The `turtle` module

Python comes with a really nice `turtle` module, that lets a small turtle move around on a canvas and draw images.

```
from turtle import *
```

Try to run the following program:

```
from turtle import forward, left


forward(200)
left(90)
forward(100)
```

Try to change the value `90` to `45`. What did you expect to happen? And did it match your expectations?

Can you draw a square with the turtle?

Imagine you wanted to draw a nice beautiful star. This very quickly gets boring...

Imagine that you have a list that contains *moves* for the turtle:

```
moves = [100, 10, 100]
```

Now imagine that we take the list, and then extract one move at the time.

Run the following code and explain exactly what happens to your neighbor:

```
from turtle import forward, left


moves = [100, 10, 100]
for move in moves:
    forward(move)
    left(90)
```

`for` Loops

Lists are actually really good for looping. We use them in a particular kind of loop: the **for** loop:

In [ ]:

```
moves = [100, 10, 300, 200, 200]

for move in moves:
    print(move)
```

In [2]:

```
my_awesome_list = [0, 1, 2, 3]

for my_awesome_number in my_awesome_list:
    if my_awesome_number < 2:
        print(my_awesome_number)
```

```
0
1
```

That's smart and handy for small lists. What if we want to do it for long lists?

In [8]:

```
for i in range(4):
    print(i)
```

```
0
1
2
3
```

In [10]:

```
for i in range(10000):
    print(i)
```

```
0
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
```

`while` Loops

Where the `for` loop takes a collection of items, the `while` loop runs as long as, or while, a certain condition is `True`. Both the `for` loop and the `while` loop will still execute the code block --- if their respective conditions are met of course!

It has the syntax:

```
while boolean_expression:
    ...
```

In [ ]:

```
from turtle import forward, left


# move the turtle 10 times
number_of_moves = 10

while number_of_moves > 0:
    forward(100)
    left(45)
    number_of_moves -= 1
```

A computer program's main loop, such as the one of this notebook, is very likely implemented via a `while` loop.

In [ ]:

```
from turtle import forward, left


while True:
    forward(100)
    left(45)
```