**Lab 1 - Python introduction**

**Exercise 0 - Example**

Assign the number 31 to a new variable, q. Write an expression that raises q to the 4th power

In [67]:

*# Your answer here*

q = 31

q\*\*4

Out[67]:

923521

**Exercise 1**

Create arbitrary list, tuple and dictionary and print them

In [68]:

*# Your answer here*

* list\_1 = ['Data','Science','4']
* print(list\_1)
* tuple\_1 = (16, 4, 2021)
* print(tuple\_1)
* dict\_1 = {'students': ['Alona', 'Kateryna', 'Anna']}
* print(dict\_1)

**Exercise 2**

Print the next elements of array s: 10th, from 10 to 20, all elements more than 0

In [69]:

**import** **numpy** **as** **np**

s = np.random.normal(0,1,100)

*# Your answer here*

* print(s[9], s[9:19])
* for i in range (100):
* if s[i] > 0:
* print(s[i])

**Exercise 3**

Make a function pow that raised base number to a fixed power power. If power >= 50 print message "power is too big" and return -1

In [70]:

*# Your answer here*

def pow (number, power):

if power >= 50:

print("power is too big")

return -1

else:

return number\*\*power

**Exercise 4**

Make a function above(x,n) which returns all elements x > n. By default x = 10. Use numpy arrays

In [71]:

**import** **numpy** **as** **np**

*# Your answer here*

**Exercise 5**

Write a function my\_ifelse(x, exp, n) which test array x with expression exp to number n. For instance, my\_ifelse(x,'>=',10) returns all elements of array x which greater or equal 10. exp can be <, >, <=, >=, ==. If exp dont match these expression array x is retured

In [72]:

**import numpy as np**

**x = np.random.normal(0,1,100)**

**s = []**

**def my\_ifelse(x, exp, n):**

**if exp == '==':**

**for i in range(100):**

**if x[i] == n:**

**s.append(x[i])**

**return s**

**elif exp == '<':**

**for i in range(100):**

**if x[i] < n:**

**s.append(x[i])**

**return s**

**elif exp == '>':**

**for i in range(100):**

**if x[i] > n:**

**s.append(x[i])**

**return s**

**elif exp == '>=':**

**for i in range(100):**

**if x[i] >= n:**

**s.append(x[i])**

**return s**

**elif exp == '<=':**

**for i in range(100):**

**if x[i] <= n:**

**s.append(x[i])**

**return s**

**else:**

**return x**

**Exercise 6**

Make a function called counter that takes a string input\_string and returns a dictionary of letter counts.

In [73]:

def counter (input\_string):

letters = {}

input\_string = list(input\_string)

for letter in input\_string:

if letter.isalpha():

if letter in letters:

letters[letter] += 1

else:

letters[letter] = 1

return letters

**Exercise 7**

From the function counter make function counter\_sorted with second boolean argumens reverse that returns sorted letter counts. Order depends of the reverse argument. By default reverse = False

In [74]:

def counter\_sorted(input\_string, reverse = False):

letters ={}

input\_string = input\_string.lower()

input\_string = list(input\_string)

for letter in input\_string:

if letter.isalpha():

if letter in letters:

letters[letter] += 1

else:

letters[letter] = 1

sorted\_letters = list(letters.keys())

sorted\_letters.sort()

if reverse:

sorted\_dict = {sorted\_letters[len(sorted\_letters) - (i+1)] : letters.get(sorted\_letters[i]) for i in range (len(sorted\_letters))}

else:

sorted\_dict = {sorted\_letters[i] : letters.get(sorted\_letters[i]) for i in range (len(sorted\_letters))}

return sorted\_dict