



Państwowa Wyższa Szkoła Zawodowa w Tarnowie

Python: Deep Learning TensorFlow + Keras
Instrukcja instalacji

Skład grupy:
Arkadiusz Kopacz
Wojciech Nowak
Miłosz Wilczyński
Patrik Zając

Prowadzący:
prof. dr hab. inż. Antoni Ligęza

Krótki spis treści

<u>Potrzebne narzędzia i biblioteki</u>	<u>3</u>
<u>Instrukcja instalacji na systemie Windows 10</u>	<u>4</u>
<u>Moment wspólnej instrukcji instalacji</u>	<u>13</u>
<u>Instrukcja instalacji na systemie Ubuntu w wersji 12.04</u>	<u>16</u>
<u>Instrukcja instalacji na systemie opartym na Ubuntu w wersji 19.10</u>	<u>24</u>
<u>Program testowy, sprawdzenie poprawności instalacji plików</u>	<u>26</u>

Deep Learning TensorFlow + Keras

Potrzebne narzędzia	Biblioteki pomocnicze
a) Python 3.4 lub wyżej	Numpy
b) pip python	SciPy
c) virtualenv	Matplotlib
TensorFlow	Scikit-learn
Keras	Pandas

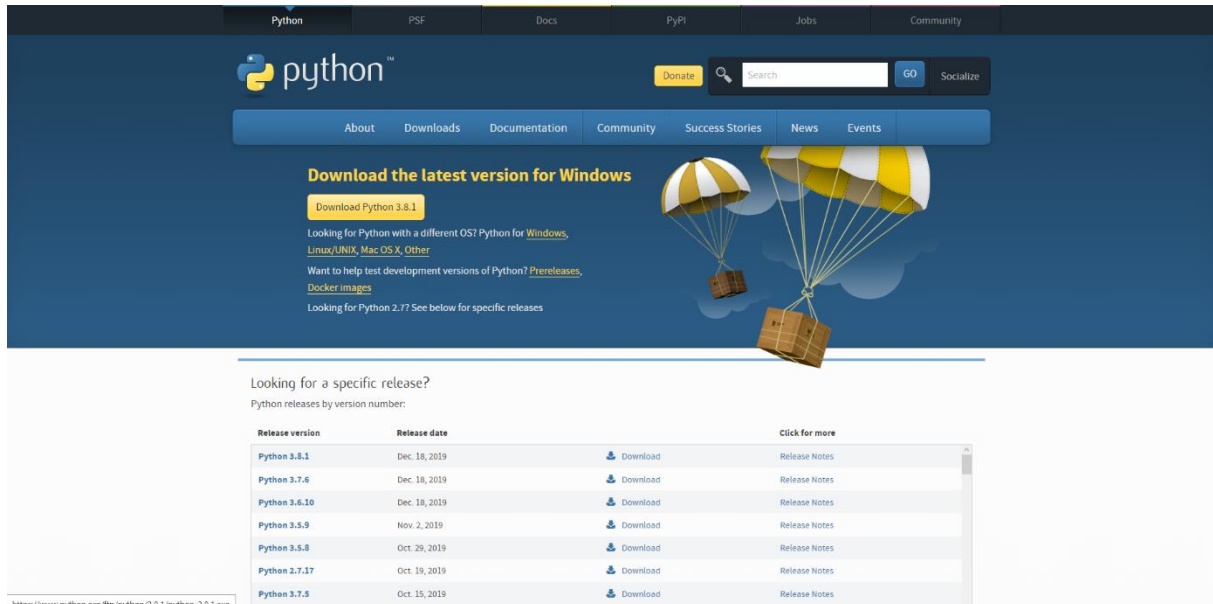
Chciałbym zaznaczyć, że od pewnego momentu **czynności na obydwu systemach nie będą się różnić**, dlatego w odpowiednim momencie **odwołam czytelnika do odpowiedniego punktu instrukcji**.
Odwołanie będzie się znajdować w instrukcji instalacji dla systemu Ubuntu!

- a) Według twórców zalecany jest Python3 w wersji 3.4 lub wyższej, w tej instrukcji użyliśmy Pythona 3.5 na Ubuntu w wersji 12.04 i wszystko działało w porządku. O ile się nie pomyliliśmy to na uczelni w laboratorium C109 zainstalowana jest wersja Pythona 3.5.
- b) **Pip** – system zarządzanie pakietami pythona.
- c) **Virtualenv** – emulator środowiska, zalecany przez producenta. Operacje wykonywane na bibliotekach TensorFlow + Teras mogą zawierać instrukcję z którymi może mieć problem normalne środowisko...

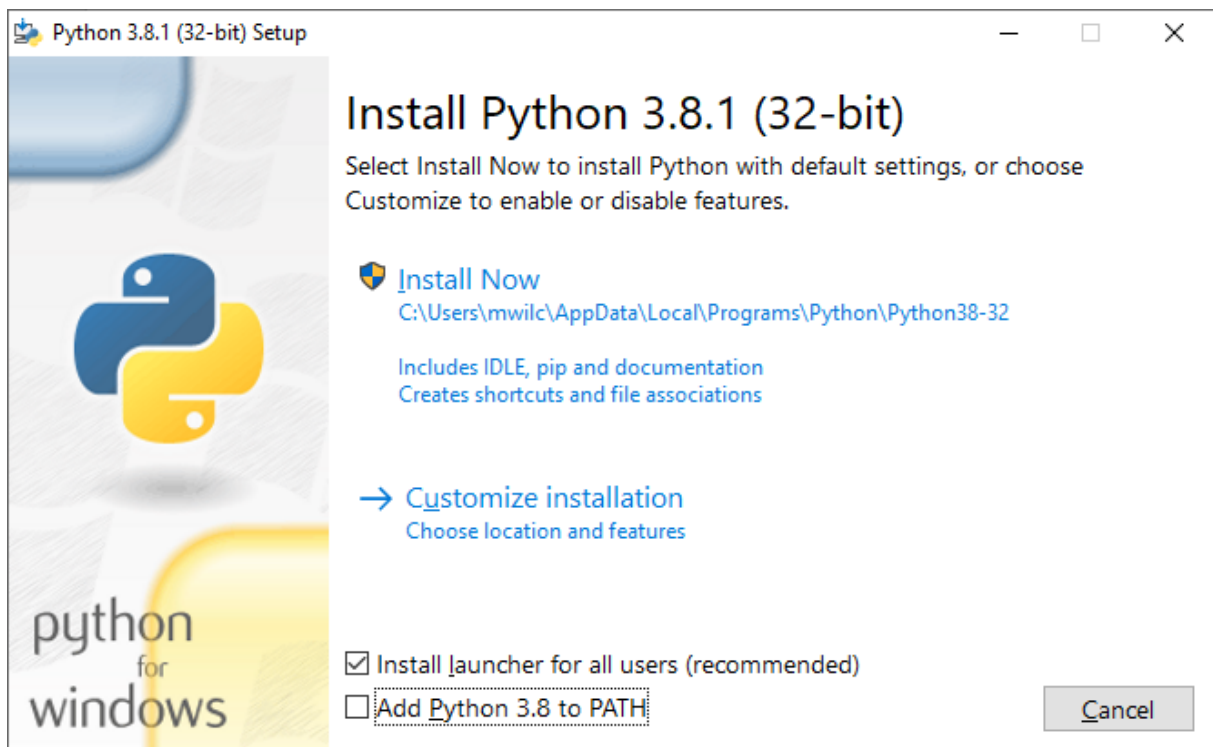
Instalacja na systemie Windows 10

1. Pobieramy program Python ze strony

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



2. Uruchamiamy instalator Pythona.

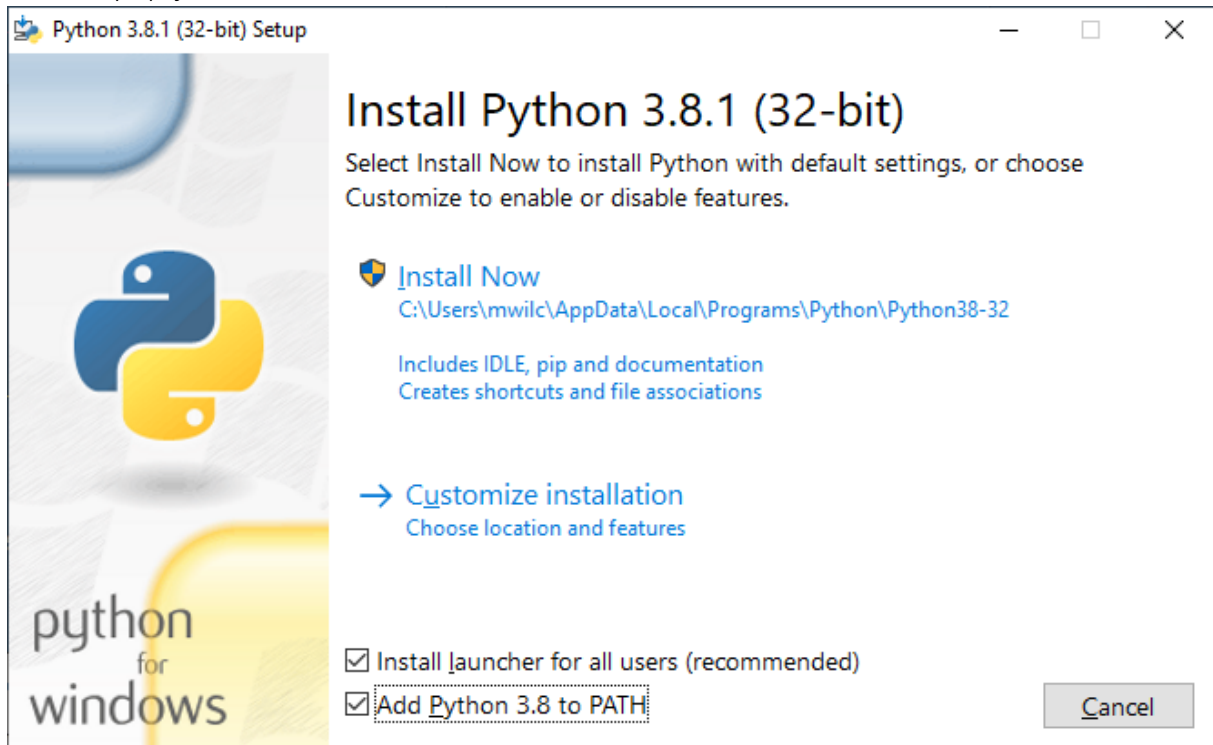


3. **BARDZO WAŻNE!** Zaznaczyć opcje

Add Python 3.X to PATH

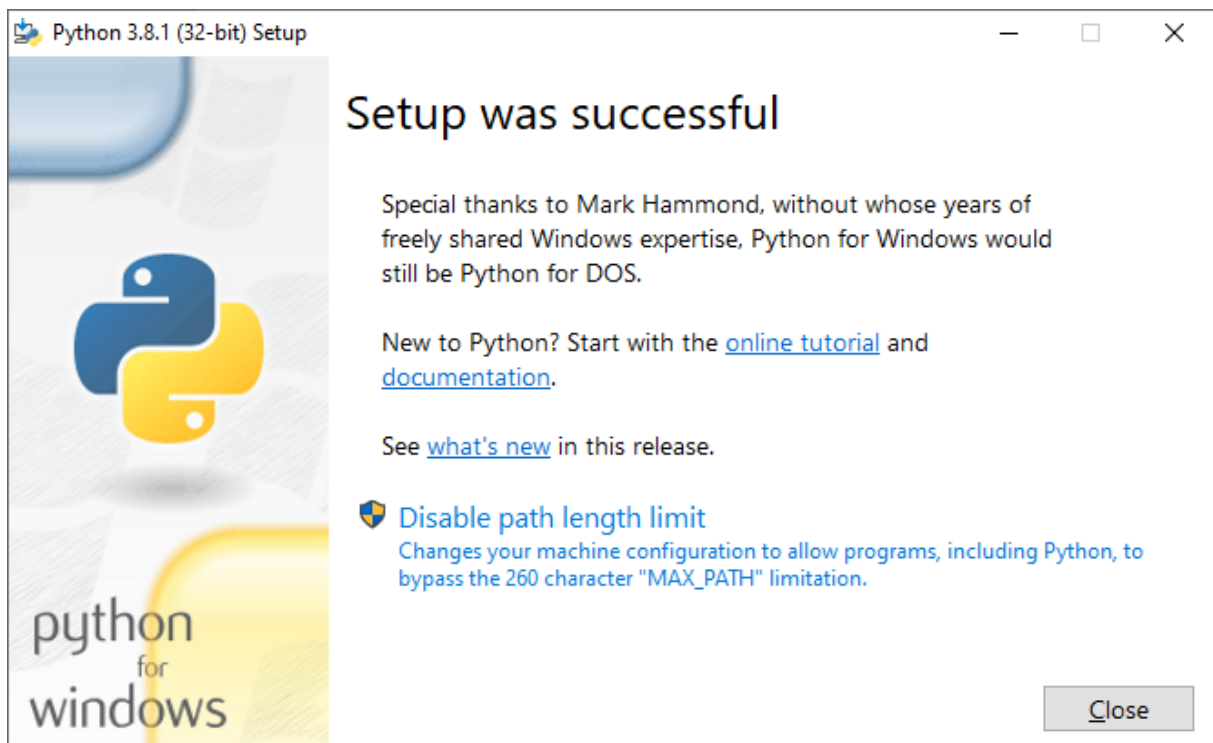
(gdzie X to wersja pythona).

