

Laborator 1 - Introducere in Python

Potrivit companiei TIOBE, Python este al treilea cel mai popular limbaj de programare al anului 2018 dupa Java si C/C++ [1].

Care este istoria acestui limbaj de programare? [1] Fondatorul acestui limbaj de programare este danezul Guido Van Rossum. El a inceput sa lucreze la acest limbaj de programare in timpul lui liber in iarna anului 1989. Numele este inspirat din serialul BBC "The Monty Python's Flying Circus".

Python este un limbaj potrivit pentru programare orientata pe obiecte si pentru programare functionala.

In Python nu trebuie sa declaram variabile. Codul din Python este semnificativ mai scurt decat cel din C++/Java [1]. Python este un limbaj de scripting, ceea ce inseamna ca este **interpretat** si nu **compilat**, economisind mult timp in procesul de dezvoltare si debugging.

În funcție de modul de execuție al programelor, limbajele de programare se împart în două categorii [3]:

- **interpretate:** instrucțiunile sunt citite linie cu linie de un program numit interpretor și traduse în instrucțiuni mașină; *avantaje:* simplitate + se scurteaza timpul de dezvoltare al programului deoarece nu este nevoie de compilare si de "linking"; *dezavantaj:* viteza de execuție redusă;
- **compilate:** codul sursă al programelor este transformat de compilator într-un cod ce poate fi executat direct de procesor; *avantaj:* execuție rapidă; *dezavantaj:* lipsa portabilității, codul compilat (într-un format de nivel scăzut) nu poate fi rulat decât pe platforma pe care a fost compilat.

Pe langa distributia standard de CPython, exista pachete modificate care includ functionalitati suplimentare. Noi vom folosi "Anaconda", care include module stiintifice populare (cum ar fi numpy, scipy si pandas) si manager-ul de pachete conda.

Python este o alegere buna in domeniul analizei numerice si al calculului stiintific.

Principalul avantaj este ca Python este gratuit si "open source". Python poate fi utilizat in sistemele de operare Windows, Mac OS X si Unix.

Python permite verificarea erorilor mai mult decat limbajul C. Python este mai util pentru programele mari decat script-urile shell si fisierele batch.

[Shell-ul este principala interfata de comunicare între utilizator si sistemul de operare. Desi, în mod intuitiv, shell-ul este identificat cu o interfata în linia de comanda, poate fi si o interfata grafica. Exemplu este Explorer-ul sistemului de operare Windows.]

*Sistemele de operare Unix ofera o interfata de tip CLI (Command Line Interface). Desi cu o curba de învățare mai mare decat o interfata grafica, CLI permite un control mult mai bun al sistemului. Mai mult, shell-ul dispune de un limbaj de programare. Un program shell, denumit **script shell**, este folosit pentru a îmbina mai multe comenzi si diverse structuri de control pentru a obtine o noua functionalitate sau pentru automatizarea sarcinilor. În acest fel un script shell este un instrument esential pentru sarcinile administrative si alte rutine repetitive care nu necesita functionalitati ale unor limbaje de programare avansate.*

Un shell cunoscut pe sisteme Unix este Bash (Bourne Again SHell). Microsoft ofera PowerShell pe sistemele Windows. PowerShell are o abordare orientata pe obiecte si un set de functionalitati care acopera nevoile de administrare ale unui sistem Windows.

Fişierele batch (batch files) reprezintă seturi de comenzi MS – DOS care sunt executate succesiv, grupate în fişiere cu extensia **.bat**, cu un rol important în reducerea timpului consumat de task-urile repetitive. [6]]

Python este util pentru automatizari, de exemplu, daca dorim sa cautam un cuvant si sa il inlocuim cu un alt cuvant intr-un numar foarte mare de fisiere text [4].

Python este extensibil.

Programele scrise in Python sunt mai scurte decat cele din C, C++ sau Java din mai multe motive [4]:

- tipurile de date "high-level" ne permit sa exprimam operatii complexe intr-o singura instructiune
- gruparea instructiunilor se realizeaza prin indentare in loc de acolade care sa marcheze inceputul si sfarsitul instructiunii
- nu sunt necesare declaratii de variabile si de argumente

Utilizare

Se instaleaza "Anaconda Python Distribution" pe Windows, Anaconda 5.2, versiunea Python 3.6.

