



Școala
informală
de IT



Python Development

Week 1. Introducere



Școala
informală
de IT



Let's get to know each other

- NUME, PRENUME
- DIN CE ORAȘ EȘTI?
- ÎN CE ORAȘ ÎNVEȚI LA FACULTATE?
- LA CE FACULTATE?
 - când ai programată sesiunea?
- DE CE AI ALES CURSUL DE PYTHON?
 - de ce Python și nu altceva?
 - ce vrei să dezvolti folosind Python?
- CE HOBBY-URI AI / ÎȚI PLACE SĂ FACI ÎN TIMPUL LIBER?



Cuprins

1. Cum va decurge cursul?
2. Ce vom învăța?
3. Ce este Python?
4. De ce Python?
5. PyCharm
6. Git & GitHub
7. Instalare tool-uri
 - a. Python
 - b. PyCharm



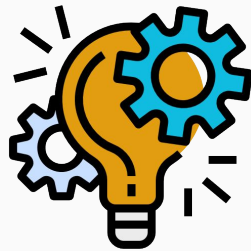
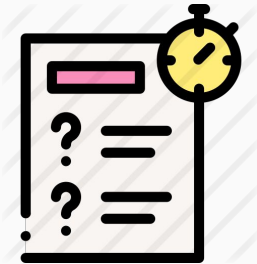
Cum va decurge cursul?

Week 1. Introducere



Cum va decurge cursul?

- Program:
 - cursul se va desfășura pe parcursul a **10 săptămâni**.
 - în fiecare săptămână va avea loc un curs a câte **3 ore**, însumând un total de **30 de ore**.
 - întâlnirea săptămânală va avea loc **joia** la **ora 18:30**.
 - data estimativă de final - **12 august 2021**.
- Fiecare cursant trebuie să aloce un număr de ore suplimentare pentru fixarea noțiunilor învățate.
- Vor exista teme de casă.
- Fiecare student va avea de prezentat un proiect individual.



Cum va decurge cursul?

- Planificare cursuri:
 - 6 Mai - **Curs 1** - Introducere
 - 13 Mai - **Curs 2** - Pythonic “Hello, World!”
 - 20 Mai - **Curs 3** - Ifs, loops and functions.
 - *Sesiune ușoară!*
 - 24 Iunie - **Curs 4** - Memory savers, files & web scraping
 - 1 Iulie - **Curs 5** - Object Oriented Programming
 - 8 Iulie - **Curs 6** - Desktop Apps (Tkinter + MySQL)
 - 15 Iulie - **Curs 7** - Web Apps (Django + MySQL)
 - 22 Iulie - **Curs 8** - Django admin
 - 29 Iulie - **Curs 9** - Unit tests
 - *Pregătire proiect final*
 - 12 August - **Curs 10** - Prezentare proiect final



Ce vom învăța?

Week 1. Introducere



Ce vom învăța?

- Vom învăța principiile de bază ale programării.
- Vom învăța cum să scriem și să rulăm un program dezvoltat în Python.
- Vom crea baza standardelor ce trebuie urmărite în scrierea codului (**PEP 8**)
- Vom lucra la un proiect individual ce va fi prezentat în cadrul ultimului curs:
 - va fi un proiect pornit de la 0.
 - va fi un proiect ce va reprezenta baza noastră ca Python developer.



Ce este Python?

Week 1. Introducere



Ce este Python?

- Python este un limbaj de programare creat în anul 1991 de către Guido van Rossum.
- Este un limbaj interpretabil, nu unul compilabil. Instrucțiunile programului sunt executate direct, fără o compilare anterioară în limbaj cod-mașină.
- Este un limbaj de nivel înalt. Oferă o abstractizare față de detaliile calculatorului:
 - folosește elemente din limbajul natural pentru scrierea codului
 - simplifică procesul de dezvoltare al aplicațiilor prin automatizarea sau înlăturarea nevoii de gestionare a diferitelor necesități ale calculatorului (ex: gestionarea memoriei).
- Este un limbaj care își gestionează singur memoria. Memoria Python este de tip heap și conține toate obiectele și structurile de date. Gestionarea acesteia este asigurată intern de către **Python Memory Management**.
- Releases:
 - Python 2 a fost lansat în anul 2000. În anul 2020 s-a oprit oficial update-ul acestei versiuni.
 - **Python 3** a fost lansat în 2008. A reprezentat o revizie majoră a limbajului, astfel că multe programe dezvoltate în Python 2 au necesitat modificări astfel încât să poată funcționa folosind Python 3.

