For example, if the condition was x=1, the loop will only run if the variable x is equal to 1, but if it is instead equal to anything else, the loop will not run.

In python, WHILE loops are set up like this:

The **x** is simply a variable. It could have any name.

It is however a special kind of variable known as the '***most recent value***'

```
while x == != > < n:
```

We must finish the statement with a colon

The ***n*** is simply representing a value that we want x to either equal, not equal, be greater than etc depending on the while loop condition.

If n=5 and the condition was while **x != 5** (*not equal to 5*) then the loop would repeat until x equals 5.

Example:

Study the following code:

```
1 letter = "s"
2 while letter != "s":
3     letter = input("Enter a letter: ")
4 print("Loop has ended")
```

This code, will continue to ask the user to input a letter, until they enter the letter 's'.

The moment they do enter the letter 's'. the loop will stop and the next line of code in the program will execute – in this example, the print statement will run, outputting 'Loop has ended'.