4. Klikamy opcję **Install Now!**

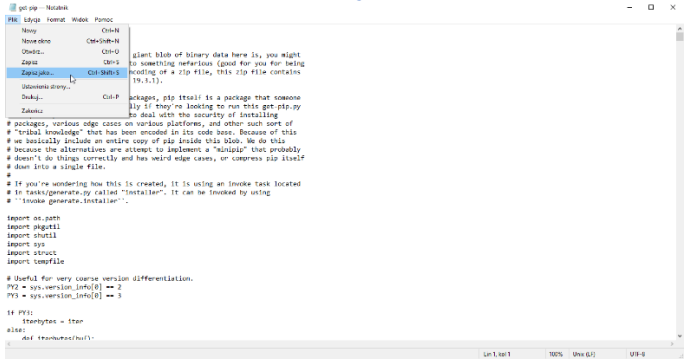
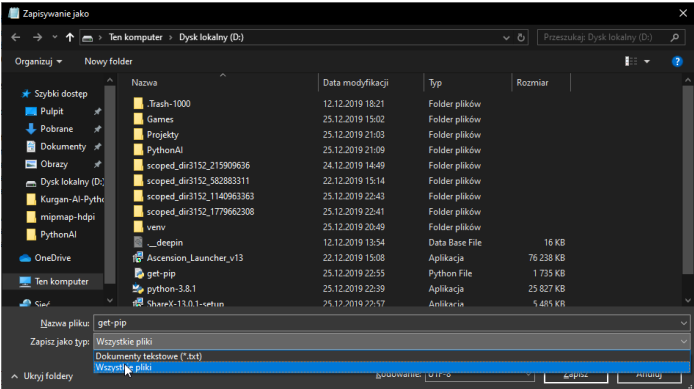


5. Czekamy, aż instalator zakończy pracę.

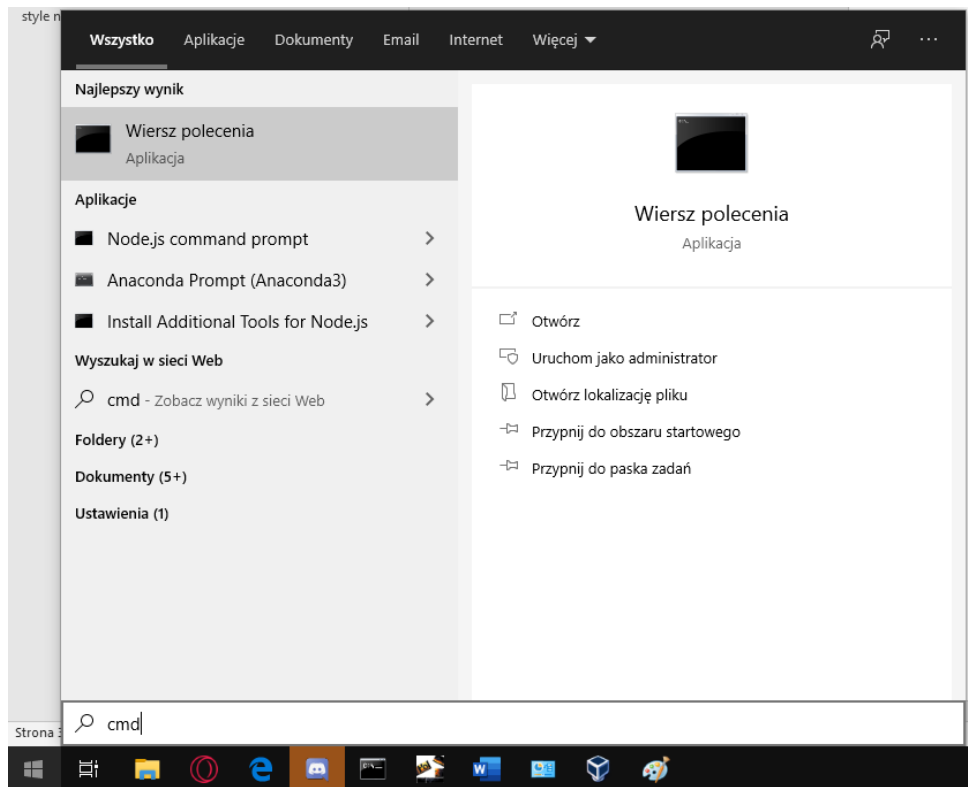
6. Opcjonalnie możemy kliknąć opcję **Disable path length limit**. W innym wypadku naciskamy przycisk **Close**.



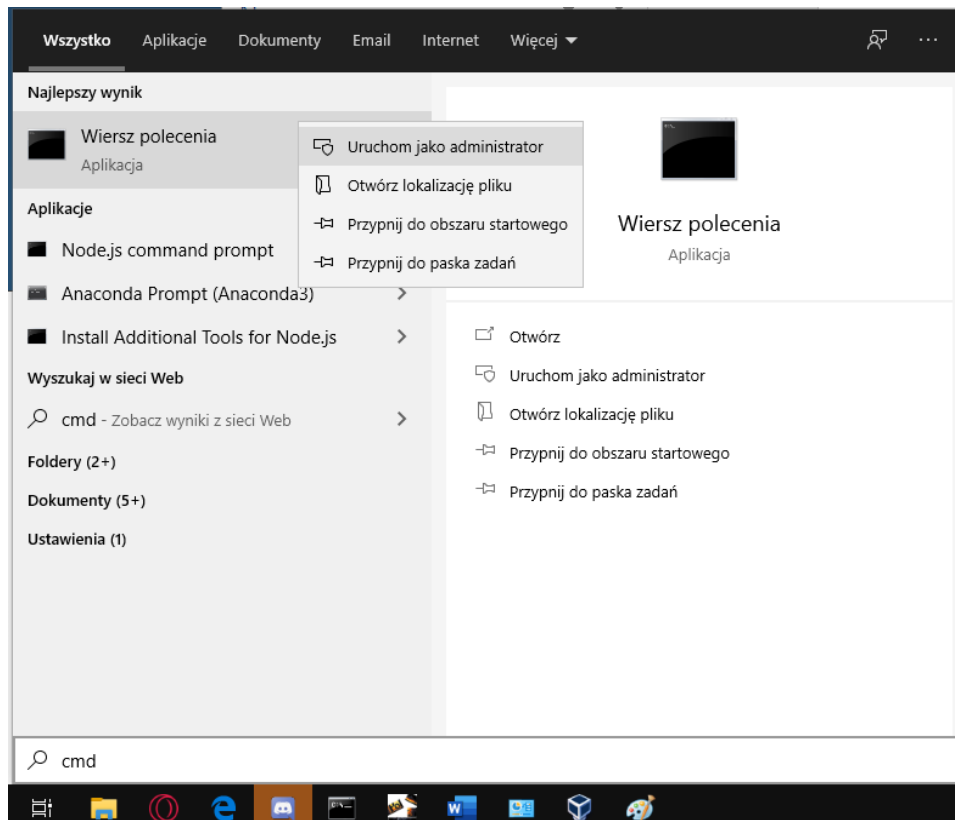
7. Wykonujemy instrukcje po lewej lub po prawej stronie tabeli

<p>1. Ściągamy plik ze strony: https://bootstrap.pypa.io/get-pip.py</p> <p>2 Gdy otworzy się strona, klikamy PPM (prawym przyciskiem myszy) na białym tle.</p>	<p>1. Wchodzimy na stronę: https://pastebin.com/UKr0cP1t</p> <p>2. Kopiujemy tekst zawarty na stronie.</p>
<p>3. Po pojawieniu się menu opcji wybieramy Zapisz jako.</p>	<p>3. Otwieramy notatnik i wklejamy kod.</p>
<p>4. Miejsce zapisu pliku polecam wybrać jakieś łatwo dostępne miejsce, jak partycję C:/ lub D:/.</p>	<p>4. W notatniku z menu opcji wybieramy Plik, a następnie Zapisz jako.</p> 
<p>5. Wybieramy jakieś łatwo dostępne miejsce do zapisu pliku, jak partycję C:/ lub D:/.</p>	<p>6. W nowo pokazanym okienku „Zapisywanie jako” szukamy opcje Zapisz jako typ:</p>
<p>8. Wybieramy tę opcję i zmieniamy wybór z Dokumenty tekstowe (*.txt) na Wszystkie pliki.</p>	<p>9. W okienku Nazwa pliku: napiszmy get-pip.py (.py to rozszerzenie, które obsługuje python).</p> 

8. Uruchamiamy **Wiersz poleceń**. Wchodzimy w **START** wpisujemy **CMD**.

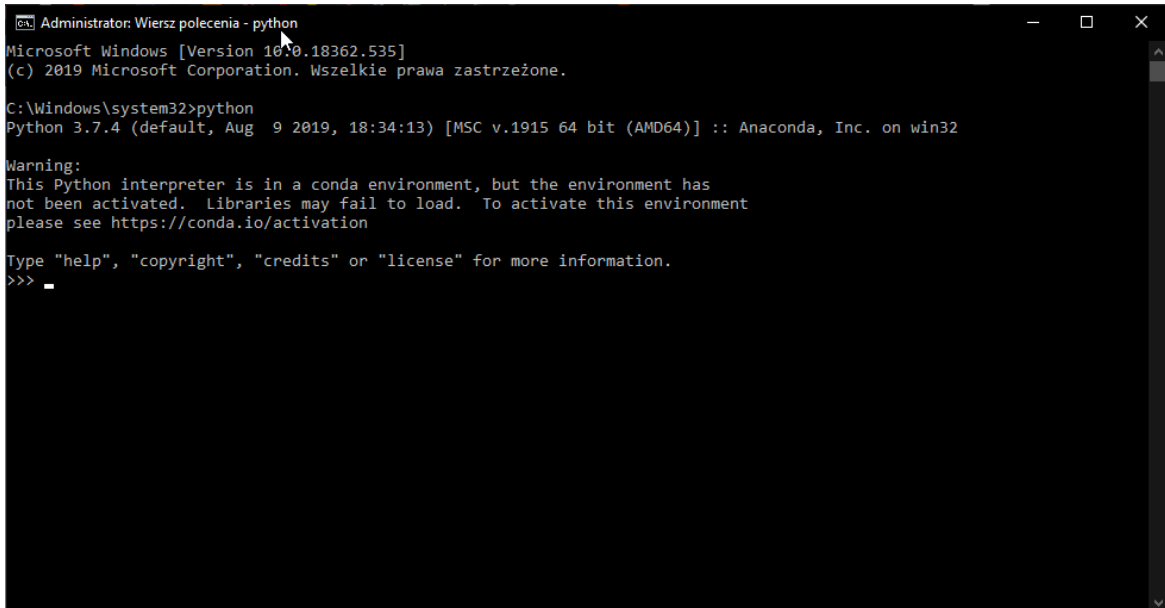


9. Klikamy **PPM (prawym przyciskiem myszy)** na podświetlony **Wiersz poleceń** i klikamy **Uruchom jako administrator**.



10. W wierszu poleceń wpisujemy komendę

python



```
Administrator: Wiersz polecenia - python
Microsoft Windows [Version 10.0.18362.535]
(c) 2019 Microsoft Corporation. Wszelkie prawa zastrzeżone.

C:\Windows\system32>python
Python 3.7.4 (default, Aug  9 2019, 18:34:13) [MSC v.1915 64 bit (AMD64)] :: Anaconda, Inc. on win32

Warning:
This Python interpreter is in a conda environment, but the environment has
not been activated. Libraries may fail to load. To activate this environment
please see https://conda.io/activation

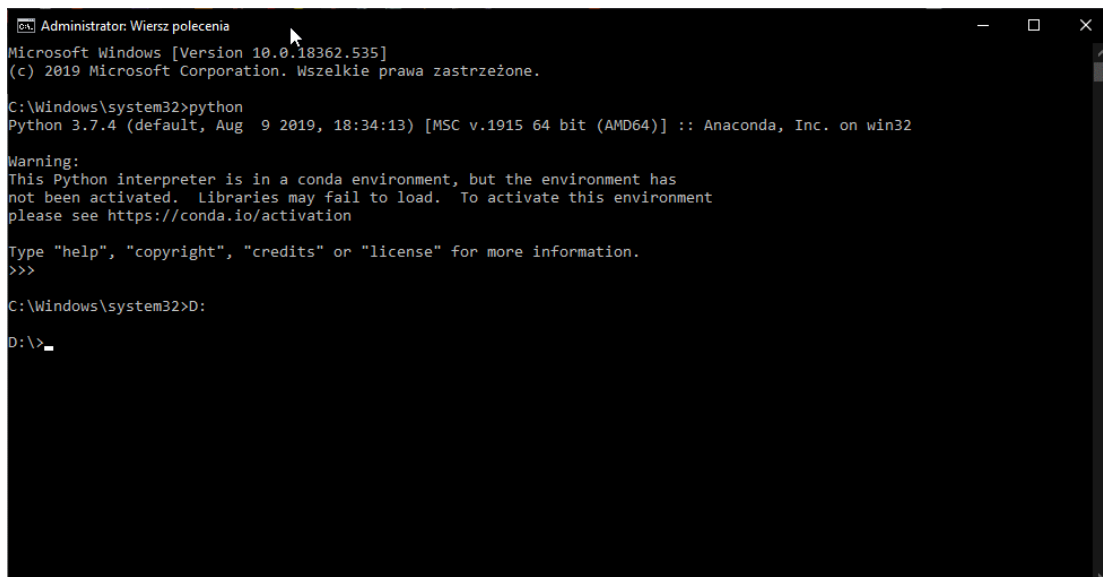
Type "help", "copyright", "credits" or "license" for more information.
>>> _
```

Powinno odpalić się Python. Pierwsza linia informuje o aktualnie używanej wersji pythona.