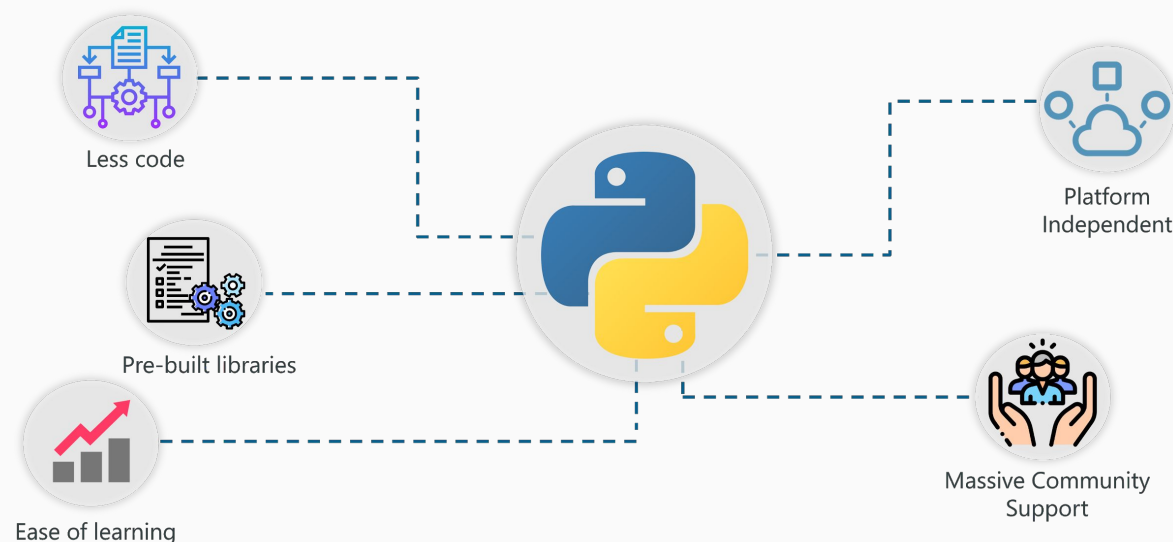


De ce Python?

