# Conditions and Loops

If you need Python to choose between two actions, then you can use an `if`/`else` statement. Try running this example code:

```
a = 42
if a < 10:
  print ('the number is less than 10')
else:
  print ('the number is greater or equal to 10')
```

Note the indentation and punctuation (especially the colons), because they are important.

If we leave out an `else`, then the program continues on. Try running this program with different initial values of `a` and `b`:

```
if a + b == 4:
  print ('printed when a + b equals four')
print ('always printed')
```

If you want to repeat an action several times, you can use a while loop. The following program prints `Hello` once, then adds 1 to the `greetings` counter. It then prints `Hello` twice because `greetings` is equal to 2, then adds 1 to `greetings`. After printing `Hello` three times, `greetings` becomes 4, and the `while` condition of `greetings <= 3` is no longer satisfied, so the program would continue past the `while` loop.

```
greetings = 1
while greetings <= 3:
  print ('Hello! ' * greetings)
  greetings = greetings + 1
```

Be careful! If you accidentally create an infinite loop, your program will freeze and you will have to abort it. Here's an example of an infinite loop. Make sure you see why it will never exit the `while` loop:

```
greetings = 1
while greetings <= 3:
  print ('Hello! ' * greetings)
  greetings = greetings + 0 # Bug here
```

If you want to carry out some action on every element of a list, the `for` loop will be handy

```
names = ['Alice', 'Bob', 'Charley']
for name in names:
  print ('Hello, ' + name)
```

And if you want to repeat an action exactly nn times, you can use the following template:

```
n = 10
for i in range(n):
  print (i)
```

In the above code, `range` is a function that creates a list of integers between 00 and nn, where nn is not included.
Finally, try seeing what the following code prints when you run it:

```
print (range(5, 12))
```