UWAGA! Jeżeli wiersz poleceń nie rozpoznaje tego polecenia to oznacza, że nie zaznaczyłeś Add PATH podczas instalacji. Aby to naprawić polecam ponownie zainstalować pythona i uruchomić ponownie komputer. Albo skorzystać z tego linku

<https://superuser.com/questions/143119/how-do-i-add-python-to-the-windows-path>

11. Wychodzimy z pythona skrótem klawiszowym **Control + Z**, **Control + C** lub poleceniem **exit()**.



```
Administrator: Wiersz polecenia
Microsoft Windows [Version 10.0.18362.535]
(c) 2019 Microsoft Corporation. Wszelkie prawa zastrzeżone.

C:\Windows\system32>python
Python 3.7.4 (default, Aug  9 2019, 18:34:13) [MSC v.1915 64 bit (AMD64)] :: Anaconda, Inc. on win32

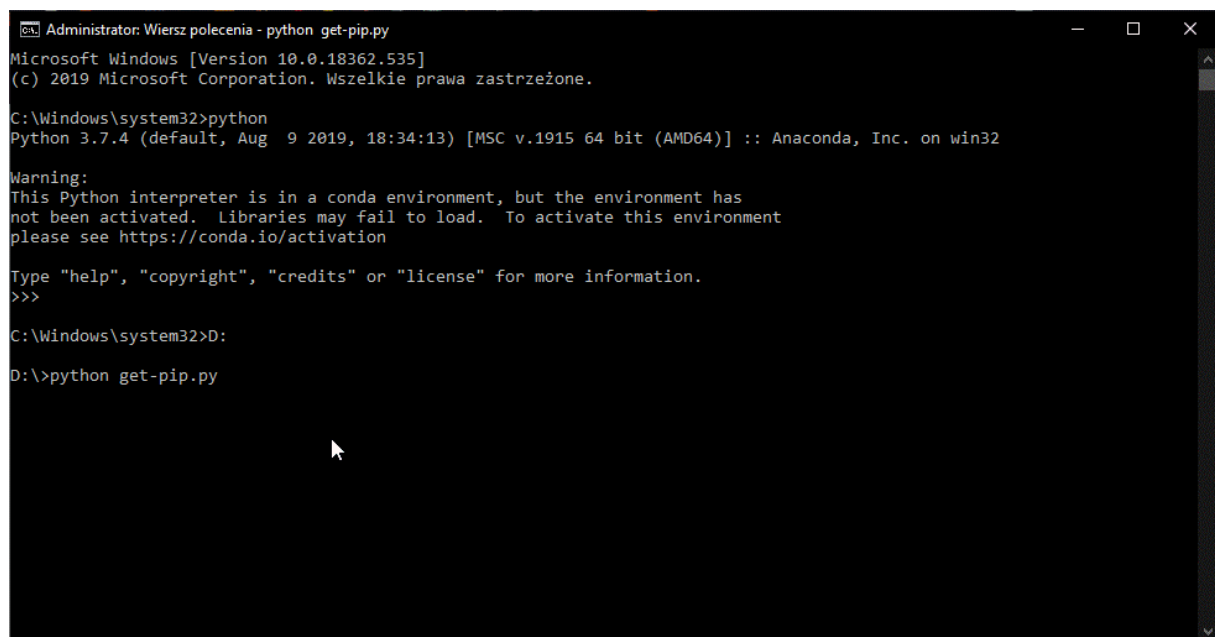
Warning:
This Python interpreter is in a conda environment, but the environment has
not been activated. Libraries may fail to load. To activate this environment
please see https://conda.io/activation

Type "help", "copyright", "credits" or "license" for more information.
>>>
C:\Windows\system32>exit()
D:\>_
```

12. Przechodzimy do miejsca, gdzie zapisaliśmy nasze **get-pip.py**. U mnie jest to partycja **D:**.

Aby przemieścić się pomiędzy partycjami wpisujemy po prostu **literę partycji + :** (dwukropek). Na przykład **D:**

13. Wpisujemy w wierszu poleceń komendę `python get-pip.py`



```
Administrator: Wiersz polecenia - python get-pip.py
Microsoft Windows [Version 10.0.18362.535]
(c) 2019 Microsoft Corporation. Wszelkie prawa zastrzeżone.

C:\Windows\system32>python
Python 3.7.4 (default, Aug  9 2019, 18:34:13) [MSC v.1915 64 bit (AMD64)] :: Anaconda, Inc. on win32

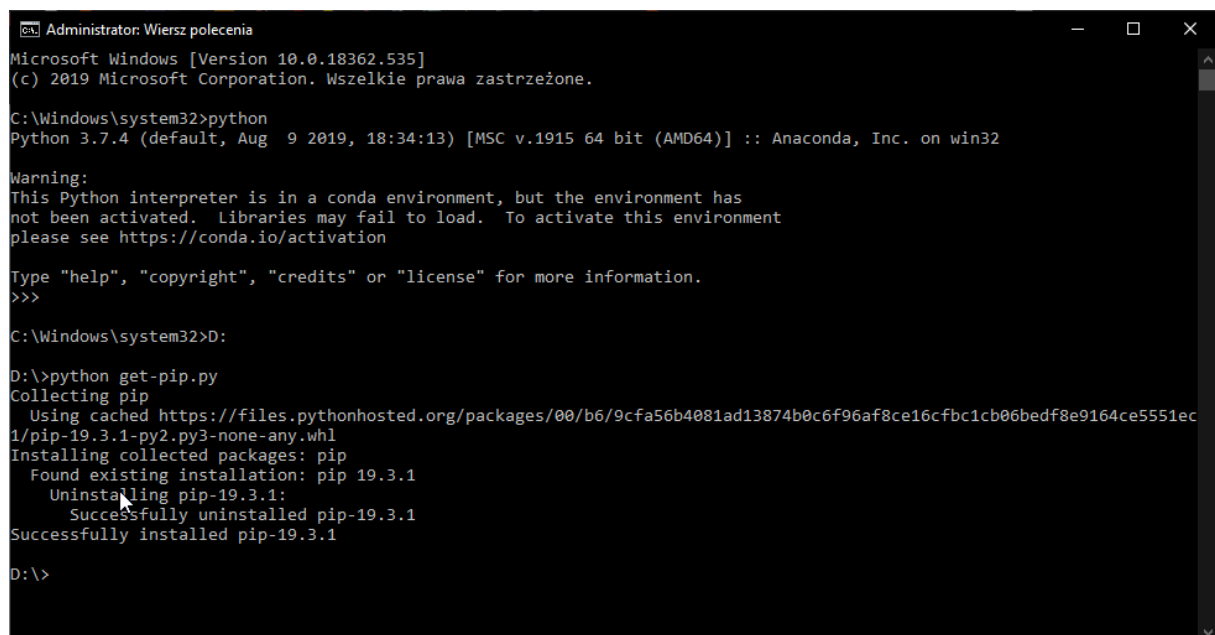
Warning:
This Python interpreter is in a conda environment, but the environment has
not been activated. Libraries may fail to load. To activate this environment
please see https://conda.io/activation

Type "help", "copyright", "credits" or "license" for more information.
>>>

C:\Windows\system32>D:
D:\>python get-pip.py
```

I wciskamy **ENTER**.

14. Czekamy, aż pip się zainstaluje.



```
Administrator: Wiersz polecenia
Microsoft Windows [Version 10.0.18362.535]
(c) 2019 Microsoft Corporation. Wszelkie prawa zastrzeżone.

C:\Windows\system32>python
Python 3.7.4 (default, Aug  9 2019, 18:34:13) [MSC v.1915 64 bit (AMD64)] :: Anaconda, Inc. on win32

Warning:
This Python interpreter is in a conda environment, but the environment has
not been activated. Libraries may fail to load. To activate this environment
please see https://conda.io/activation

Type "help", "copyright", "credits" or "license" for more information.
>>>

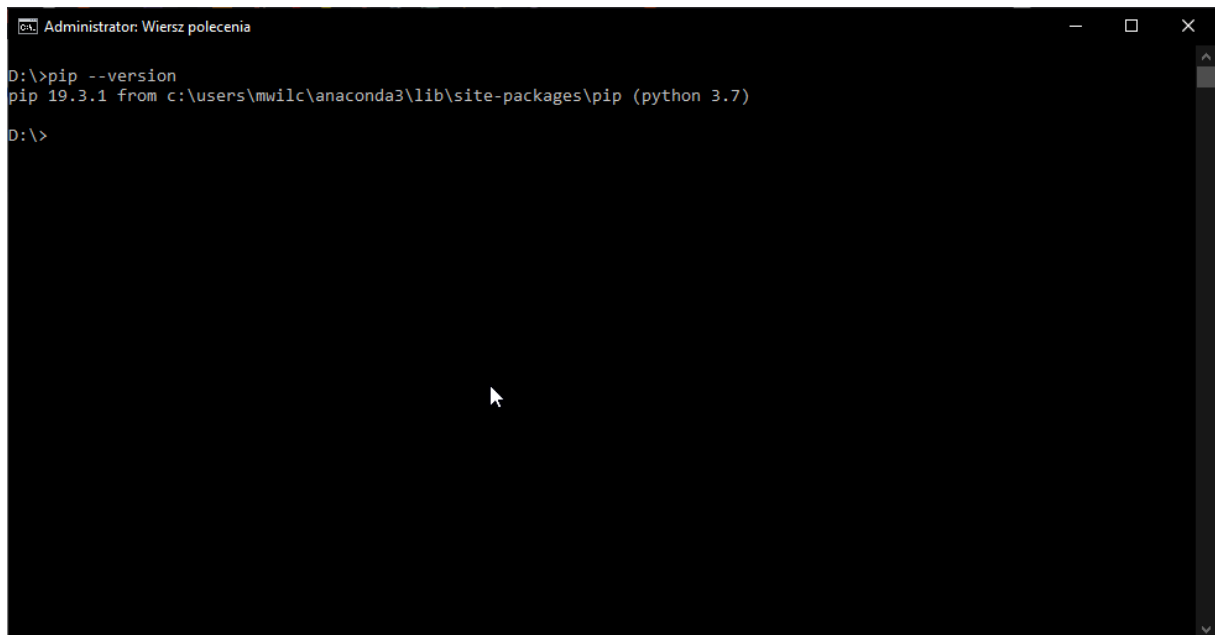
C:\Windows\system32>D:
D:\>python get-pip.py
Collecting pip
  Using cached https://files.pythonhosted.org/packages/00/b6/9cfa56b4081ad13874b0c6f96af8ce16cfbc1cb06bedf8e9164ce5551ec
1/pip-19.3.1-py2.py3-none-any.whl
Installing collected packages: pip
  Found existing installation: pip 19.3.1
    Uninstalling pip-19.3.1:
      Successfully uninstalled pip-19.3.1
Successfully installed pip-19.3.1

D:\>
```

Na tym screenie nie było żadnych pasków progresu, ponieważ pakiet był już zainstalowany.