Week 1. Introducere

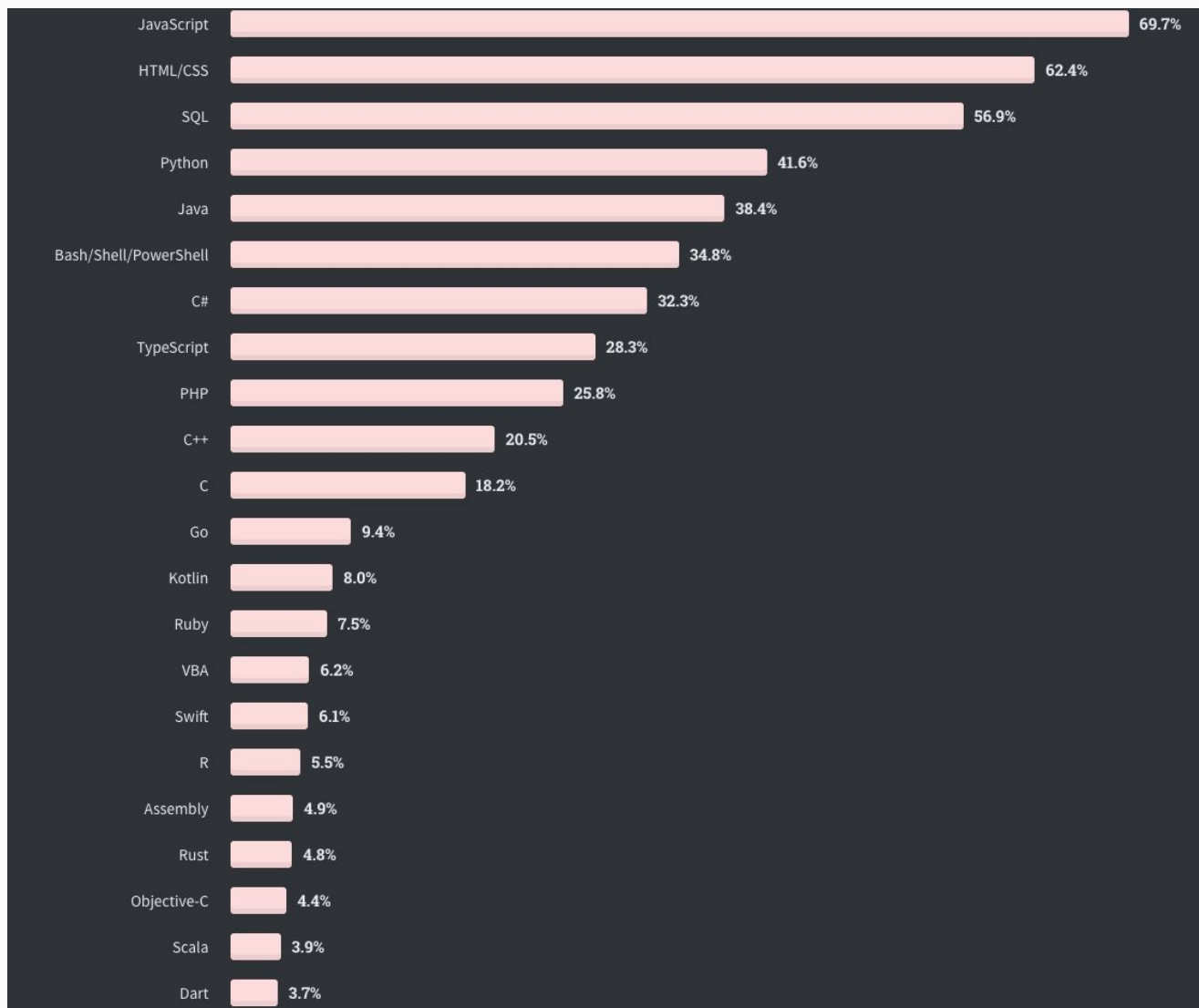


De ce Python?



- Este un limbaj **ușor de învățat** datorită sintaxei ușor de înțeles și a utilizării cuvintelor din limbajul natural.
- Este un limbaj **independent de sistemul de operare**. Poate fi folosit pe oricare dintre marile sisteme de operare: Windows, Linux sau macOS.
- Permite **multiple paradigme de programare**. Permite atât programare funcțională, cât și orientată pe obiecte.
- Există foarte multe librării ce pot fi folosite direct.
- Comunitatea de development este foarte mare.

De ce Python?



- Este limbajul de programare cu cea mai mare creștere în ultimii ani.
- Această creștere se datorează multiplelor domenii în care poate fi folosit:
 - desktop applications
 - web development
 - IoT (Internet of Things)
 - machine learning
 - data analysis
 - AI (Artificial Intelligence)



De ce Python?



Companii de top au ales Python. **Tu de ce nu ai face-o?**



PyCharm

Week 1. Introducere



PyCharm

- **PyCharm** este un IDE (Integrated Development Environment) dezvoltat de compania cehă **JetBrains**.
- Este folosit cu precădere în dezvoltarea aplicațiilor Python.
- Oferă:
 - analiza codului
 - debugger cu interfață grafică
 - testare unitară integrată
 - integrare cu version control systems (VCSes)
 - suport pentru dezvoltarea aplicațiilor web folosind framework-ul Django (not-free).
- Este un tool disponibil pe oricare dintre marile sisteme de operare: Windows, Linux și macOS.
- Este disponibil în două versiuni:
 - Community Edition - licență gratuită
 - Professional Edition - licență contra cost.



