Python Loops

Python has two primitive loop commands:

- while loops
- for loops

The while Loop

With the while loop we can execute a set of statements as long as a condition is true.

In [ ]:

```
# Example
# Print i as long as i is less than 6:

i = 1
while i < 6:
    print(i)
    i += 1
```

Note: remember to increment i, or else the loop will continue forever.

The while loop requires relevant variables to be ready, in this example we need to define an indexing variable, i, which we set to 1.

The break Statement

With the break statement we can stop the loop even if the while condition is true:

In [5]:

```
# Example
# Exit the loop when i is 3:

i = 1
while i < 6:
    print(i)
    if i == 3:
        break
    i += 1
```

```
1
2
3
```

The continue Statement

With the continue statement we can stop the current iteration, and continue with the next:

In [4]:

```
# Example
# Continue to the next iteration if i is 3:

i = 0
while i < 6:
    i += 1
    if i == 3:
        continue
    print(i)
```

```
1
2
4
5
6
```

The else Statement

With the else statement we can run a block of code once when the condition no longer is true:

In [6]:

```
# Example
# Print a message once the condition is false:

i = 1
while i < 6:
    print(i)
    i += 1
else:
     print("i is no longer less than 6")
```

```
1
2
3
4
5
i is no longer less than 6
```

In [10]:

```
# Test Yourself With Exercises
# Exercise:
# Print i as long as i is less than 6.

i = 1
while i < 6 :

    print(i)
    i += 1
```

```
1
2
3
4
5
```