15. Sprawdźmy wersję **pip'a**. Wpiszmy w wierszu poleceń komendę

pip --version

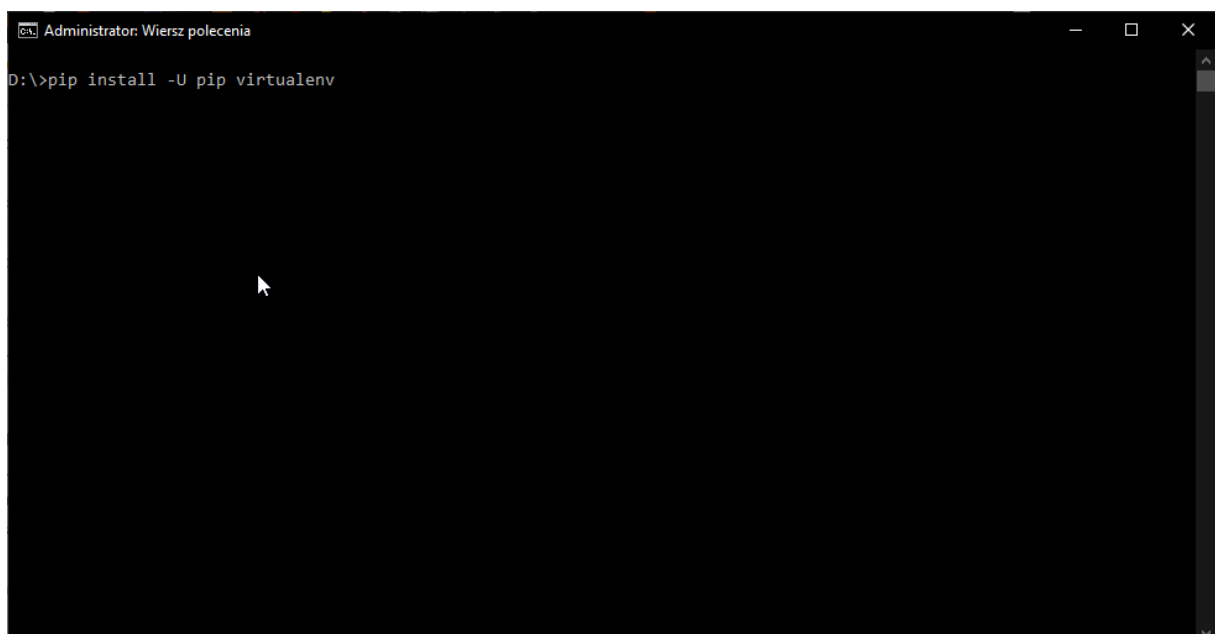


```
Administrator: Wiersz polecenia
D:\>pip --version
pip 19.3.1 from c:\users\mmwlc\anaconda3\lib\site-packages\pip (python 3.7)
D:\>
```

Jeżeli działa to znaczy, że pip został zainstalowany pomyślnie.

16. Teraz musimy zainstalować **Virtual Environment**. W tym celu wpisujemy w wierszu poleceń komendę

pip install -U pip virtualenv



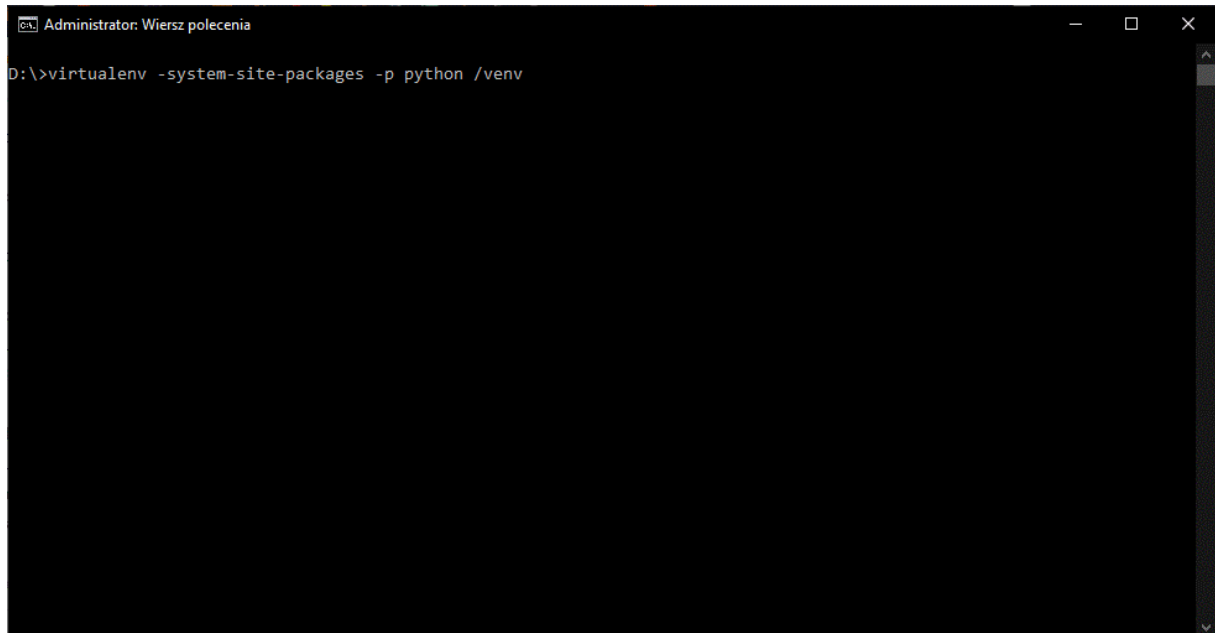
```
Administrator: Wiersz polecenia
D:\>pip install -U pip virtualenv
D:\>
```

Klikamy **ENTER**.

17. Czekamy, aż instalator zakończy pracę.

18. Tworzymy nowe środowisko, które będzie używać interpretera pythona. „**venv**” będzie nazwą katalogu, gdzie będzie zainstalowane to środowisko.
Wpisujemy więc komendę

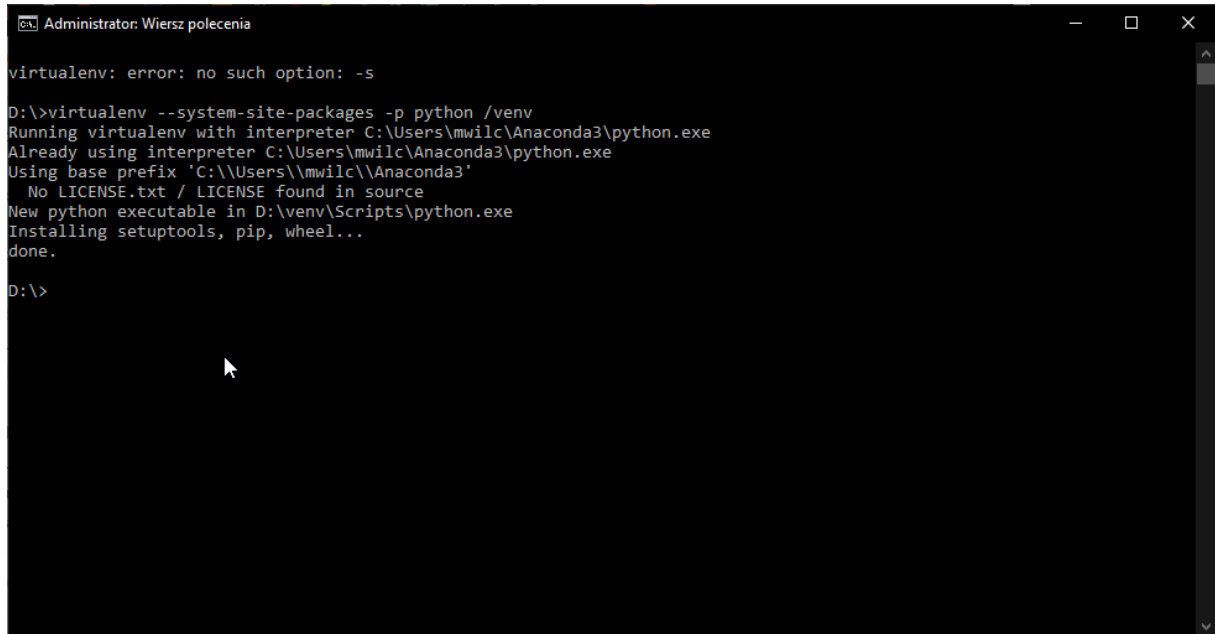
virtualenv --system-site-packages -p python /venv



```
Administrator: Wiersz polecenia
D:\>virtualenv -system-site-packages -p python /venv
```

I klikamy **ENTER**.

20. Czekamy, aż instalator zakończy pracę.



```
Administrator: Wiersz poleceń

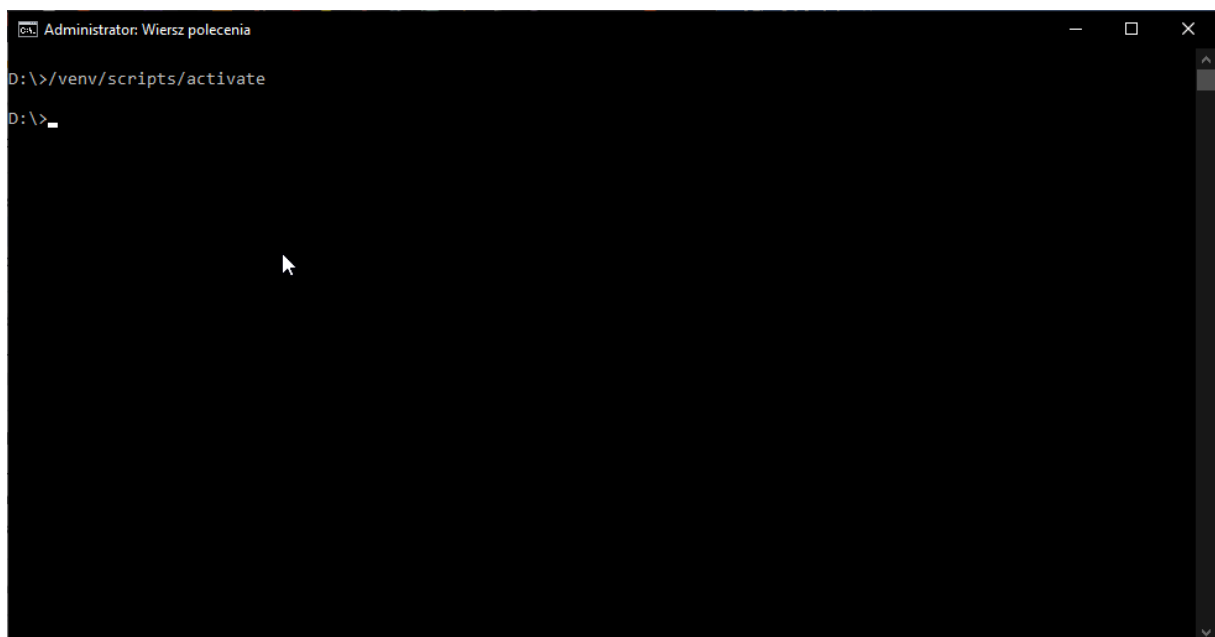
virtualenv: error: no such option: -s

D:\>virtualenv --system-site-packages -p python /venv
Running virtualenv with interpreter C:\Users\mwilc\Anaconda3\python.exe
Already using interpreter C:\Users\mwilc\Anaconda3\python.exe
Using base prefix 'C:\Users\mwilc\Anaconda3'
No LICENSE.txt / LICENSE found in source
New python executable in D:\venv\Scripts\python.exe
Installing setuptools, pip, wheel...
done.

D:\>
```

21. Wpisujemy komendę

/venv/Scripts/activate

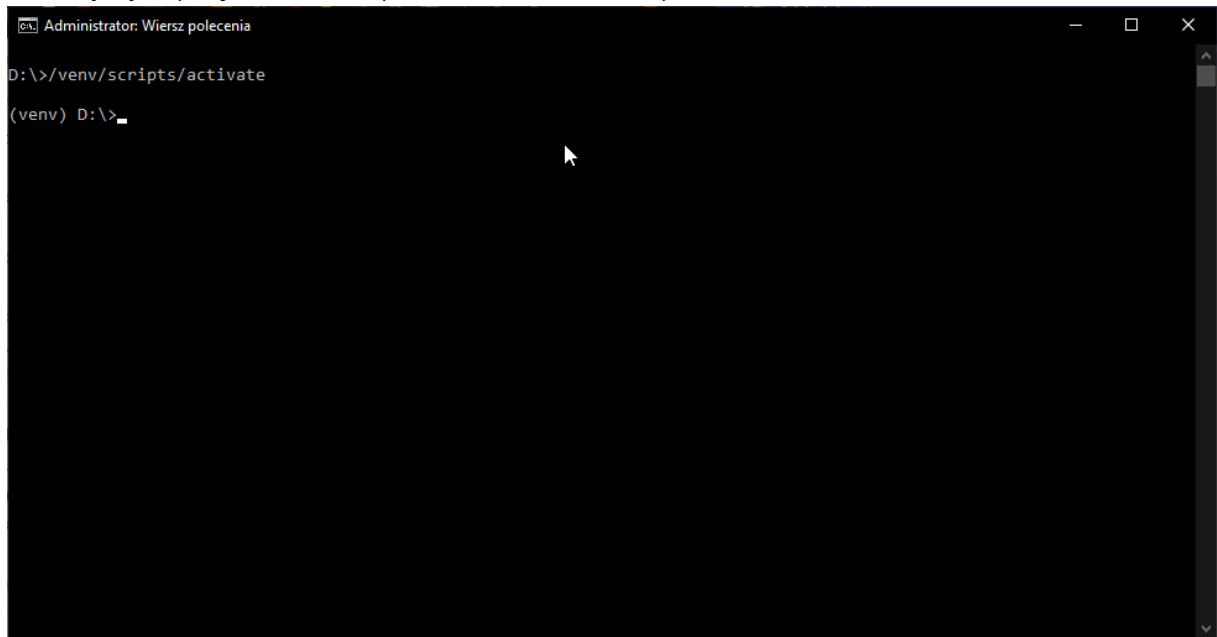


```
Administrator: Wiersz poleceń

D:\>/venv/scripts/activate
D:\>_
```

i klikamy **ENTER**.