Git & GitHub

Week 1. Introducere



Git & GitHub

- **Git** este un sistem distribuit de versionare ce are rolul să gestioneze un proiect de orice dimensiune cu viteză și eficiență sporită.
- Funcționalitatea ce face Git-ul să depășească în popularitate aproape orice alt SCM (Source Code Management) existent este **modelul de branching**.
 - alte SCM-uri existente: Subversion, CVS, Perforce, ClearCase ș.a.
- Git-ul permite folosirea multiplelor branch-uri ce pot fi independente.
- Din moment ce cam toate operațiile sunt făcute local, oferă un mare avantaj din punct de vedere al vitezei.
- **GitHub**-ul este o platformă ce permite găzduirea unui manager de versiuni și colaborarea între developeri.



Git & GitHub

- Terminologie:
 - **repository** - în teorie este o colecție de fișiere și istoria schimbărilor suferite de acestea; în practică reprezintă proiectul nostru și toate fișierele aferente acestuia.
 - **clone** - este o comandă cu ajutorul căreia putem clona local un repository.
 - **branch** - reprezintă o versiune a proiectului care diverge din proiectul principal.
 - **checkout** - reprezintă schimbarea branch-urilor (mutarea de pe un branch pe altul).
 - **add** - acțiunea de a adăuga fișierele într-o zona intermediară numită Staging Area.
 - **commit** - acțiunea de a înregistra modificările făcute într-un repository.
 - **push** - acțiunea de a uploada versiunea locală pe server.
 - **merge** - reprezintă procesul de îmbinare a două versiuni diferite.
 - **pull** - acțiunea de a obține de la server ultimele modificări ale datelor.
 - **pull request** - reprezintă procesul de code review și merge al unui branch.



Instalare tool-uri

Week 1. Introducere



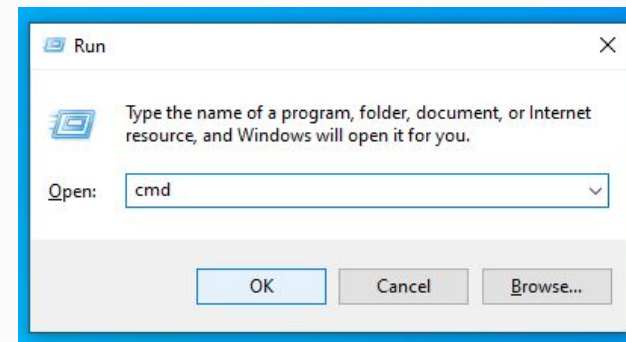
Instalare tool-uri - Python

- Folosind acest ghid puteți instala tool-urile necesare primilor pași în dezvoltarea programelor folosind limbajul Python.

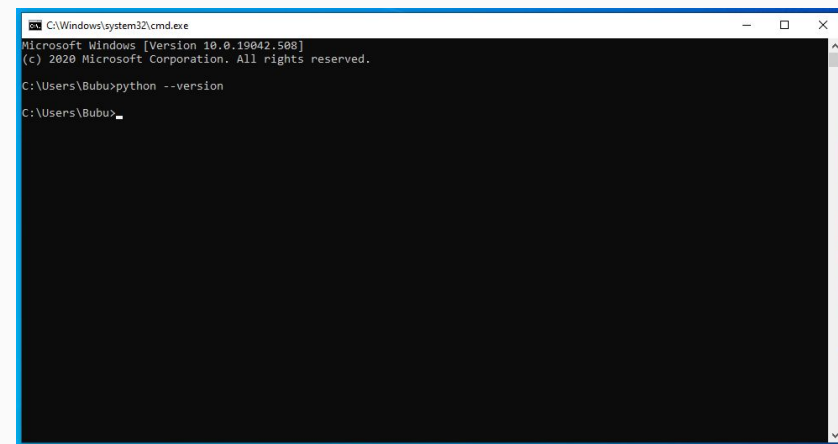
1. Deschideți un Command Line. Pentru a face asta folosiți combinația



pentru a deschide fereastra din imagine. Tastați **cmd** și dați **OK**.



