In [1]:
```
# FK5024 - TUTORIAL 10
# Python basics
```

In [2]:
```
a = 4
print(a)
```

```
4
```

In [3]:
```
a = a+5
```

In [4]:
```
a;
```

In [5]:
```
print(a)
```

```
9
```

In [6]:
```
print("Hello world!")
```

```
Hello world!
```

In [7]:
```
# FUNCTIONS
# Functions are very important because they avoid repetitions of the code,
# thus minimizing errors and making the code more efficient and more readable

a="Hello"
print (type(a))
```

```
<class 'str'>
```

In [8]:
```
print (type(a))
```

```
<class 'str'>
```

In [9]:
```
# Let's define our own function

def mySum (a,b):
    c = a + b;
    return c;

print (mySum (32,7))
```

```
39
```

In [10]:
```
# CONDITIONS
# Conditions are widely used in any computing language.
# They allow to evaluate different codes when different conditions are met

a = 5
b = 8
if a >= 0 :
    b = b + 20
    print("b =",b)
else :
    print("a is inferior to zero.")
```

```
b = 28
```

In [11]:
```
# SOME COMMON MATHEMATICAL OPERATORS
# <
# >
# >=
# <=
# == (equal)
# != (different)
```

In [12]:
```
# LOOPS
# They allow one to perform operations in sequence until a certain condition
is met

sentence = "Hello world !"
for letter in sentence:
    print (letter)
```

```
H
e
l
l
o

w
o
r
l
d

!
```

In [13]:
```
# You can combine the loops with the conditions

for letter in sentence:
    if letter not in "aeiouy":
        print (letter)
    else:
        print ('.')
```

```
H
.
l
l
.

w
.
r
l
d

!
```

In [14]:
```
# Libraries are files where different related functions are stored,
# and you can use them if you import the library.
```

In [15]:
```
a = 25
#print(sqrt(a)) # this will not work a priori
```

In [16]:
```
# To be able to calculate a square root,
# we need to import the mathematics module

a=25
import math
print(math.sqrt(a))
```

```
5.0
```

In [17]:
```
#help("math") # to see all the functions available
```

In [18]:
```
help("math.sqrt")
```

```
Help on built-in function sqrt in math:

math.sqrt = sqrt(...)
    sqrt(x)

    Return the square root of x.
```

In [19]:
```
# You can also import only the functions you need
# from a module instead of importing

from math import sqrt
```

In [20]:
```
# You can import * which will spare you the "math." before the functions

from math import *
print (sqrt(a))
```

```
5.0
```

In [21]:
```
# Now we will introduce two different libraries: NUMPY and MATPLOTLIB

# NUMPY allows you to use arrays, whereas MATPLOTLIB allows you to plot
```

In [22]:
```
import numpy as np
```

In [23]:
```
# We can now define a vector...

a = np.array([1,4,6,2])
print(a)
```

```
[1 4 6 2]
```

In [24]:
```
# ...and a matrix

A = np.array([[1,1],[4,2],[9,3]])
print(A)
```

```
[[1 1]
 [4 2]
 [9 3]]
```

In [25]:
```
# You can also access the numbers of rows and columns in
# your matrix, with the function "shape" in the object A

A.shape
```

Out[25]:
```
(3, 2)
```

In [26]:
```
# The total number of element is given by the function "size"

A.size
```

Out[26]: 6

In [27]:
```
# Once you have a vector or a matrix, you can easily access
# every single element in them. Note that the first row and
# column are labeled with 0, not with 1

A[0,0]
```

Out[27]: 1

In [28]:
```
A[2,1] # the first number refers to the row,
# the second number to the column
```

Out[28]: 3

In [29]:
```
print(A[2,:])
```

```
[9 3]
```

In [30]:
```
# You can use loops here too

for row in A:
    print(row)
```

```
[1 1]
[4 2]
[9 3]
```

In [31]:
```
for i in range(len(A)):
    print(A[i])
```

```
[1 1]
[4 2]
[9 3]
```

In [32]:
```
# Fill an array:

B = np.ma.zeros(10)
print(B)
```

```
[0. 0. 0. 0. 0. 0. 0. 0. 0. 0.]
```

In [33]:
```
range(0,10)
```

Out[33]: range(0, 10)

In [34]:
```
# Note that the indentation in python is extremely important

for i in range(10):
    B[i]=i**3
print (B)
```

```
[  0.   1.   8.  27.  64. 125. 216. 343. 512. 729.]
```

In [35]:
```
# Operations on matrices

B = B*2
print(B)
```

```
[   0.    2.   16.   54.  128.  250.  432.  686. 1024. 1458.]
```