22. Znajdujemy się teraz w naszym środowisku wirtualnym.

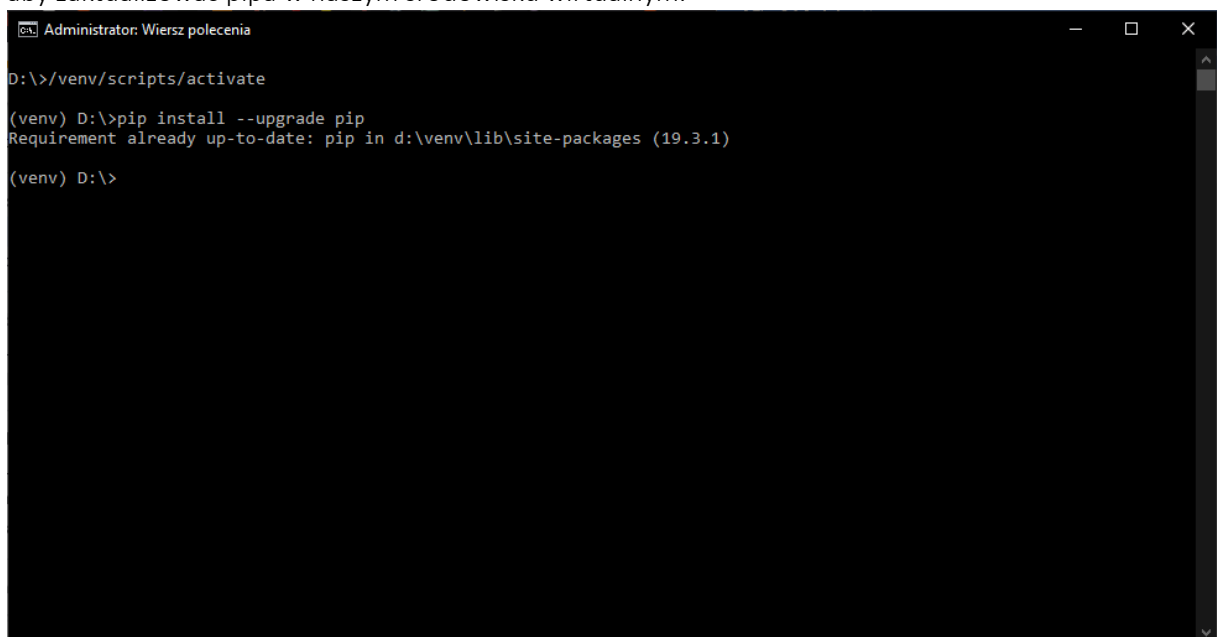


```
Administrator: Wiersz polecenia
D:\>/venv/scripts/activate
(venv) D:\>
```

23. Wpiszmy komendę

pip install --upgrade pip

aby zaktualizować pip w naszym środowisku wirtualnym.



```
Administrator: Wiersz polecenia
D:\>/venv/scripts/activate
(venv) D:\>pip install --upgrade pip
Requirement already up-to-date: pip in d:\venv\lib\site-packages (19.3.1)
(venv) D:\>
```

24. Aby sprawdzić listę zainstalowanych pakietów pythona możemy wpisać polecenie

pip list

```
Administrator: Wiersz polecenia
tensorboard                2.0.2
tensorflow                 2.0.0
tensorflow-estimator       2.0.1
termcolor                 1.1.0
terminado                 0.8.2
testpath                  0.4.2
toolz                     0.10.0
tornado                   6.0.3
tqdm                      4.36.1
traitlets                 4.3.3
unicodectsv               0.14.1
urllib3                   1.24.2
virtualenv                16.7.9
wcwidth                   0.1.7
webencodings              0.5.1
Werkzeug                  0.16.0
wheel                     0.33.6
widgetsnbextension        3.5.1
win-inet-pton             1.1.0
win-unicode-console       0.5
wincertstore              0.2
wrapt                     1.11.2
xlrd                      1.2.0
XlsxWriter                1.2.1
xlwings                   0.15.10
xlwt                      1.3.0
zict                      1.0.0
zipp                      0.6.0

(venv) D:\>
```

Na screenie jest ich dość dużo, ponieważ twórca tych screenów ma zainstalowaną dystrybucję Pythona – **Anaconda**. Aby wyjść z wirtualnego środowiska musimy wpisać komendę **deactivate**.

UWAGA! Polecam sprawdzić zainstalowane biblioteki! Ponownie wpisanie komendy **pip install --upgrade**, które znajdują się w następnym punkcie może spowodować usunięcie zainstalowanej już biblioteki!

25. Teraz nastąpi instalacja bibliotek, których będziemy używać w naszej prezentacji, a więc:

A) Aby zainstalować **Tensorflow** wpisujemy komendę

pip install --upgrade tensorflow

```
Administrator: Wiersz polecenia
Requirement already satisfied, skipping upgrade: werkzeug>=0.11.15 in c:\users\mwilc\anaconda3\lib\site-packages (from tensorflow) (0.16.0)
Requirement already satisfied, skipping upgrade: markdown>=2.6.8 in c:\users\mwilc\anaconda3\lib\site-packages (from tensorflow) (3.1.1)
Requirement already satisfied, skipping upgrade: requests>=2.21.0 in c:\users\mwilc\anaconda3\lib\site-packages (from tensorflow) (2.22.0)
Requirement already satisfied, skipping upgrade: google-auth-oauthlib<0.5,>=0.4.1 in c:\users\mwilc\anaconda3\lib\site-packages (from tensorflow) (0.4.1)
Requirement already satisfied, skipping upgrade: cachetools<5.0,>=2.0.0 in c:\users\mwilc\anaconda3\lib\site-packages (from google-auth-oauthlib<0.5,>=0.4.1->tensorflow) (4.0.0)
Requirement already satisfied, skipping upgrade: rsa<4.1,>=3.1.4 in c:\users\mwilc\anaconda3\lib\site-packages (from google-auth<2,>=1.6.3->tensorflow) (4.0)
Requirement already satisfied, skipping upgrade: pyasn1-modules>=0.2.1 in c:\users\mwilc\anaconda3\lib\site-packages (from google-auth<2,>=1.6.3->tensorflow) (0.2.7)
Requirement already satisfied, skipping upgrade: certifi>=2017.4.17 in c:\users\mwilc\anaconda3\lib\site-packages (from requests>=2.21.0->tensorflow) (2019.9.11)
Requirement already satisfied, skipping upgrade: chardet<3.1.0,>=3.0.2 in c:\users\mwilc\anaconda3\lib\site-packages (from requests>=2.21.0->tensorflow) (3.0.4)
Requirement already satisfied, skipping upgrade: urllib3!=1.25.0,!1.25.1,<1.26,>=1.21.1 in c:\users\mwilc\anaconda3\lib\site-packages (from requests>=2.21.0->tensorflow) (1.24.2)
Requirement already satisfied, skipping upgrade: idna<2.9,>=2.5 in c:\users\mwilc\anaconda3\lib\site-packages (from requests>=2.21.0->tensorflow) (2.8)
Requirement already satisfied, skipping upgrade: requests-oauthlib>=0.7.0 in c:\users\mwilc\anaconda3\lib\site-packages (from google-auth-oauthlib<0.5,>=0.4.1->tensorflow) (1.3.0)
Requirement already satisfied, skipping upgrade: pyasn1>=0.1.3 in c:\users\mwilc\anaconda3\lib\site-packages (from rsa<4.1,>=3.1.4->google-auth<2,>=1.6.3->tensorflow) (0.4.8)
Requirement already satisfied, skipping upgrade: oauthlib>=3.0.0 in c:\users\mwilc\anaconda3\lib\site-packages (from requests-oauthlib>=0.7.0->google-auth-oauthlib<0.5,>=0.4.1->tensorflow) (3.1.0)

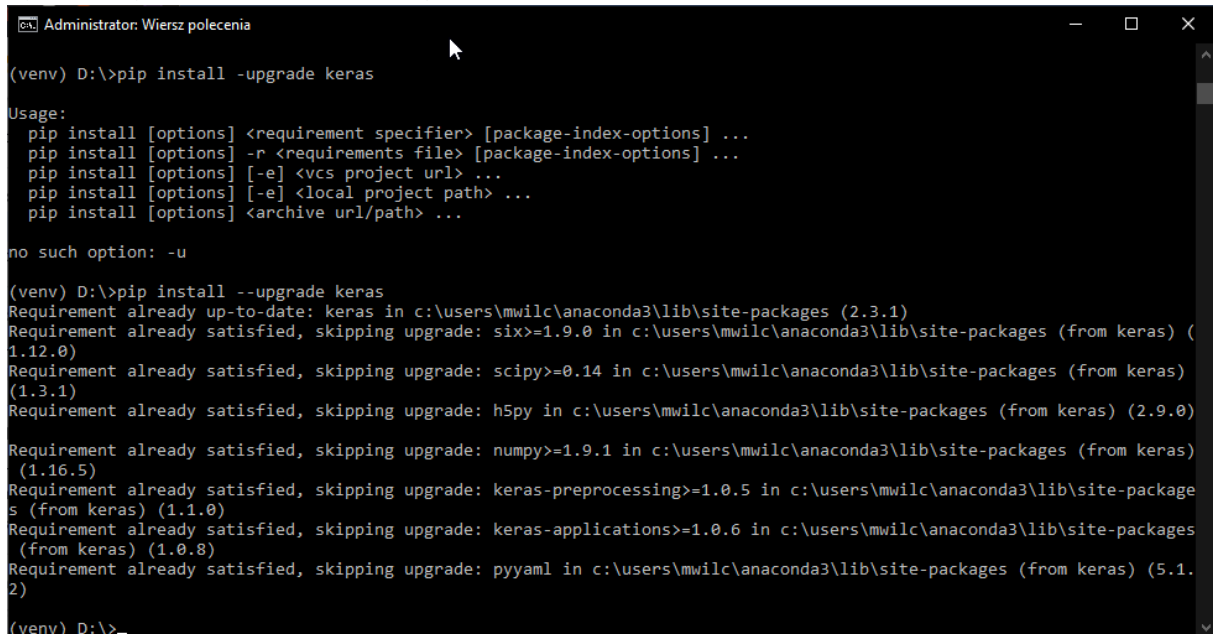
(venv) D:\>
```

B) Czekamy, aż instalacja się zakończy pracę.

C) Teraz instalujemy **Keras'a**. Wpisujemy komendę

pip install --upgrade keras

D) Czekamy na zakończenie instalacji.



```
(venv) D:\>pip install --upgrade keras

Usage:
  pip install [options] <requirement specifier> [package-index-options] ...
  pip install [options] -r <requirements file> [package-index-options] ...
  pip install [options] [-e] <vcs project url> ...
  pip install [options] [-e] <local project path> ...
  pip install [options] <archive url/path> ...

no such option: -u

(venv) D:\>pip install --upgrade keras
Requirement already up-to-date: keras in c:\users\mwilc\anaconda3\lib\site-packages (2.3.1)
Requirement already satisfied, skipping upgrade: six>=1.9.0 in c:\users\mwilc\anaconda3\lib\site-packages (from keras) (1.12.0)
Requirement already satisfied, skipping upgrade: scipy>=0.14 in c:\users\mwilc\anaconda3\lib\site-packages (from keras) (1.3.1)
Requirement already satisfied, skipping upgrade: h5py in c:\users\mwilc\anaconda3\lib\site-packages (from keras) (2.9.0)
Requirement already satisfied, skipping upgrade: numpy>=1.9.1 in c:\users\mwilc\anaconda3\lib\site-packages (from keras) (1.16.5)
Requirement already satisfied, skipping upgrade: keras-preprocessing>=1.0.5 in c:\users\mwilc\anaconda3\lib\site-packages (from keras) (1.1.0)
Requirement already satisfied, skipping upgrade: keras-applications>=1.0.6 in c:\users\mwilc\anaconda3\lib\site-packages (from keras) (1.0.8)
Requirement already satisfied, skipping upgrade: pyyaml in c:\users\mwilc\anaconda3\lib\site-packages (from keras) (5.1.2)

(venv) D:\>
```

E) Powtarzamy kroki **A) i B)**, ale dla bibliotek **Numpy, SciPy, Matplotlib, Scikit-learn, Pandas**