2. În command line-ul deschis folosiți comanda **python --version** pentru a verifica dacă aveți instalat Python și ce versiune este instalată. Unele sisteme Windows vin cu o variantă default de Python. Indicat ar fi să o dezinstalați pe aceea și să instalați una fresh conform acestui ghid.



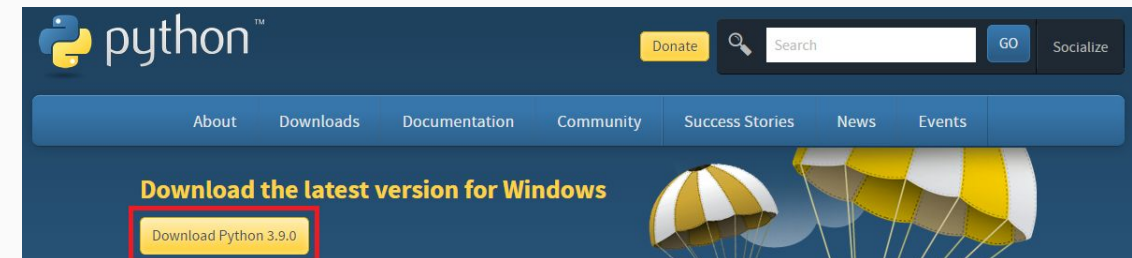
Instalare tool-uri - Python

1. Deschideți un browser pe pagina Google și căutați după textul:

“download python”

2. Folosiți rezultatul de pe pagina **python.org**, de obicei este primul în lista cu rezultate.

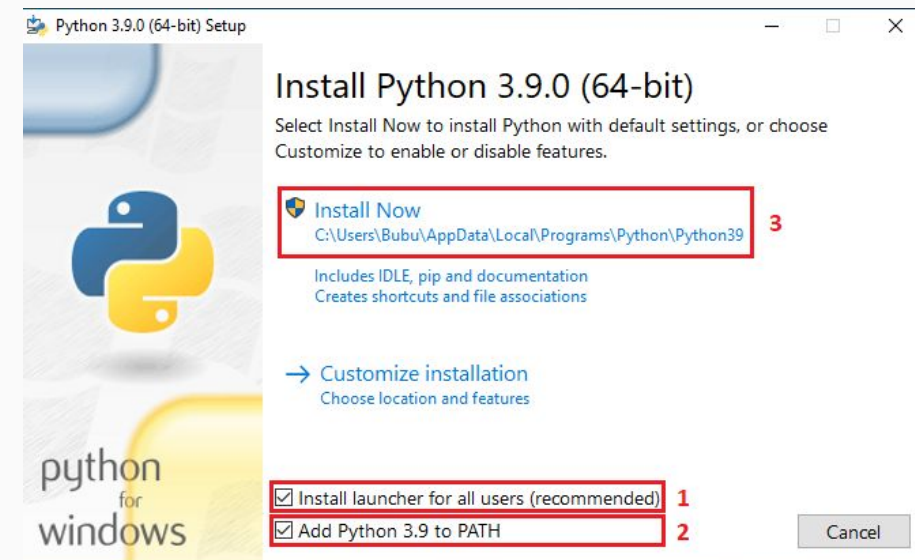
3. Pe pagina Python folosiți butonul de Download pentru a downloada ultima versiune de Python.



