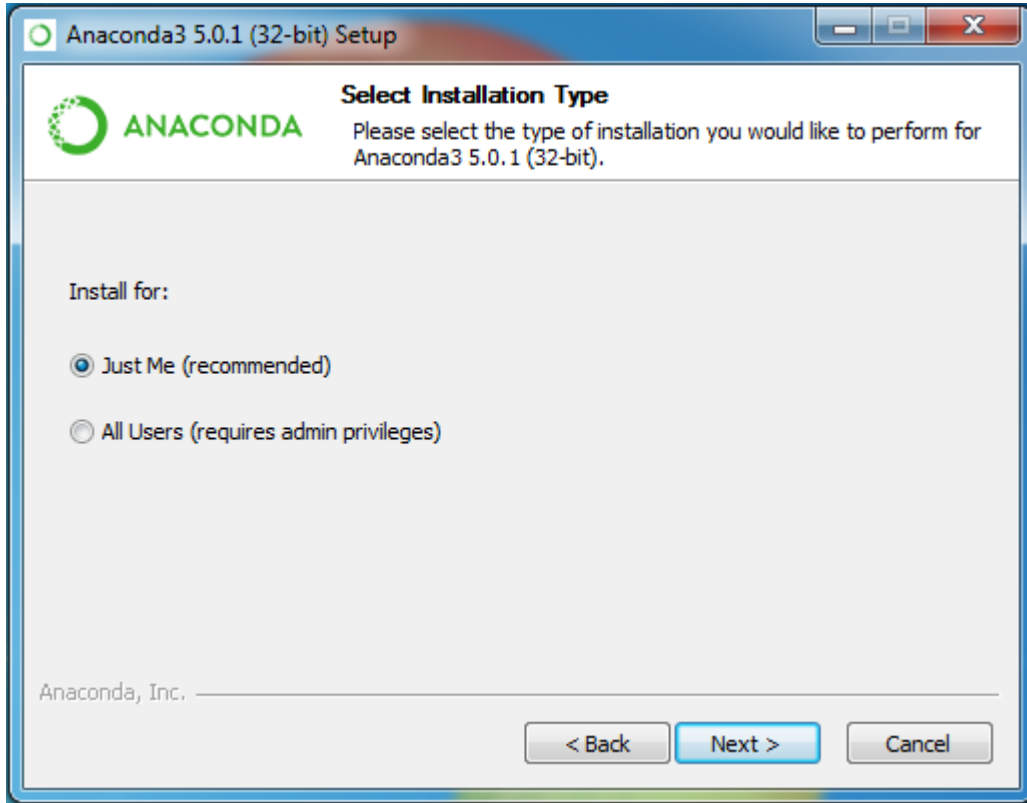


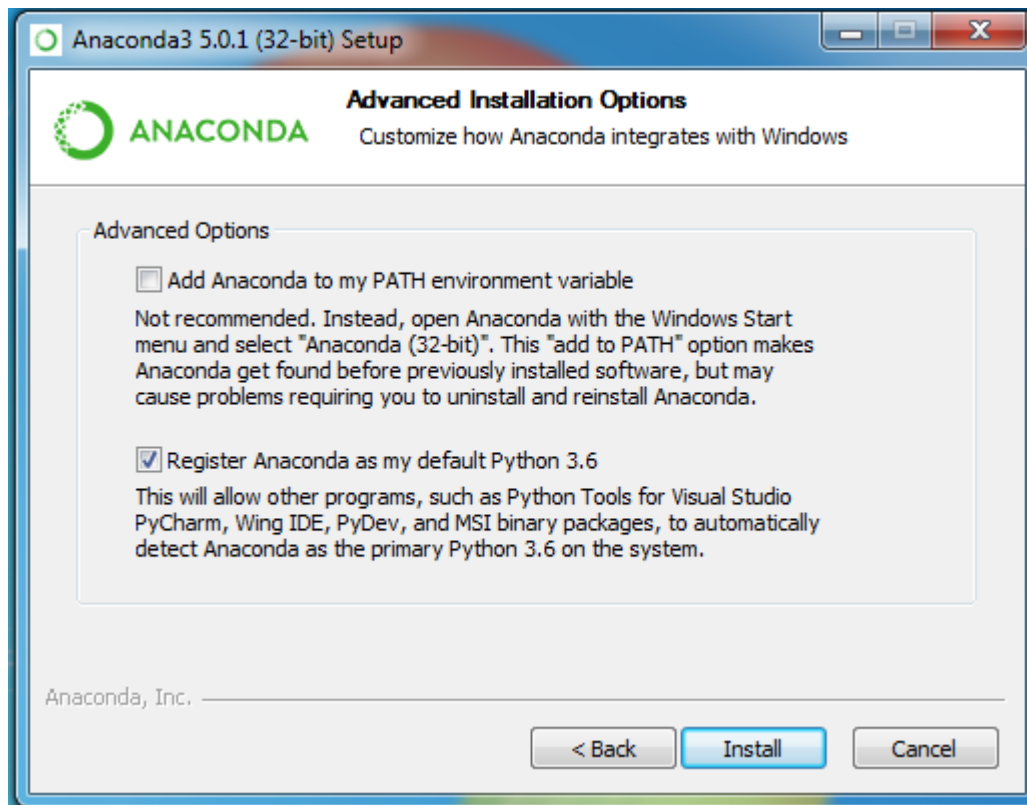
# Laborator 1

## Instalare Anaconda

1. Se descarca Anaconda (<https://www.anaconda.com/download/>); folositi versiunea pe 64 de biti daca sistemul de operare este de acelasi tip.
2. Pe parcursul instalarii se alege "Install just for me" - nu neceista drepturi de adminstrare pe calculatorul respectiv



3. Pentru destination folder alegeti o unitate si un director pentru care aveti drepturi de scriere, de ex. D:\lucru
4. Bifa de "Add Anaconda to my path environment variable" sa fie debifata, cea de "Register Anaconda..." bifata:



5. Actualizarea pachetelor se face in doi pasi:

## Setare mediu virtual, instalare de pachete

In Anaconda se pot defini medii virtuale (virtual environments). Avantajul acestora este ca se permite ca intr-un mediu sa fie instalate pachete cu dependinte specifice fara a interfera cu pachetele din alte medii.

Implicit, la instalarea Anaconda se creeaza un mediu base. Se pot crea medii virtuale urmand pasii de mai jos.

1. Creati un mediu virtual (virtual environment) de data-science cu comanda:

```
conda create --name data-science python=3.6 anaconda --yes
```

Trecerea in mediul virtual se face cu:

```
activate data-science
```

iar iesirea cu:

```
deactivate
```

Lista mediilor virtuale se obtine prin:

```
conda info --envs
```

2. Instalati pachete auxiliare pentru Jupyter notebook:

```
conda install -c conda-forge jupyter_contrib_nbextensions  
jupyter contrib nbextension install --user
```

3. Se pot instala si pachete care nu se gasesc in canalul oficial de conda, folosind comanda pip. Orice distributie de Python are acces la pachete prin acest manager:

```
pip install librosa
```

sau pentru upgrade al unui pachet instalat:

```
pip install keras --upgrade
```

Lista tuturor pachetelor instalate in distributia Anaconda, in mediul curent, se obtine cu comanda

```
conda list
```

Cele in dreptul carora apare in ultima coloana specificatia <pip> se pot actualiza doar prin package manager-ul pip; celelalte - folosind utilitarul conda.

## Manifestul Python

Lansati interpretorul interactiv Python in linia de comanda: `ipython` si apoi rulati comanda: `import this`. Se va afisa pe ecran manifestul Python the Zen of Python (<https://www.python.org/dev/peps/pep-0020/>).

## Ciclari, siruri de caractere

Se recomanda ca urmatoarele exercitii sa le lucrari in Jupyter notebook. In linia de comanda (command prompt sau PowerShell) scrieti comanda:

```
jupyter notebook
```

Se va deschide automa browserul implicit la adresa localhost:8888 (daca portul 8888 nu e liber, se va cauta automat un alt port).