A więc za wpisujemy komendę

pip install --upgrade X

za **X** podstawiamy kolejno **Numpy, SciPy, Matplotlib, Scikit-learn i Pandas**

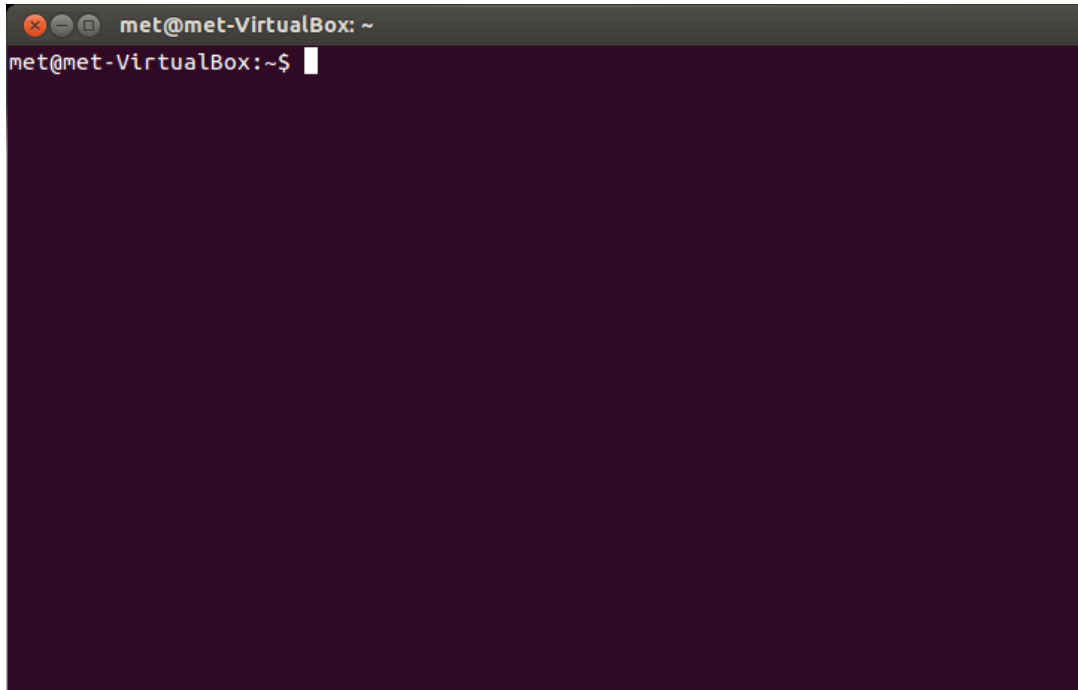
Gratuluje! Instalacja została zakończona! Pamiętaj, aby uruchamiać programy w naszym właśnie stworzonym środowisku wirtualnym!

Instalacja na Ubuntu 12.04

1. Włączamy terminal systemowy skrótem klawiszowym

Control + Alt + T

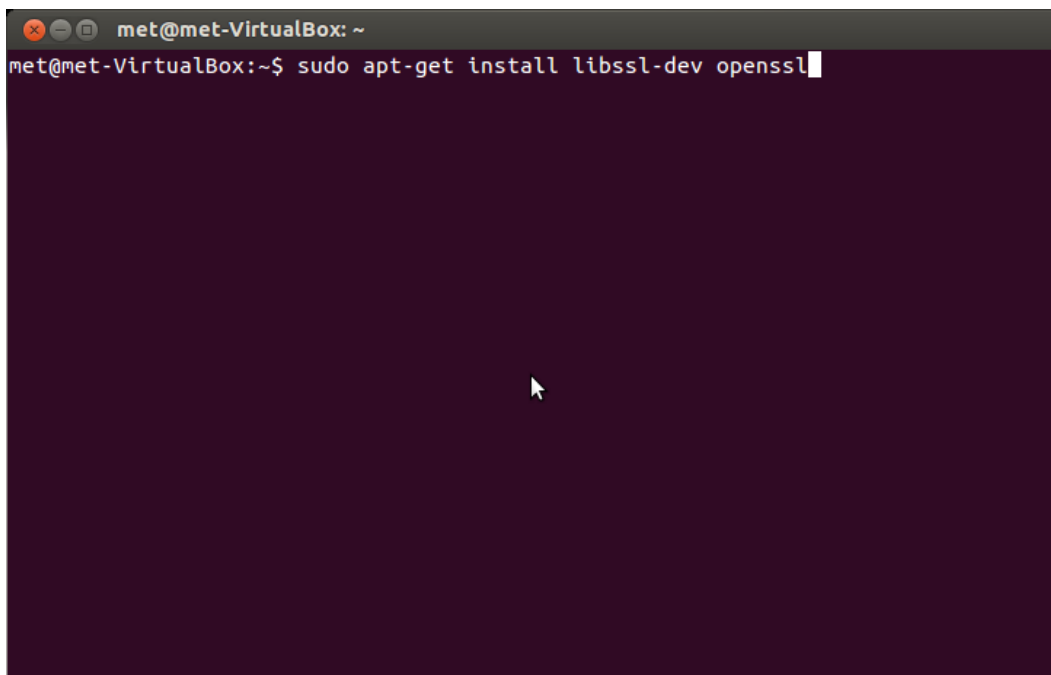
albo uruchamiamy go z menu.



2. A więc w **terminalu** wpisujemy komendę

sudo apt-get install libssl-dev openssl

i klikamy **ENTER**.



Podajemy hasło użytkownika. Gdy pojawi się komunikat, czy kontynuować wpisujemy **T** i klikamy ponownie **ENTER**.

```
met@met-VirtualBox: ~  
met@met-VirtualBox:~$ sudo apt-get install libssl-dev openssl  
[sudo] password for met:  
Czytanie list pakietów... Gotowe  
Budowanie drzewa zależności  
Odczyt informacji o stanie... Gotowe  
Zostaną zainstalowane następujące dodatkowe pakiety:  
  libssl-doc libssl1.0.0  
Zostaną zainstalowane następujące NOWE pakiety:  
  libssl-dev libssl-doc  
Następujące pakiety zostaną zaktualizowane:  
  libssl1.0.0 openssl  
2 aktualizowanych, 2 nowo instalowanych, 0 usuwanych i 304 nieaktualizowanych.  
Konieczne pobranie 4202 kB archiwów.  
Po tej operacji zostanie dodatkowo użyte 7809 kB miejsca na dysku.  
Kontynuować [T/n]? T
```

3. Czekamy, aż **libssl** się zainstaluje.

```
met@met-VirtualBox: ~  
Pobrano 4202 kB w 1s (3151 kB/s)  
Prekonfiguracja pakietów ...  
(Odczytywanie bazy danych ... 211589 plików i katalogów obecnie zainstalowanych.  
)  
Przygotowanie do zastąpienia libssl1.0.0 1.0.1-4ubuntu5.16 (wykorzystując .../li  
bssl1.0.0_1.0.1-4ubuntu5.39_amd64.deb) ...  
Rozpakowanie pakietu zastępującego libssl1.0.0 ...  
Konfigurowanie libssl1.0.0 (1.0.1-4ubuntu5.39) ...  
Przetwarzanie wyzwalaczy dla libc-bin...  
ldconfig deferred processing now taking place  
(Odczytywanie bazy danych ... 211589 plików i katalogów obecnie zainstalowanych.  
)  
Przygotowanie do zastąpienia openssl 1.0.1-4ubuntu5.16 (wykorzystując .../openss  
l_1.0.1-4ubuntu5.39_amd64.deb) ...  
Rozpakowanie pakietu zastępującego openssl ...  
Wybieranie wcześniej niewybranego pakietu libssl-dev.  
Rozpakowanie libssl-dev (z .../libssl-dev_1.0.1-4ubuntu5.39_amd64.deb) ...  
Wybieranie wcześniej niewybranego pakietu libssl-doc.  
Rozpakowanie libssl-doc (z .../libssl-doc_1.0.1-4ubuntu5.39_all.deb) ...  
Przetwarzanie wyzwalaczy dla man-db...  
Konfigurowanie openssl (1.0.1-4ubuntu5.39) ...  
Konfigurowanie libssl-dev (1.0.1-4ubuntu5.39) ...  
Konfigurowanie libssl-doc (1.0.1-4ubuntu5.39) ...  
met@met-VirtualBox:~$
```

4. Teraz nastąpi instalacja pythona w wersji 3.5 musimy więc dodajemy dwa klucze PPA. Kolejno komendami

sudo add-apt-repository ppa:deadsnakes/ppa
sudo add-apt-repository ppa:fkruhl/deadsnakes

```
met@met-VirtualBox: ~  
met@met-VirtualBox:~$ sudo add-apt-repository ppa:fkruhl/deadsnakes  
[sudo] password for met:  
You are about to add the following PPA to your system:  
This repository is kept for historical purposes, but NOT UPDATED. Please use the new repository at  
https://launchpad.net/~deadsnakes/+archive/ubuntu/ppa  
instead!  
More info: https://launchpad.net/~fkruhl/+archive/ubuntu/deadsnakes  
Press [ENTER] to continue or ctrl-c to cancel adding it  
  
gpg: zbiór kluczy „,/tmp/tmpBdqJN8/secring.gpg” został utworzony  
gpg: zbiór kluczy „,/tmp/tmpBdqJN8/pubring.gpg” został utworzony  
gpg: zapytanie o klucz DB82666C z hkp serwera keyserver.ubuntu.com  
gpg: /tmp/tmpBdqJN8/trustdb.gpg: baza zaufania utworzona  
gpg: klucz DB82666C: zaimportowano klucz publiczny „Launchpad Old Python Versions”  
gpg: Ogółem przetworzonych kluczy: 1  
gpg: dołączono do zbioru: 1 (RSA: 1)  
OK  
met@met-VirtualBox:~$ sudo add-apt-repository ppa:deadsnakes/ppa
```

Gdy terminal poprosi nas o wciśnięcie **ENTER** wciskamy go.

5. Teraz musimy wpisać komendę

sudo apt-get update

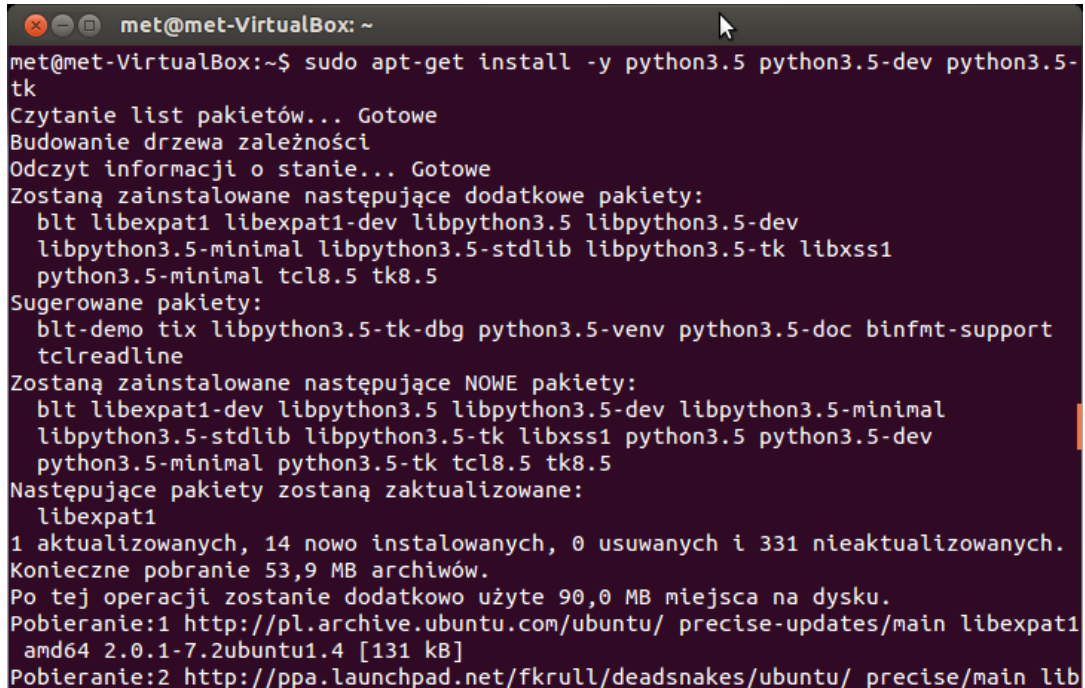
aby nasze klucze zostały zaktualizowane. Wpisujemy komendę i wciskamy **ENTER**.

```
met@met-VirtualBox: ~  
Stary http://pl.archive.ubuntu.com precise/universe Translation-en  
Stary http://pl.archive.ubuntu.com precise-updates/main Translation-en  
Stary http://pl.archive.ubuntu.com precise-updates/multiverse Translation-en  
Stary http://pl.archive.ubuntu.com precise-updates/restricted Translation-en  
Stary http://pl.archive.ubuntu.com precise-updates/universe Translation-en  
Stary http://pl.archive.ubuntu.com precise-backports/main Translation-en  
Stary http://pl.archive.ubuntu.com precise-backports/multiverse Translation-en  
Stary http://pl.archive.ubuntu.com precise-backports/restricted Translation-en  
Stary http://pl.archive.ubuntu.com precise-backports/universe Translation-en  
Pobieranie:10 http://ppa.launchpad.net precise/main amd64 Packages [14 B]  
Ign. http://extras.ubuntu.com precise/main Translation-pl_PL  
Ign. http://extras.ubuntu.com precise/main Translation-pl  
Ign. http://extras.ubuntu.com precise/main Translation-en  
Pobieranie:11 http://ppa.launchpad.net precise/main i386 Packages [14 B]  
Pobieranie:12 http://ppa.launchpad.net precise/main TranslationIndex [193 B]  
Pobieranie:13 http://ppa.launchpad.net precise/main Sources [5161 B]  
Pobieranie:14 http://ppa.launchpad.net precise/main amd64 Packages [19,8 kB]  
Pobieranie:15 http://ppa.launchpad.net precise/main i386 Packages [19,8 kB]  
Pobieranie:16 http://ppa.launchpad.net precise/main TranslationIndex [202 B]  
Pobieranie:17 http://ppa.launchpad.net precise/main Translation-en [14 B]  
Pobieranie:18 http://ppa.launchpad.net precise/main Translation-en [6955 B]  
Pobrano 874 kB w 2s (397 kB/s)  
Czytanie list pakietów... Gotowe  
met@met-VirtualBox:~$
```

6. Teraz nastąpi instalacja **pythona 3.5**. Wpisujemy więc komendę

`sudo apt-get install -y python3.5 python3.5-dev python3.5-tk`

i wciskamy **ENTER**.