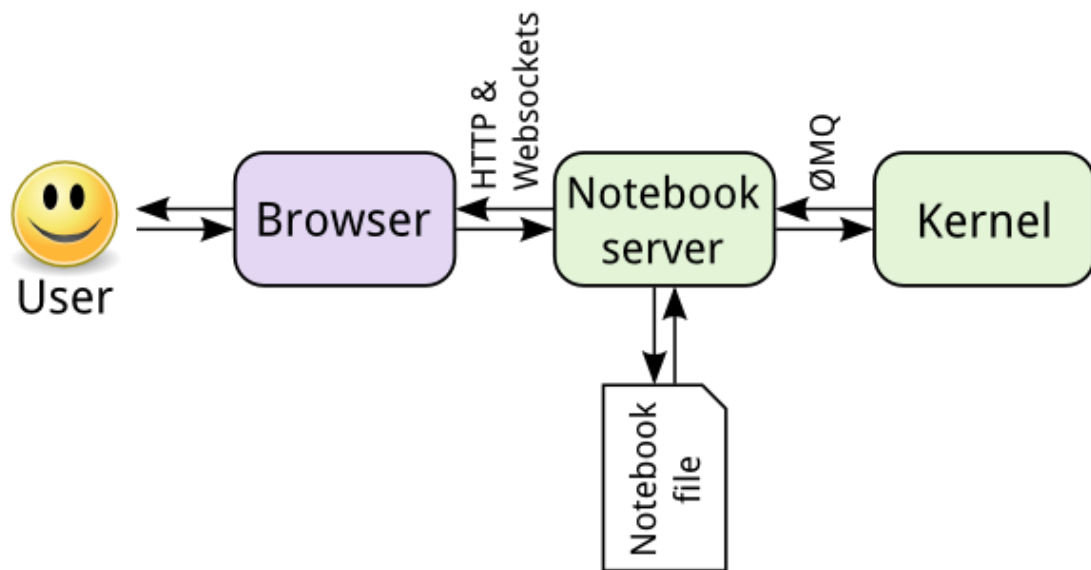
Se deschide Anaconda Navigator - Jupyter Notebook.

Se creeaza un director nou cu New - Folder si un notebook nou cu New - Python 3.

(Uniform Resource Locator, adresa web) URL-ul aplicatiei web este in mod implicit <http://localhost:8888> (acest calculator, portul 8888). Se deschide web browser-ul implicit la aceasta adresa. In mod implicit, server-ul notebook-ului porneste la portul 8888. Daca portul 8888 nu este disponibil sau este deja utilizat, atunci server-ul notebook-ului cauta urmatorul port disponibil. [8]

La deschiderea notebook-ului in navigatorul web, apare asa numitul Notebook Dashboard, care cuprinde o lista de notebook-uri, fisiere si subdirectoare in directorul in care a fost pornit serverul notebook-ului. Serverul notebook-ului se porneste in directorul care contine toate notebook-urile. Notebook-ul este de fapt o interfata (este partea de frontend).

Se selecteaza codul care vrem sa fie executat. Se apasa Shift+Enter sau butonul de Run din partea de sus a paginii.



Folosind Jupyter, se ruleaza codul, se stocheaza codul si rezultatul intr-un document editabil numit notebook. [9] Cand salvam notebook-ul, acesta este trimis de la navigatorul web la server-ul notebook-ului, care il salveaza pe un disc ca fisier JSON cu extensia .ipynb.

[JSON (JavaScript Object Notation) este un format pentru schimb de date. Este usor de citit si de scris de catre oameni. Este usor de analizat si de generat de catre calculatoare. Se bazeaza pe limbajul de programare JavaScript.

JSON este construit pe doua structuri:

- o colectie de perechi de nume si valori; in limbaje de programare variate, astfel de perechi se pot realiza cu ajutorul obiectelor, structurilor, dictionarelor, tabelor hash, listelor cu cheie;
- o lista ordonata de valori; in majoritatea limbajelor de programare, astfel de liste ordonate se realizeaza cu ajutorul unui array, vector sau unei liste;

Acestea sunt structuri de date universale. Toate limbajele de programare moderne le cuprind intr-o forma sau alta. De aceea, un format de date care poate fi interschimbabil intre limbaje de programare ar trebui sa fie bazat pe aceste structuri.] [10]

Interfata Notebook-ului utilizeaza kernel-ul IPython. Kernel-ul IPython este un proces separat care este responsabil pentru rularea codului utilizatorului. Partea de front-end comunica cu kernel-ul IPython folosind mesaje JSON.