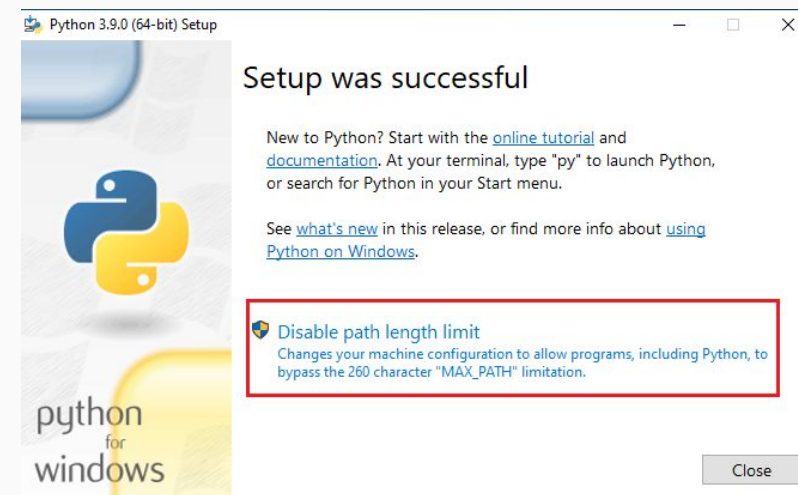
Instalare tool-uri - Python

4. După descărcarea executabilului, rulați-l și veți avea wizardul din imaginea alăturată.

- i. Bifați căsuța **Install launcher for all users** (pasul 1)
- ii. Bifați căsuța **Add Python 3.0 to PATH** (pasul 2)
- iii. Dați click pe **Install Now** (pasul 3)

