



In this section the Developer will learn...

- How to make decisions in Python

Conditional Python – If statements



'If' statements are the decision makers with the Python coding, it is their purpose to decide what processing is to be done according to a value tested.

A condition always looks for either a true or a false, whether the condition is met ... or not

An Condition test always begins with an 'if' and the condition can be enclosed by brackets.

The Condition can test for equality, non-equality and within a Range, these are termed Comparison operators.



Conditional Python – If statements

When evaluating equality, the condition uses two equal signs, any attempt to use just one results in ...

```
>>> if day = "Monday":  
    File "<stdin>", line 1  
        if day = "Monday":  
            ^
```

SyntaxError: invalid syntax

The condition in Python is followed by a colon, in other programming languages this would normally be a 'then'



Conditional Python – If statements

Generally other programming languages use physical symbols to signify an conditional block, Python is different, there are no symbols and it relies on the Developer indenting the commands in the condition block, the Python editor helps in this process by changing the normal three chevrons to three dots ...

```
>>> if day == "Monday":
```

```
...
```

This serves only to remind the Developer to manually indent the command, if that isn't done (one space will do) the following error will occur

Conditional Python – If statements



```
>>> if day == "Monday":  
... print("Its Monday")  
File "<stdin>", line 2  
    print("Its Monday")  
    ^
```

IndentationError: expected an indented block

Minimum of one space is necessary ...

```
>>> if day == "Monday":  
... print("Its Monday")  
...  
Its Monday
```

Note the second line of dots, Python expects the Developer to hit return on an empty line to close the block



Conditional Python – If statements

Every command written when Python is in 'block mode' will only execute if the condition is met, when all the commands have been entered, a double 'return' will fire the condition and return Python to normal command entry mode

In this example the condition is written in a file, note the indentation of the if statement and the second print command which will always execute regardless whether the condition is met ...

```
day = 1  
if day == 1:  
    print ('Monday')  
print ("end")
```



Conditional Python – If statements

The condition can also be written on the same line as its processing ...

```
>>> if day == "Monday": print ("Its Monday");
```

```
...
```

The condition can also be written enclosed in brackets for clarity ...

```
>>> if (day == 'Monday'):
```

```
...
```