Introduction

In this video we are going to learn how to work with different types of data in Python. We're going to start off by learning how to use variables. In the module we got our development environment up and running and we even ran our first Python program, the traditional Hello World. Let's dive a little deeper into that now.

In [2]:
```
# Print statement
print("Hello World!")
```

```
Hello World!
```

So what's just happened? Jupyter Notebook has passed our Python code through a Python interpreter. The interpreter, does exactly what it says, it interprets. In this case it interprets our print statement and decides what each word means.

When the interpreter sees the word **print** followed by parentheses, it prints to the screen whatever is between the parentheses.

You'll also notice that different sections of the print statement are in different colors. This is called syntax highlighting and it's performed by your code editor which in this case is the Jupyter Notebook. Our notebook recognizes that **print()** is the name of a function and displays that word in one color. It also notices that **"Hello World"** is a string and displays this text in another color. Each code editor has its own version of syntax highlighting so don't panic if your colors are different.

Variables

Let's update our print statement to include a Python variable.

In [3]:
```
# Print statement with variable
message = "Hello World!"
print(message)
```

```
Hello World!
```

In the code above we've just added a variable called message. In Python every variable is connected to a value, in this instance, the variable **message** is connected to the string **Hello World**.

In [4]:
```
# Print statement with numbers
answer = 1 + 1
print(answer)
```

```
2
```

In this code example we have a variable called **answer**. The value of answer is **1 + 1**. This example is just to demostrate that variables can hold different values not just strings or integers. In later lessons we will see lots of examples of this.

By adding variables to our print statements we are asking the Pyton interpreter to do more work. And that's OK, it can handle it! Let's combine our code from above.

In [5]:
```
# Print two variable with different values
message = "Hello World!"
print(message)

answer = 1 + 1
print(answer)
```

```
Hello World!
2
```

Naming Variables

In the examples above we have seen two variables, **message** and **answer**. When creating variables in Python there are a few rules and guidelines that you need to follow. Following these will help you write error free and clean code. Here are some tips to think about when creating variables in Python.

- Variable names can only be made of letters, numbers and underscores, (A-z, 0-9, and _)
- Variable names are case sensitive
- They can start with a letter or an underscore, but not with a number
- You cannot use spaces in variable names
- Try not to use Python keywords when creating variables
- Try and make your variable names descriptive and helpful

For some more guidelines when writing Python code, check out the Zen of Python.

In [6]:
```
import this
```

Name errors when using variables

One of the most common errors a new Python programmer experiences is the **NameError**. Let's take a look.

In [7]:
```
# Print statement with variable
message = "Hello World!"
print(mesage)
```

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
Cell In [7], line 3
      1 # Print statement with variable
      2 message = "Hello World!"
----> 3 print(mesage)

NameError: name 'mesage' is not defined
```

This output tells us, the Python programmer, serveral things:

- Where the error has occured
- What type of error has occured

When you see a name error it usually means that we either forgot to assign a value to a variable or there is a spelling mistake in the variable name. In this example, at line three, there is a spelling mistake in the variable name.

Exercises

1. Create a simplea Python Hello World script that prints "Hello World".
2. Create a new variable and assign a message of your choice to it. Print the variable out.
3. With your new variable change its value and print out the new value.