1. (fizz-buzz test) Sa se scrie numerele de la 1 la  $n$ ; pentru fiecare multiplu de 3 se va scrie in locul numarului 'Fizz', pentru multiplu al lui 5 se va scrie 'Buzz'; daca numarul este multiplu de 15 se va scrie in locul lui 'FizzBuzz'.
2. Sa se verifice conjectura Collatz ([https://en.wikipedia.org/wiki/Collatz\\_conjecture](https://en.wikipedia.org/wiki/Collatz_conjecture)) pentru numerele intre 1 si 1000.
3. \* Sa se creeze o functie care preia un numar  $n$  si returneaza un alt numar pe baza cifrelor lui  $n$ , astfel: se calculeaza numarul cifrelor pare din  $n$  (fie el si 0), numarul de cifre impare (poate fi si 0), suma lor si se formeaza numarul din acestea 3; daca exista vreun zero nesemnificativ, acesta se va ignora. Numarul obtinut se va supune aceleiasi transformari. Exemplu: 3->11->22->202->303->123->123->123... Verificati ca dupa un numar finit de transformari se ajunge la numarul 123; faceti aceste verificari pentru numerele din intervalul 1, 1000.
4. \* Se pleaca de la un numar intreg. Fiecare cifra a sa se scrie cu litere, in limba engleza (de exemplu 5->five). Claculati numarul total de caractere rezultate, iar pentru numarul obtinut repetati procedura. Verificati pentru numerele de la 1 la  $n$  ca se obtine intr-un numar finit de pasi numarul 4. Exemplu: 123->onetwothree->11->oneone->6->six->3->three->5->five->4->four->4->four....
5. \* Este rezultatul de mai sus valabil si pentru transcriere in limba romana?
6. \* Se pleaca de la un numar  $n$ ; se scriu toti divizorii sai, inclusiv 1 si  $n$ ; se aduna *cifrele* tuturor acestor divizori; pentru numarul obtinut se aplica acelasi procedeu. Verificati ca procesul se stabilizeaza in numarul 15. Exemplu: 20->1. 2. 4. 5. 10. 20 -> suma cifrelor: 15->1. 3. 5. 15-> suma cifrelor: 15...

## Liste, dictionare, multimi

1. Dandu-se o lista, sa se inverseze ordinea elementelor. Exista doua variante de rezolvare: slicing si folosind o metoda a tipului lista.
2. Pentru o lista de cuvinte (stringuri), sa se obtina un dictionar care are drept cheie cuvintele distincte din lista, iar valorile corespunzatoare sa fie numarul de aparitii ale fiecarui cuvint
3. Dandu-se o lista de tuple binare, sa se obtina o lista care contine valoarea maxima a fiecarui tuplu.
4. Sa se calculeze suma numerelor de indice par si produsul numerelor de indice impar dintr-o lista.
5. Documentati-va asupra tipului de date Set (multime) si folositi-l pentru a determina elementele unice dintr-o lista de numere.
6. Pentru o lista de stringuri si un numar  $n > 1$ , sa se produca o lista care contine toate stringurile din lista initiala cu toate numere 1..n. Exemplu: pentru lista = ['a', 'b', 'c'],  $n=2$  se va produce lista\_2 = ['a1', 'a2', 'b1', 'b2', 'c1', 'c2']; ordinea poate sa difere. Incercati o implementare bazata pe list comprehension.
7. Dandu-se o lista de cuvinte, sa se produca o alta lista cu aceleasi cuvinte, dar: prima litera a fiecarui cuvint sa fie mare, celelalte din cuvint - mici.
8. Documentati-va si explicati comportamentul pentru urmatoarea secventa de cod:

```
this = ['I', 'am', 'not', 'a', 'crook']
that = ['I', 'am', 'not', 'a', 'crook']
print("Test 1:", this is that)
that = this
print("Test 2:", this is that)
```

## Funcții

1. Pentru o lista de numere, sa se scrie o functie care returneaza suma elementelor si diferenta maxima dintre ele.
2. Scrieti o functie care preia doua liste si returneaza True daca cele doua liste contin cel putin k elemente comune (k dat ca parametru, intreg  $\geq 1$ , valoare implicita 1), False altfel.
3. Sa se scrie o functie care determina daca o lista contine doar cuvinte. Functia trebuie sa testeze daca elementele din lista sunt stringuri; un string este cuvint daca nu contine: spatiu, virgula si alti separatori.
4. Sa se scrie o functie care returneaza numarul de litere mici si mari - 2 rezultate - dintr-un parametru dat.
5. Sa se scrie o functie care primind o lista de dictionare, returneaza True daca toate dictionarele sunt goale si False altfel.
6. Sa se scrie o functie recursiva care sa faca ridicarea la putere a unui numar, astfel:

$$a^n = \begin{cases} (a^{n/2})^2 & \text{daca } n \text{ e par} \\ (a^{n/2})^2 \cdot a & \text{daca } n \text{ e impar} \end{cases}$$

unde prin  $n//2$  se reprezinta catul impartirii intregi intre  $n$  si 2.