

Zajęcia 1

Dlaczego Python?

Darmowy

Nowoczesna i czytelna składnia

Prosty na start, jednocześnie posiadający rozbudowane biblioteki oraz społeczność.

Dynamiczne typowanie nie wymaga podawania typu zmiennych

Bardzo wiele zastosowań – aplikacje webowe, data science

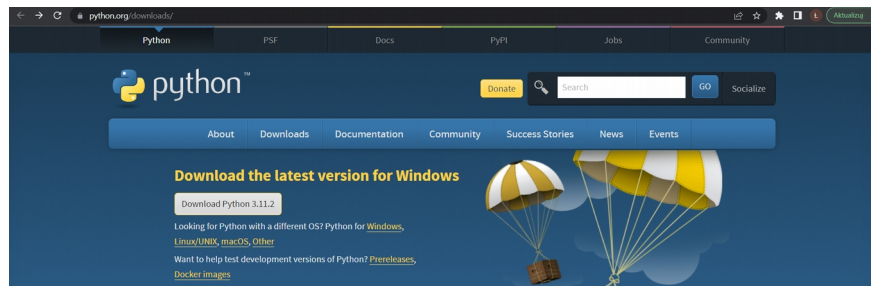
Wady:

Jest to język interpretowany, wolniejszy niż języki kompilowane.

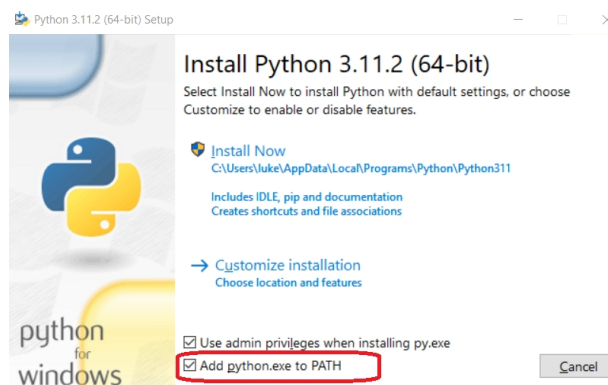
Dynamiczne typowanie – może być wadą jak i zaletą – brak statycznego typowania może powodować błędy trudne do wychwycenia.

Przed rozpoczęciem pracy, należy pobrać Pythona ze strony:

<https://www.python.org/downloads/>



Aby móc wywołać Pythona z konsoli należy dodać do zmiennej środowiskowej PATH jego ścieżkę. Można zrobić to podczas instalacji.

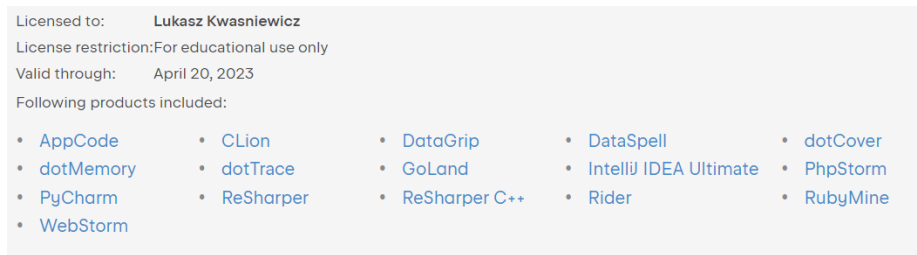


Podczas instalacji można też zmienić ścieżkę i zrezygnować z dodatkowych opcji (dokumentacja, moduły Tkinter, menadżer pakietów pip, czy narzędzi testowania).

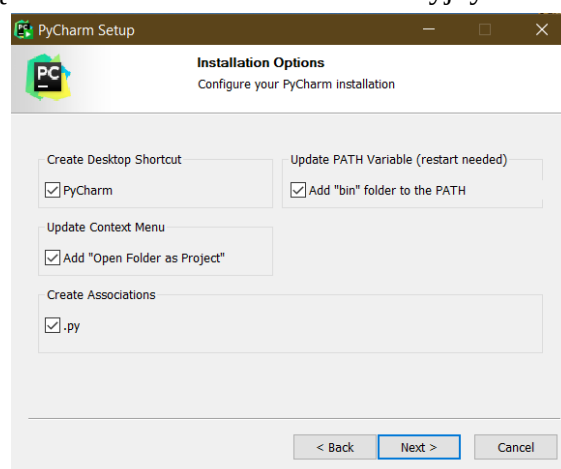
Jeżeli instalacja się udała polecenie wpisane w konsoli: `python --version` powinna zwrócić nam aktualną zainstalowaną wersję Pythona.

```
C:\Users\luke>python --version
Python 3.11.2
```