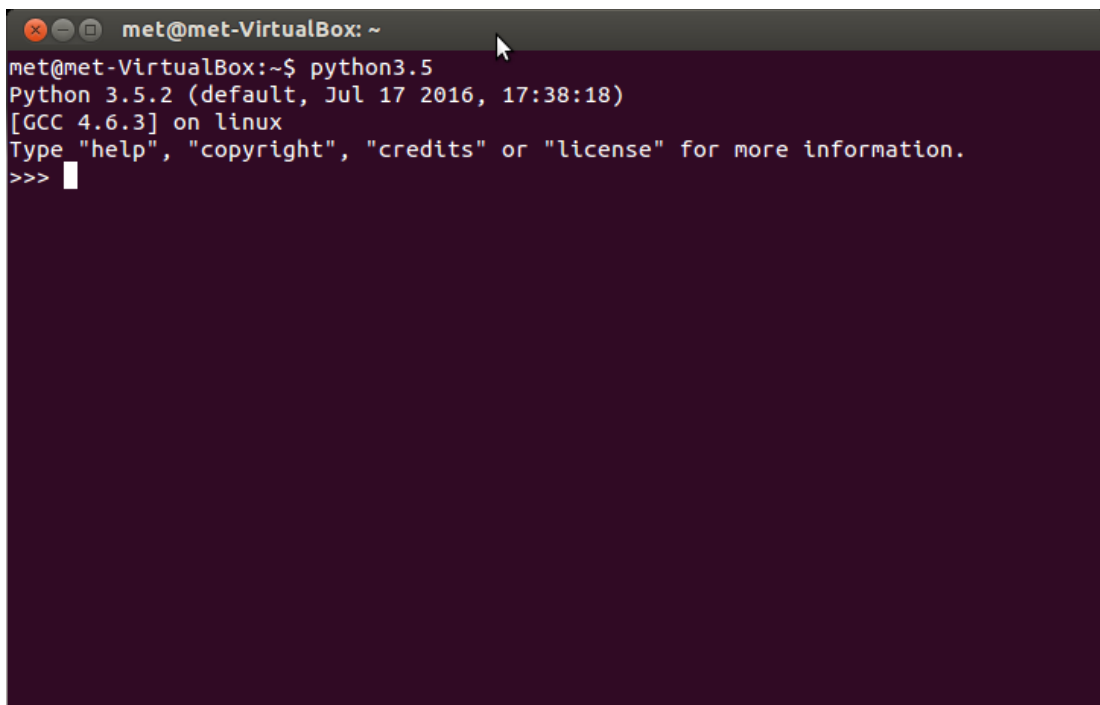


```
met@met-VirtualBox: ~  
met@met-VirtualBox:~$ sudo apt-get install -y python3.5 python3.5-dev python3.5-tk  
Czytanie list pakietów... Gotowe  
Budowanie drzewa zależności  
Odczyt informacji o stanie... Gotowe  
Zostaną zainstalowane następujące dodatkowe pakiety:  
  blt libexpat1 libexpat1-dev libpython3.5 libpython3.5-dev  
  libpython3.5-minimal libpython3.5-stdlib libpython3.5-tk libxss1  
  python3.5-minimal tcl8.5 tk8.5  
Sugerowane pakiety:  
  blt-demo tix libpython3.5-tk-dbg python3.5-venv python3.5-doc binfmt-support  
  tclreadline  
Zostaną zainstalowane następujące NOWE pakiety:  
  blt libexpat1-dev libpython3.5 libpython3.5-dev libpython3.5-minimal  
  libpython3.5-stdlib libpython3.5-tk libxss1 python3.5 python3.5-dev  
  python3.5-minimal python3.5-tk tcl8.5 tk8.5  
Następujące pakiety zostaną zaktualizowane:  
  libexpat1  
1 aktualizowanych, 14 nowo instalowanych, 0 usuwanych i 331 nieaktualizowanych.  
Konieczne pobranie 53,9 MB archiwów.  
Po tej operacji zostanie dodatkowo użyte 90,0 MB miejsca na dysku.  
Pobieranie:1 http://pl.archive.ubuntu.com/ubuntu/ precise-updates/main libexpat1  
amd64 2.0.1-7.2ubuntu1.4 [131 kB]  
Pobieranie:2 http://ppa.launchpad.net/fkrull/deadsnakes/ubuntu/ precise/main lib
```

7. Teraz po wypisaniu komendy

`python3.5`

w terminalu, powinienn się nam odpalić python w wersji 3.5.2.



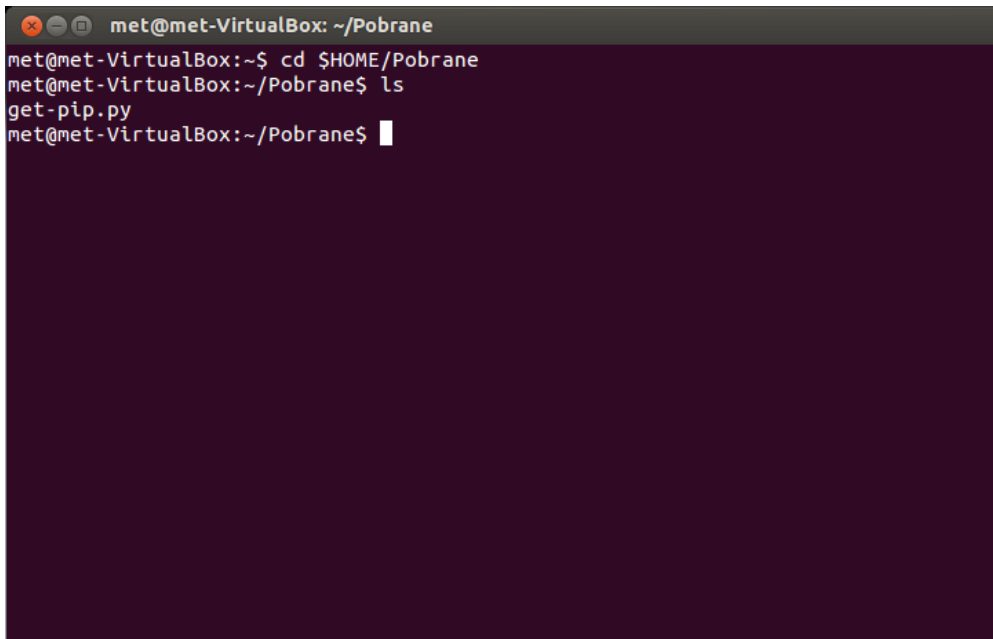
```
met@met-VirtualBox: ~  
met@met-VirtualBox:~$ python3.5  
Python 3.5.2 (default, Jul 17 2016, 17:38:18)  
[GCC 4.6.3] on linux  
Type "help", "copyright", "credits" or "license" for more information.  
>>> █
```

8. Teraz zainstalujemy **pip'a**! Wchodzimy na stronę

<https://bootstrap.pypa.io/get-pip.py>

Ściągamy plik **get-pip.py**. Najlepiej go umieścić w jakimś łatwo dostępnym katalogu jak **/\$HOME/Pobrane**. Jeżeli nie uruchomił się menadżer pobierania na twojej przeglądarce to [odsylam do punktu 7 w instrukcji instalacji dla Windows](#). Po utworzeniu pliku proszę wrócić do tego miejsca.

9. Przechodzimy do katalogu, gdzie jest plik **get-pip.py**, u mnie jest to katalog **\$HOME/Pobrane**.

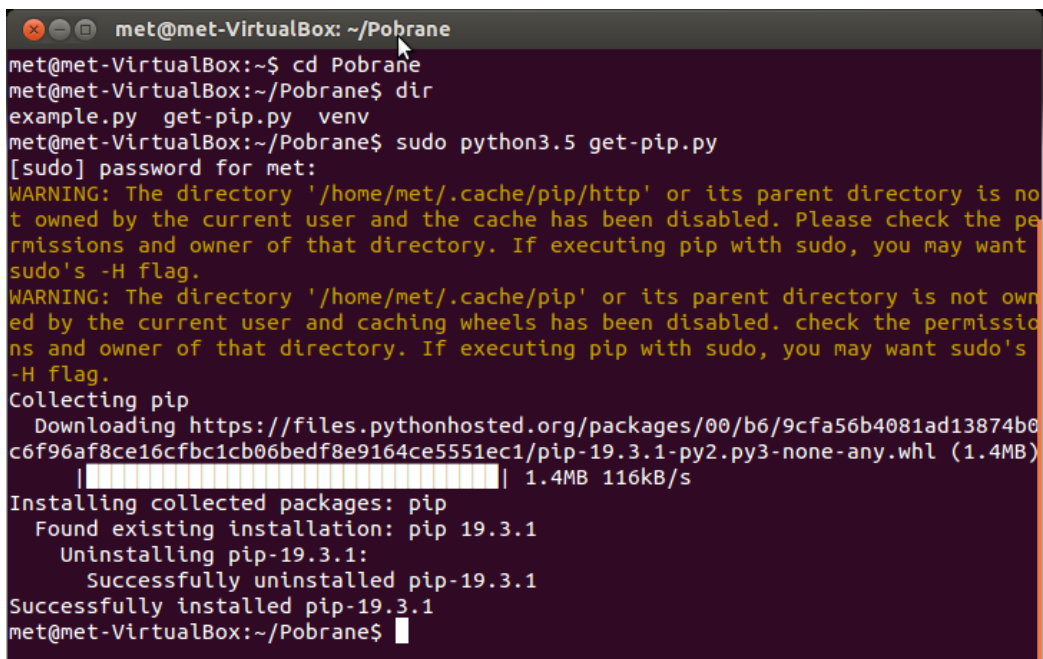


```
met@met-VirtualBox: ~/Pobrane
met@met-VirtualBox:~$ cd $HOME/Pobrane
met@met-VirtualBox:~/Pobrane$ ls
get-pip.py
met@met-VirtualBox:~/Pobrane$
```

10. Wpisujemy komendę

sudo python3.5 get-pip.py

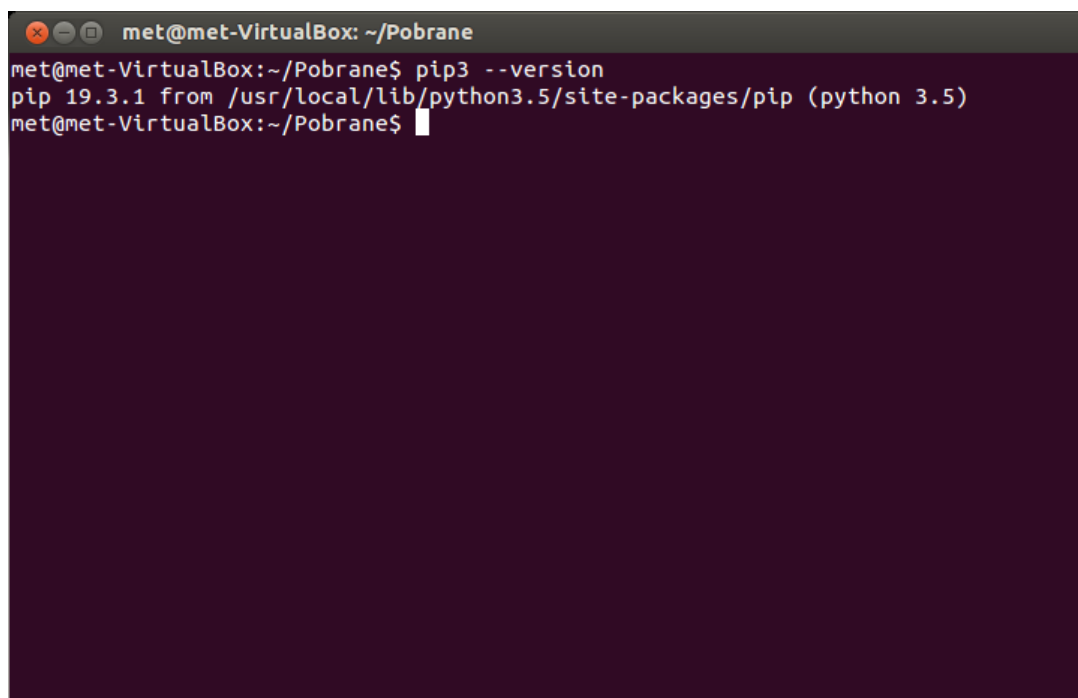
i czekamy, aż python zainstaluje pakiet.



```
met@met-VirtualBox: ~/Pobrane
met@met-VirtualBox:~$ cd Pobrane
met@met-VirtualBox:~/Pobrane$ dir
example.py  get-pip.py  venv
met@met-VirtualBox:~/Pobrane$ sudo python3.5 get-pip.py
[sudo] password for met:
WARNING: The directory '/home/met/.cache/pip/http' or its parent directory is not owned by the current user and the cache has been disabled. Please check the permissions and owner of that directory. If executing pip with sudo, you may want sudo's -H flag.
WARNING: The directory '/home/met/.cache/pip' or its parent directory is not owned by the current user and caching wheels has been disabled. check the permissions and owner of that directory. If executing pip with sudo, you may want sudo's -H flag.
Collecting pip
  Downloading https://files.pythonhosted.org/packages/00/b6/9cfa56b4081ad13874b6c6f96af8ce16cfbc1cb06bedf8e9164ce5551ec1/pip-19.3.1-py2.py3-none-any.whl (1.4MB)
    |#####| 1.4MB 116kB/s
Installing collected packages: pip
  Found existing installation: pip 19.3.1
  Uninstalling pip-19.3.1:
    Successfully uninstalled pip-19.3.1
Successfully installed pip-19.3.1
met@met-VirtualBox:~/Pobrane$
```