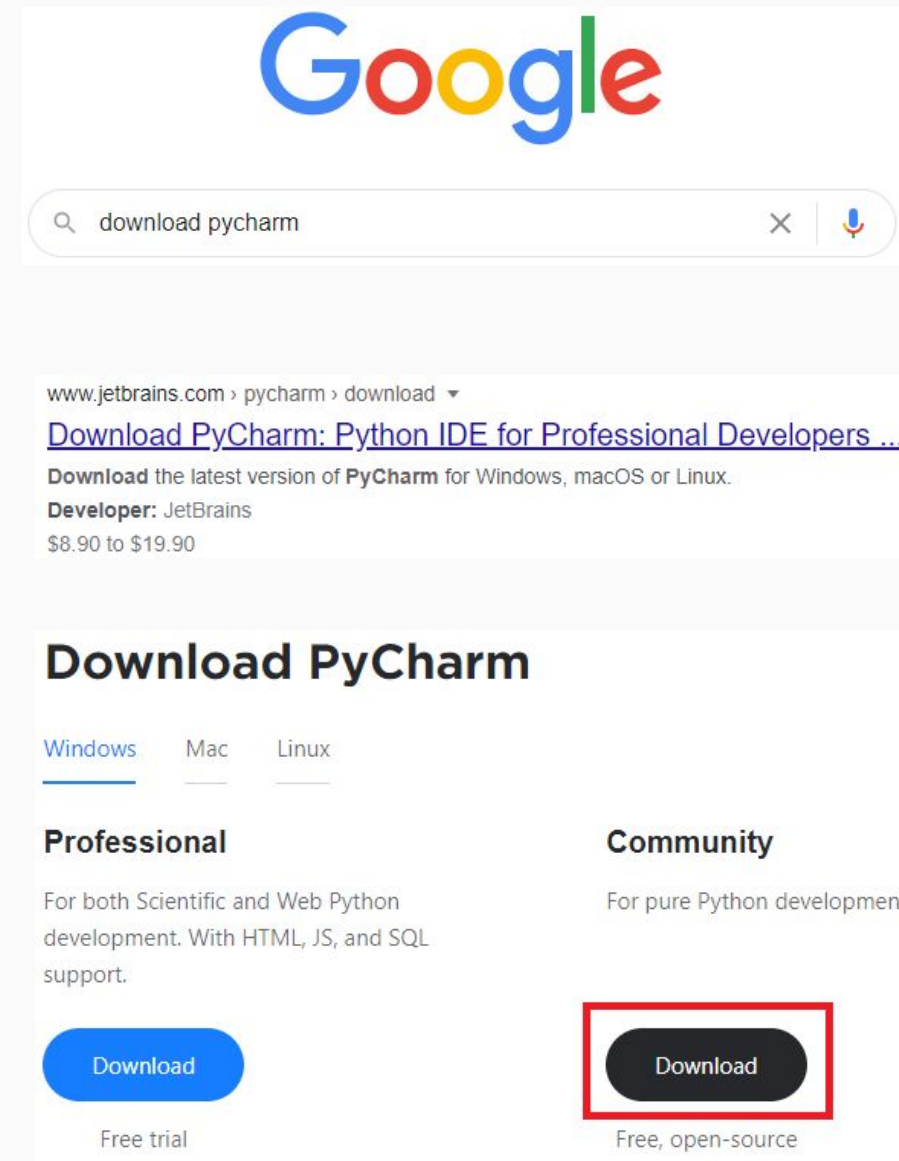


5. La finalul instalării, înainte de a apăsa butonul de Close, folosiți opțiunea **Disable path length limit**.



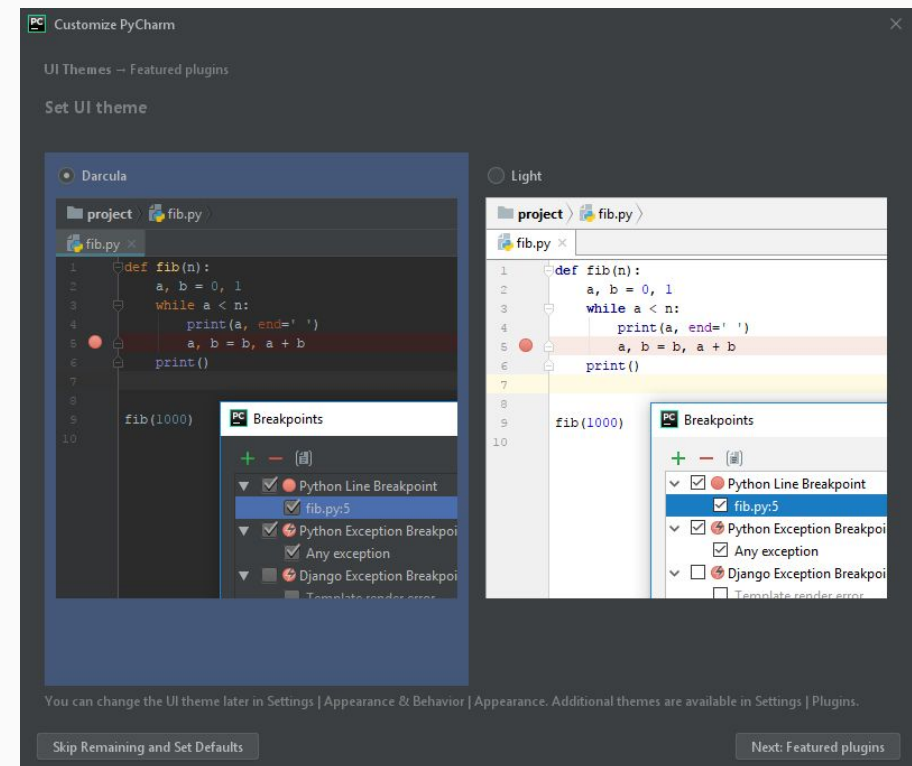
Instalare tool-uri - PyCharm

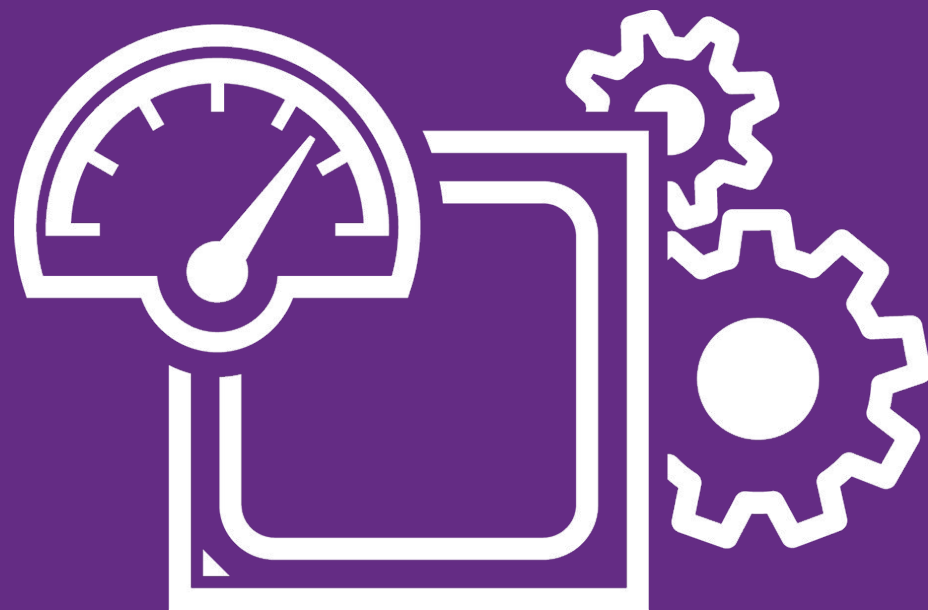
1. Folosind același motor de căutare, Google, căutați după textul “**download pycharm**”.
2. Rezultatul pe care îl căutați este pagina **jetbrains.com** (*pycharm/download*), ca în imaginea alăturată - de obicei primul rezultat.
3. Descărcați versiunea **Community**. Aceasta versiune este free, deci vă permite accesul gratuit la program și la o suită considerabilă de tool-uri folosite în programarea Python. După descărcarea executabilului, rulați-l și totul ar trebui să decurgă fără probleme.



