Loops

There are two main ways to do loops. Important to notice here is *indentation*. Instead of brackets or other means to mark blocks of code, Python simply discerns these by indentation. Each level of indentation consists of **4 spaces**. This serves readability since you'll instantly see what block certain code belongs to.

In [1]:

```
a, b = 0, 1

while b < 10:
    print(b)
    a, b = b, a + b
```

```
1
1
2
3
5
8
```

While it's possible to do multiple assignments per row, this should be avoided for readability.

The second way to do loops is using the powerful "for" contruct. A basic example uses the `range` generator to generate a list of integers to loop over:

In [1]:

```
f = range(10, 20)

print(type(f))
print(*f)
print(*range(0, 5, 2))
print(*range(7, 1, -2))

# our squares list from last lesson
squares = [x**2 for x in range(15)]

for i in range(0, 5):
    print(i, ":", squares[i])
```

```
<class 'range'>
10 11 12 13 14 15 16 17 18 19
0 2 4
7 5 3
0 : 0
1 : 1
2 : 4
3 : 9
4 : 16
```

`for` iterates over all elements of a list:

In [3]:

```
for element in squares:
    print(element, end=", ")
```

```
0, 1, 4, 9, 16, 25, 36, 49, 64, 81, 100, 121, 144, 169, 196,
```