Python Functions

A function is a block of code which only runs when it is called.

You can pass data, known as parameters, into a function.

A function can return data as a result.

creatig a funcion

In [ ]:
```
# In Python a function is created using the def keyword (from define):
def my_function():
  print("Hello from a function")
```

In [ ]:
```
my_function()
```

passing parameters

Data can be passed to functions as parameter.

Parameters are specified after the function name, inside the parentheses. You can add as many parameters as you want, just separate them with a comma.

In [ ]:
```
def greeting(name):
  print("Hello, " + name + '!')
```

In [ ]:
```
greeting('Askhat1')
greeting('Askhat2')
greeting('Askhat3')
```

default parameter values

sometimes it is necessary to provide default parameter to ease callign a function

In [ ]:
```
def my_function(country = "Norway"):
  print("I am from " + country)

my_function("Sweden")
my_function("India")
my_function()
my_function("Brazil")
```

Passing a List as a Parameter

You can send any data types of parameter to a function (string, number, list, dictionary etc.), and it will be treated as the same data type inside the function.

Attention! Understand what you are sending and how to work with it

In [ ]:
```
def my_function(food):
  for x in food:
    print(x)

fruits = ["apple", "banana", "cherry"]

my_function(fruits)
```

Return Values

To let a function return a value, use the `return` keyword:

In [ ]:
```
def my_function(x):
  return 5 * x

print(my_function(3))
print(my_function(5))
print(my_function(9))
```

Recursion

Python also accepts function recursion, which means a defined function can call itself.

Recursion is a common mathematical and programming concept. It means that a function calls itself. This has the benefit of meaning that you can loop through data to reach a result.

The developer should be very careful with recursion as it can be quite easy to slip into writing a function which never terminates, or one that uses excess amounts of memory or processor power. However, when written correctly recursion can be a very efficient and mathematically-elegant approach to programming.

In [ ]:
```
def recursion(k):
    if k == 0 :
        print ('zero')
    else:
        print (k)
        recursion(k-1)
```

In [ ]:
```
recursion (4)
```

Modules

Consider a module to be the same as a code library.

A file containing a set of functions you want to include in your application.

Create a Module

To create a module just save the code you want in a file with the file extension .py:

In [ ]:
```
# this is new file created from notebook browser - new - Text File
# inside my_module.py
def greeting(name):
  print("Hello, " + name)
```

Use a Module

Now we can use the module we just created, by using the import statement:

In [ ]:
```
# Import the module named mymodule, and call the greeting function:
import mymodule
mymodule.greeting("Jonathan")
```

In [ ]:
```

```