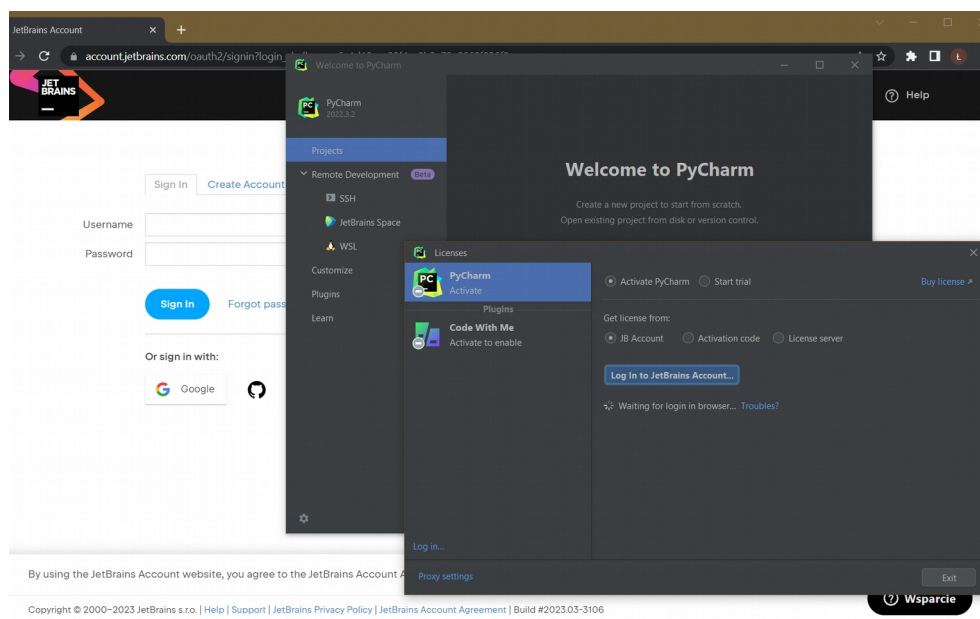
Studenci wykorzystując uczelnianego e-maila mogą uzyskać licencję na oprogramowanie JetBrains takie jak IntelliJ Idea, czy PyCharm:
<https://account.jetbrains.com/licenses>



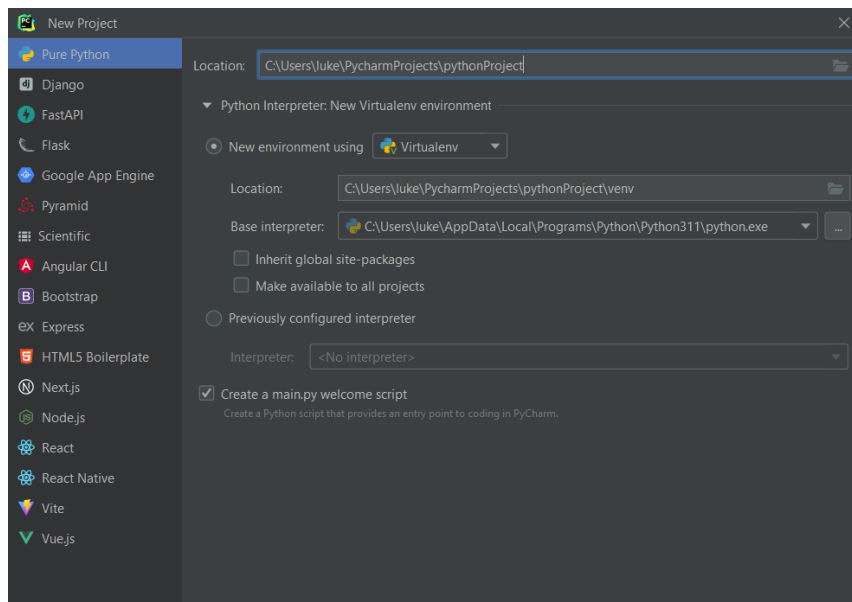
I zainstalować wersję **Professional** do celów edukacyjnych.