Instalare tool-uri - PyCharm

4. După instalarea programului veți avea niște pași de configurare. Acești pași, în mare, țin de preferințele voastre legate de culori și de modul în care să arate editorul.
- pentru sănătatea ochilor voștri, vă recomand tema dark, chiar dacă la curs veți vedea că folosesc tema light. Motivul pentru care se întâmplă asta la curs este că e mai ușor de urmărit pentru voi.





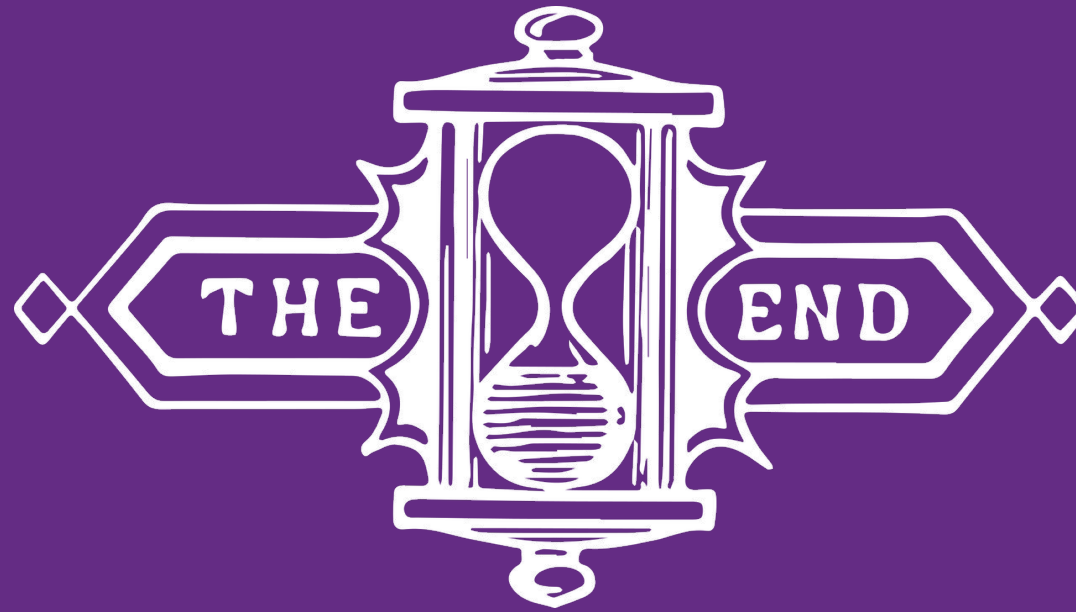
Temă



Temă

1. Instalați **PyCharm Community** pe calculatorul/laptop-ul personal.
2. Instalați ultima versiunea de **Python 3** pe calculatorul/laptop-ul personal.
 - Dacă folosiți sistemul Windows, nu uitați să adăugați Python la environment variables.
 - Acest pas se regăsește în wizard-ul de instalare (este un checkbox)
3. Creați-vă cont pe platforma **GitHub**.
 - Creați un repository public.
 - Trimiteți-mi pe Slack.





Vă mulțumesc!

