Functions

Usually, you need a certain functionality more often than once. This functionality can then be defined as a function. Let's use the Fibonacci numbers as an example:

In [4]:

```
def fibonacci(n):
    """Print the Fibonacci series up to argument n."""
    a, b = 0, 1

    while a < n:
        print(a, end=' ' )
        a, b = b, a + b


    print()

fibonacci(5)
fibonacci(81)
```

```
0 1 1 2 3
0 1 1 2 3 5 8 13 21 34 55
```

There's no check to see if the input `n` is smaller than zero. A better approach would be:

In [2]:

```
def fibonacci(n):
    """Print the Fibonacci series up to argument n."""
    if n > 0:
        a, b = 0, 1
        while a < n:
            print(a, end=' ')
            a, b = b, a + b
        print()
    else:
        print("Error: n must be larger than 0")

fibonacci(10)
fibonacci(-10)
```

```
0 1 1 2 3 5 8
Error: n must be larger than 0
```

Functions may also return some values:

In [4]:

```
def multiply(a, b):
    return a * b

print(multiply(5, 10))

result = multiply(7, 7)
print(result)
```

```
50
49
```

As an example, let's combine a few techniques here:

In [1]:

```
def fibonacciList(n):
    """Get the Fibonacci series up to argument n in a list."""
    if n > 0:
        result = []     # an empty list
        a, b = 0, 1

        while a < n:
            result.append(a)
            a, b = b, a + b

        return result
    else:
        print("Error: n must be larger than 0")
        return []

print(fibonacciList(20))

for fibonacciNumber in fibonacciList(20):
    print(fibonacciNumber, end=', ')
```

```
[0, 1, 1, 2, 3, 5, 8, 13]
0, 1, 1, 2, 3, 5, 8, 13,
```