Po zainstalowaniu Pycharm'a należy aktywować licencję za pomocą konta



Następnie możemy utworzyć nowy projekt w Pycharm (Pure Python):



```
import webbrowser
if __name__ == '__main__':
    print("hello world")
    url = input("Podaj stronę internetową: ")
    webbrowser.open(url)
```

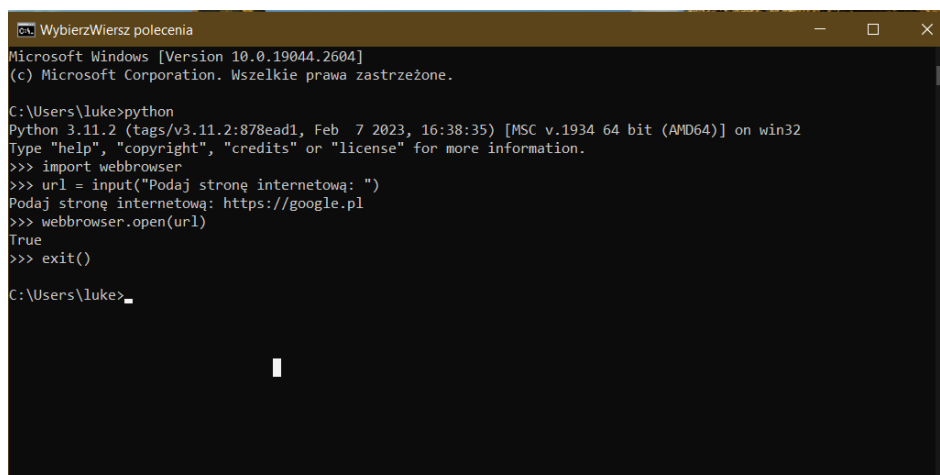
Jak widać przy tworzeniu projektu utworzyliśmy nowe środowisko wirtualne (virtualenv). Dla każdego projektu dobrym pomysłem jest utworzenie nowego środowiska wirtualnego – w celu uniknięcia konfliktów modułów i pakietów (np. 2 różne projekty używają tego samego modułu ale w różnej wersji).

Przykładowy program importuje moduł webbrowser, Sprawdza czy `__name__` jest równe `__main__` , ma to na celu sprawdzenie czy plik jest uruchamiany bezpośrednio – jako skrypt, czy poprzez import.

https://codecouple.pl/2016/02/27/python-specjalna-zmienna-__name__/

Jeżeli jest uruchamiany jako skrypt wyświetla napis „hello world”, prosi o podanie strony i otwiera ją w przeglądarce.

Pythona można używać też za pomocą interaktywnej powłoki:



Również możemy uruchomić skrypt w wirtualnym środowisku:

```
C:\Users\luke\PycharmProjects\pythonProject>.venv\Scripts\activate.bat

(venv) C:\Users\luke\PycharmProjects\pythonProject>python
Python 3.11.2 (tags/v3.11.2:878ead1, Feb  7 2023, 16:38:35) [MSC v.1934 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>> exit()

(venv) C:\Users\luke\PycharmProjects\pythonProject>python main.py
hello world
Podaj stronę internetową: https://google.com

(venv) C:\Users\luke\PycharmProjects\pythonProject>deactivate
C:\Users\luke\PycharmProjects\pythonProject>
```

W wirtualnym środowisku zainstalujemy moduł pozwalający odtworzyć mp3:
(moduł wymaga 64 bitowej biblioteki z programu vlc – libvlc.dll)
<https://get.videolan.org/vlc/3.0.18/win64/vlc-3.0.18-win64.exe>