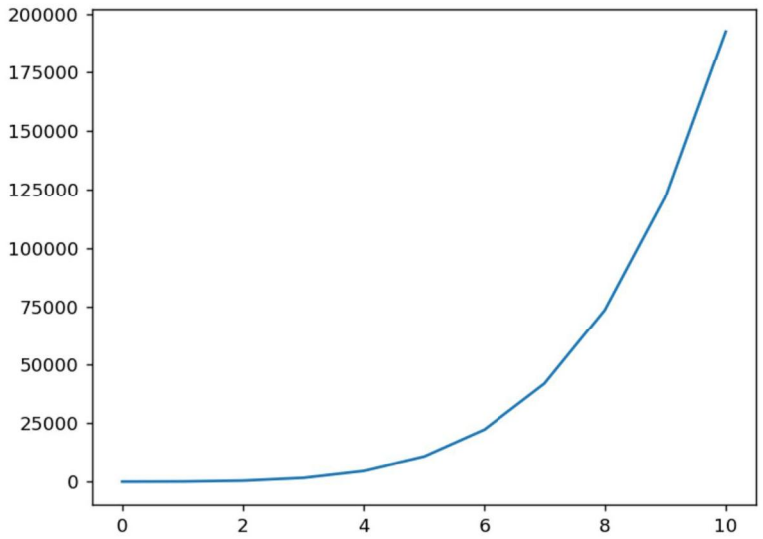
In [36]:
```
# Let's go now to MATPLOTLIB. You use it to make plots

import matplotlib.pyplot as plt
```

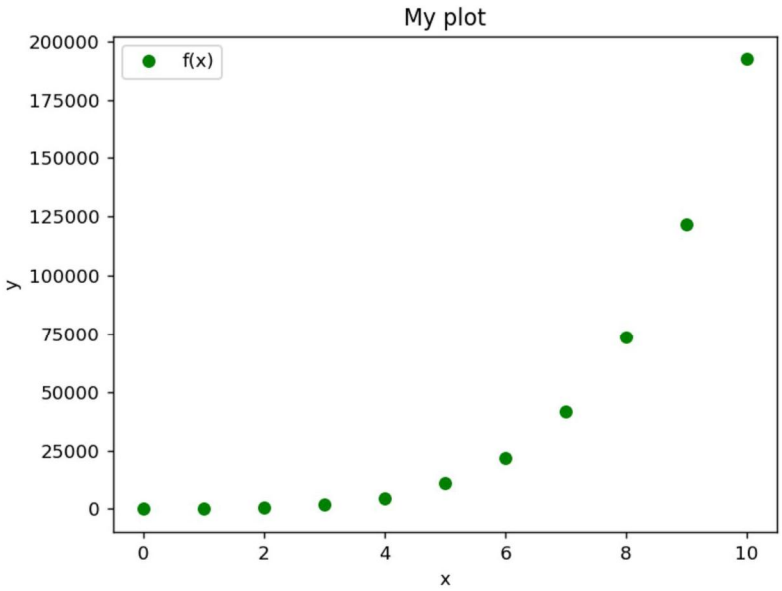
In [37]:
```
x = range(5)
f = np.ma.zeros(len(x))
for i in range(len(x)):
    f[i]=i**2
print('x: ',x)
print('f: ',f)
```

```
x:  range(0, 5)
f:  [ 0.  1.  4.  9. 16.]
```

In [67]:
```
%matplotlib nbagg
# the command above is just to make the plot appear below,
# not in another window
plt.plot(x,f)
plt.show()
```



In [66]:
```
%matplotlib nbagg
plt.plot(x,f,"o",color="green",label="f(x)")
plt.xlabel('x')
plt.ylabel('y')
plt.title('My plot')
plt.legend()
plt.show()
```



In [40]:
```
# And many, many other possibilities!
```

In [41]:
```
# EXERCISES
# (1) Create a null vector of size 10 but where the fifth value is 1.
# (2) Write a function which can compute the factorial of a given number.
# (3) Write a function that will calculate (a + b)^n. Then plot the result fo
r a=1.4, n=5, and b in [0;10].
# (4) Read the data file exercise4_stars.txt and plot the luminosity as a fun
ction of temperature.
# (5) Read the data file FIRAS_CIB_spectrum.dat and plot the monopole spectru
m of the CMB.
#     (5b) Fit the CMB specturm with a fourth-order polynomial.
```

In [42]:
```
# (1)

Z = np.zeros(5)
Z[4] = 1
print(Z)
```

```
[0. 0. 0. 0. 1.]
```