Tworzenie wirtualnego środowiska i instalacja pakietu:

```
C:\Users\luke>mkdir projekt0

C:\Users\luke>cd projekt0

C:\Users\luke\projekt0>python -m venv wirtualne

C:\Users\luke\projekt0>.\wirtualne\Scripts\activate.bat
```

```
(wirtualne) C:\Users\luke\projekt0>where python
C:\Users\luke\projekt0\wirtualne\Scripts\python.exe
C:\Users\luke\AppData\Local\Programs\Python\Python311\python.exe
C:\Users\luke\AppData\Local\Microsoft\WindowsApps\python.exe

(wirtualne) C:\Users\luke\projekt0>pip install python-vlc
Collecting python-vlc
  Using cached python_vlc-3.0.18121-py3-none-any.whl (79 kB)
Installing collected packages: python-vlc
Successfully installed python-vlc-3.0.18121

[notice] A new release of pip available: 22.3.1 -> 23.0.1
[notice] To update, run: python.exe -m pip install --upgrade pip
```

```
(wirtualne) C:\Users\luke\projekt0>python
Python 3.11.2 (tags/v3.11.2:878ead1, Feb  7 2023, 16:38:35) [MSC v.1934 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>> import vlc
>>> p = vlc.MediaPlayer('file:///iCiebieTezBardzo.mp3')
>>> p.play()
0
>>> p.stop()
>>> exit()

(wirtualne) C:\Users\luke\projekt0>dir
Volume in drive C has no label.
Volume Serial Number is 8ED6-D3F2

Directory of C:\Users\luke\projekt0

06.03.2023  04:22    <DIR>          .
06.03.2023  04:22    <DIR>          ..
06.03.2023  04:22             3 386 233 iCiebieTezBardzo.mp3
06.03.2023  04:20    <DIR>          wirtualne
                1 File(s)          3 386 233 bytes
                3 Dir(s)   21 579 517 952 bytes free

(wirtualne) C:\Users\luke\projekt0>.\wirtualne\Scripts\deactivate.bat
C:\Users\luke\projekt0>python
Python 3.11.2 (tags/v3.11.2:878ead1, Feb  7 2023, 16:38:35) [MSC v.1934 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>> import vlc
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
ModuleNotFoundError: No module named 'vlc'
>>>
```

Alternatywna wersja bez VLC:
pip install playsound

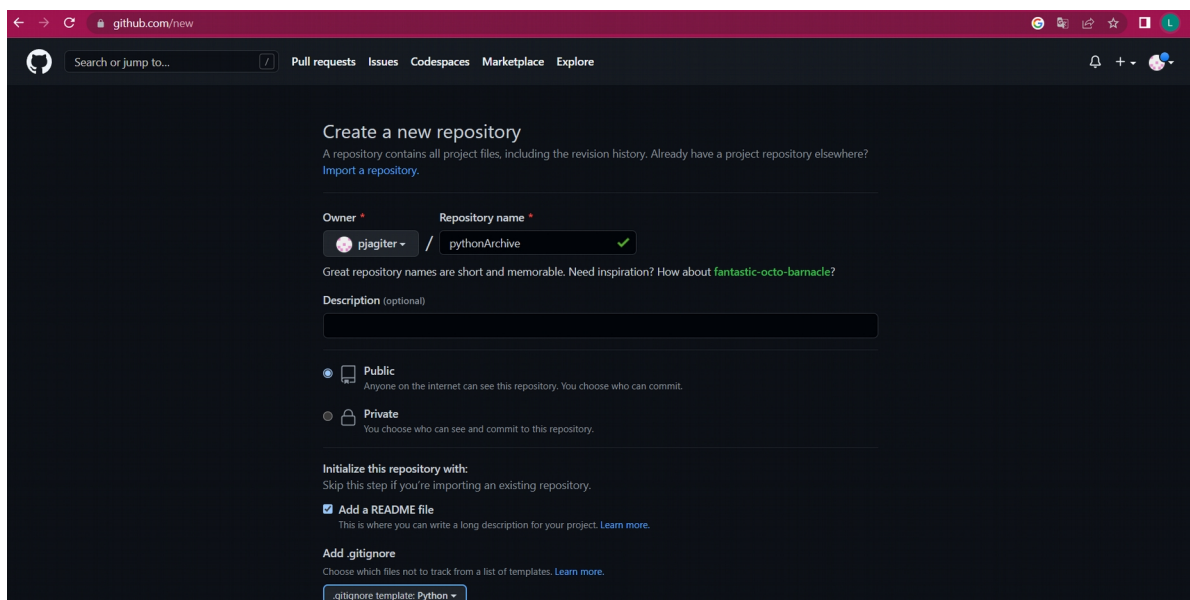
```
from playsound import playsound  
playsound('/path/file.mp3')
```

Git:

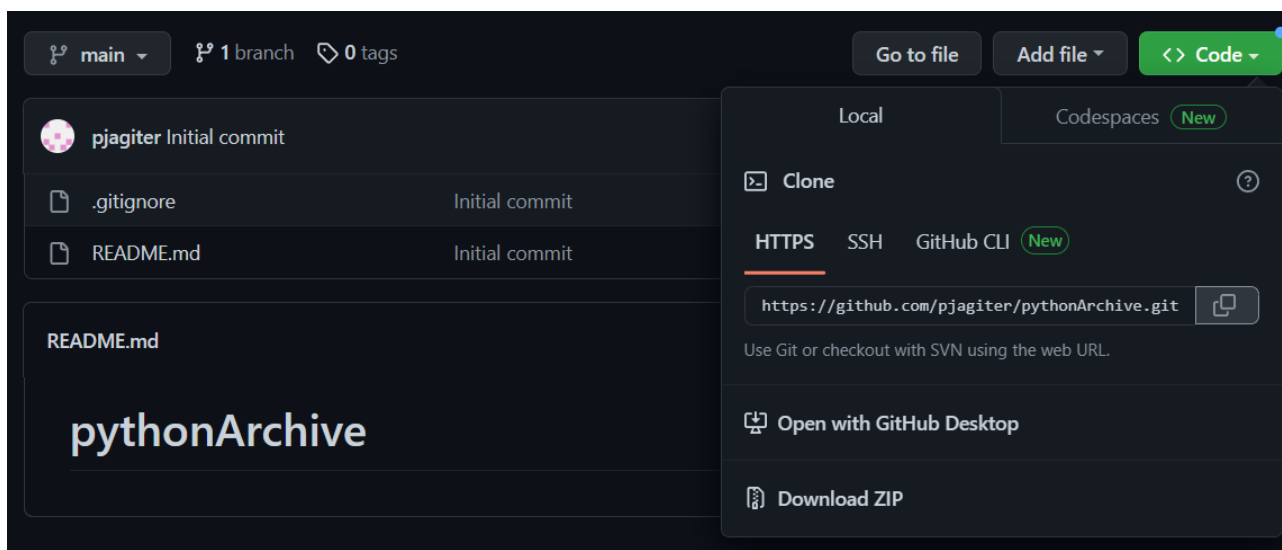
Pobieramy Gita ze strony: <https://gitforwindows.org/> i instalujemy z domyślnymi ustawieniami.



Zakładamy konto na githubie (<https://github.com/>) oraz tworzymy nowe repozytorium:



Sklonujemy repozytorium z githaba lokalnie:



```
git clone https://github.com/pjagiter/pythonArchive.git
```

Ustawiamy nazwę użytkownika i email w konsoli:

```
C:\Users\luke>git config --global user.name "user"
```

```
C:\Users\luke>git config --global user.email "user@pjwstk.edu.pl"
```

Możemy skopiować skrypt z projektu utworzonego w pycharmie (main.py) do katalogu ze sklonowanym repozytorium.

Dodajemy pliki do repozytorium: `git add .`

Robimy commit dodając komentarz, np. : `git commit -m "skrypt otwierający stronę"`

Zanim wrzucimy zmiany sprawdzimy, czy nie ma innych zmian w repozytorium za pomocą `git pull`

Następnie wrzucamy zmiany na serwer github używając `git push`

```
luke@DESKTOP-SBMVVMV MINGW64 ~/pythonArchive (main)
$ git status
On branch main
Your branch is ahead of 'origin/main' by 1 commit.
(use "git push" to publish your local commits)

nothing to commit, working tree clean

luke@DESKTOP-SBMVVMV MINGW64 ~/pythonArchive (main)
$ git push
Enumerating objects: 4, done.
Counting objects: 100% (4/4), done.
Delta compression using up to 12 threads
Compressing objects: 100% (3/3), done.
Writing objects: 100% (3/3), 452 bytes | 452.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
To https://github.com/pjagiter/pythonArchive.git
b078672..01ca718 main -> main
```

Po poprawnym zalogowaniu zmiany zostaną wrzucone na serwer.

Zadania:

1. Zainstalować na swoim komputerze Python'a
2. oraz PyCharm.
3. Utworzyć nowe środowisko wirtualne zainstalować moduł do odtwarzania dźwięku i odtworzyć mp3 – w skrypcie.
4. Zmodyfikować skrypt z github tak aby pobierał archiwalne 3 wersje podanej strony internetowej wysłać zmiany na github. (będzie potrzebny dodatkowy moduł -requests)

```
#przyklad dzialania:
#pageurl - pobrana strona, date-data w formacie rok miesiac dzien np. 20230126
#zapytanie do api:
#http://archive.org/wayback/available?url=example.com&timestamp=20060101
url = "http://archive.org/wayback/available?url="+pageurl+"&timestamp="+str(date)
response = requests.get(url)
d = response.json()
page = d["archived_snapshots"]["closest"]["url"]
webbrowser.open(page)
```

5. Dlaczego w Pythonie mówi się o interpreterze, a w javie o maszynie wirtualnej?

Całość - kod, polecenia z konsoli, screeny stron internetowych i odpowiedź na pytanie umieścić w pdf na teams w przypisanym Zadaniu .