11. Sprawdzamy wersję **pip'a** komendą

pip3 --version



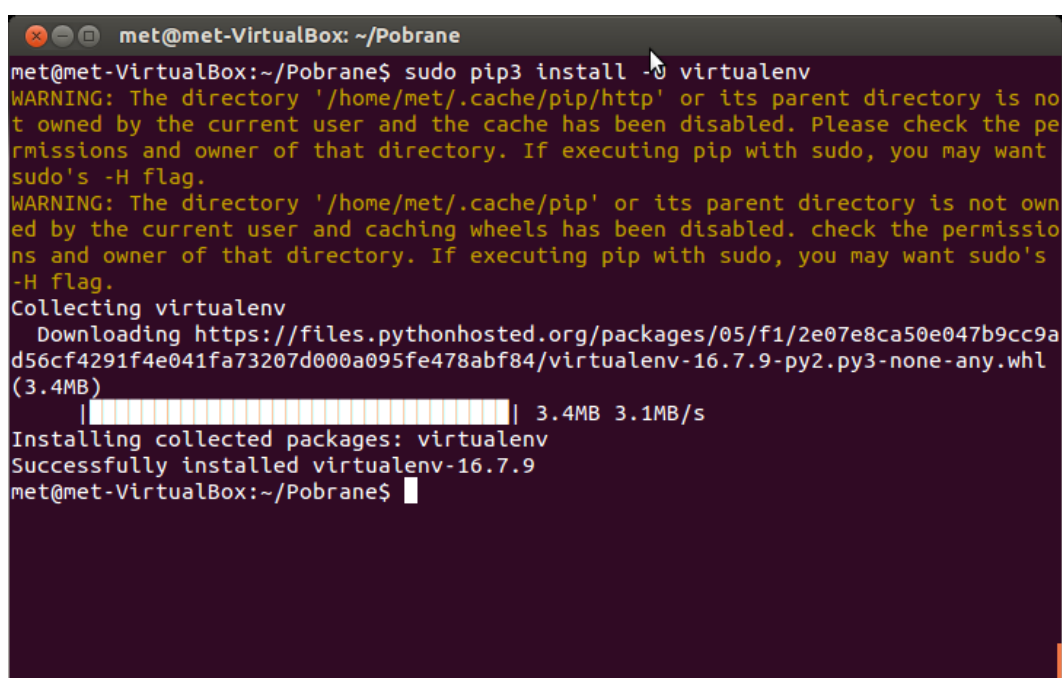
```
met@met-VirtualBox: ~/Pobrane
met@met-VirtualBox:~/Pobrane$ pip3 --version
pip 19.3.1 from /usr/local/lib/python3.5/site-packages/pip (python 3.5)
met@met-VirtualBox:~/Pobrane$
```

Jeżeli pip został zainstalowany pomyślnie, to powinna na ekranie pojawić się wersja pip'a, katalog, gdzie został zainstalowany oraz dla jakiej wersji pythona3.

12. Teraz nastąpi instalacja środowisk wirtualnych. Wpisujemy więc

sudo pip3 install -U virtualenv

i klikamy **ENTER**

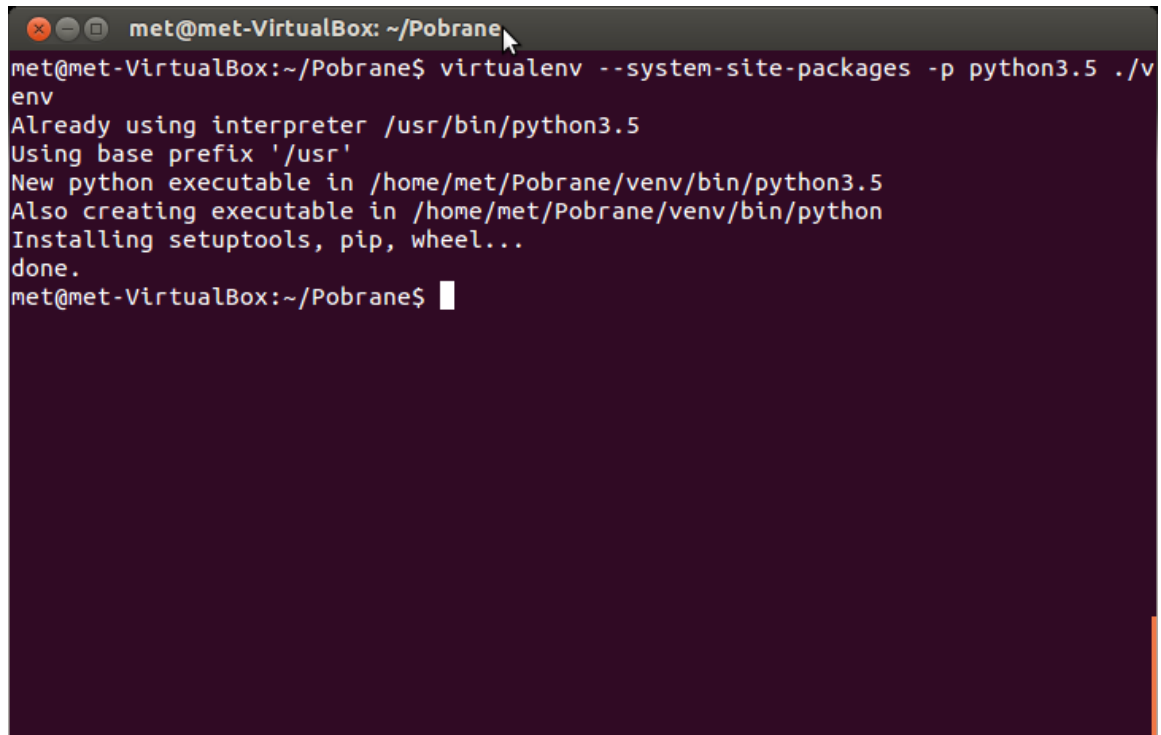


```
met@met-VirtualBox: ~/Pobrane
met@met-VirtualBox:~/Pobrane$ sudo pip3 install -U virtualenv
WARNING: The directory '/home/met/.cache/pip/http' or its parent directory is not owned by the current user and the cache has been disabled. Please check the permissions and owner of that directory. If executing pip with sudo, you may want sudo's -H flag.
WARNING: The directory '/home/met/.cache/pip' or its parent directory is not owned by the current user and caching wheels has been disabled. check the permissions and owner of that directory. If executing pip with sudo, you may want sudo's -H flag.
Collecting virtualenv
  Downloading https://files.pythonhosted.org/packages/05/f1/2e07e8ca50e047b9cc9ad56cf4291f4e041fa73207d000a095fe478abf84/virtualenv-16.7.9-py2.py3-none-any.whl (3.4MB)
    |#####| 3.4MB 3.1MB/s
Installing collected packages: virtualenv
Successfully installed virtualenv-16.7.9
met@met-VirtualBox:~/Pobrane$
```

13. Tworzymy teraz jedno środowisko wirtualne komendą

`virtualenv --system-site-packages -p python3.5 ./venv`

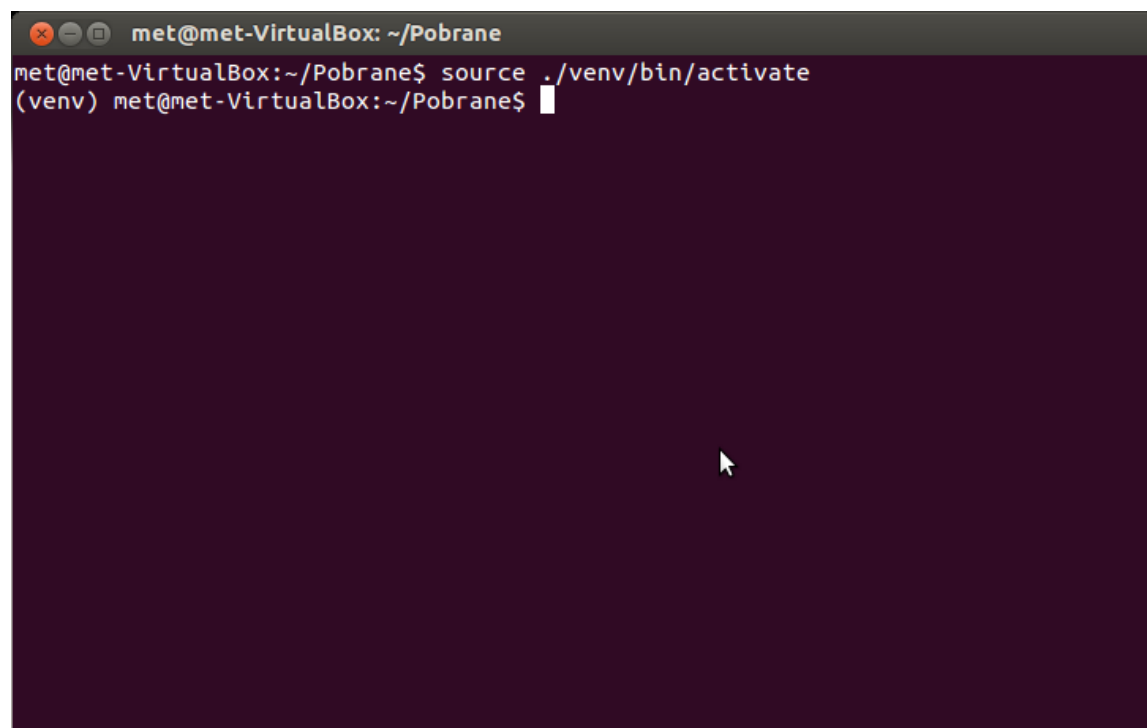
`./venv` to katalog w którym będzie znajdować się środowisko

A terminal window titled 'met@met-VirtualBox: ~/Pobrane' showing the execution of the command 'virtualenv --system-site-packages -p python3.5 ./venv'. The output indicates that the environment is already using the interpreter /usr/bin/python3.5, the base prefix is '/usr', and new python executables are being created in /home/met/Pobrane/venv/bin. It also shows the installation of setuptools, pip, and wheel. The prompt returns to 'met@met-VirtualBox:~/Pobrane\$' with a cursor.

```
met@met-VirtualBox: ~/Pobrane
met@met-VirtualBox:~/Pobrane$ virtualenv --system-site-packages -p python3.5 ./venv
env
Already using interpreter /usr/bin/python3.5
Using base prefix '/usr'
New python executable in /home/met/Pobrane/venv/bin/python3.5
Also creating executable in /home/met/Pobrane/venv/bin/python
Installing setuptools, pip, wheel...
done.
met@met-VirtualBox:~/Pobrane$
```

14. Aby odpalić środowisko używamy komendy

`source ./venv/bin/activate`

A terminal window titled 'met@met-VirtualBox: ~/Pobrane' showing the execution of the command 'source ./venv/bin/activate'. The prompt changes from 'met@met-VirtualBox:~/Pobrane\$' to '(venv) met@met-VirtualBox:~/Pobrane\$', indicating the virtual environment is now active. A mouse cursor is visible in the terminal.

```
met@met-VirtualBox: ~/Pobrane
met@met-VirtualBox:~/Pobrane$ source ./venv/bin/activate
(venv) met@met-VirtualBox:~/Pobrane$
```

18. Od tego punktu wszystko biblioteki instaluje się **tak samo** jak w **systemie windows**. Polecenie

pip install --upgrade X

instaluje biblioteki. Gdzie **X** to nazwa biblioteki.

Odsyłam do instrukcji dla systemu Windows
Punkt 22

Aby **wyjść** ze środowiska wirtualnego wpisujemy komendę

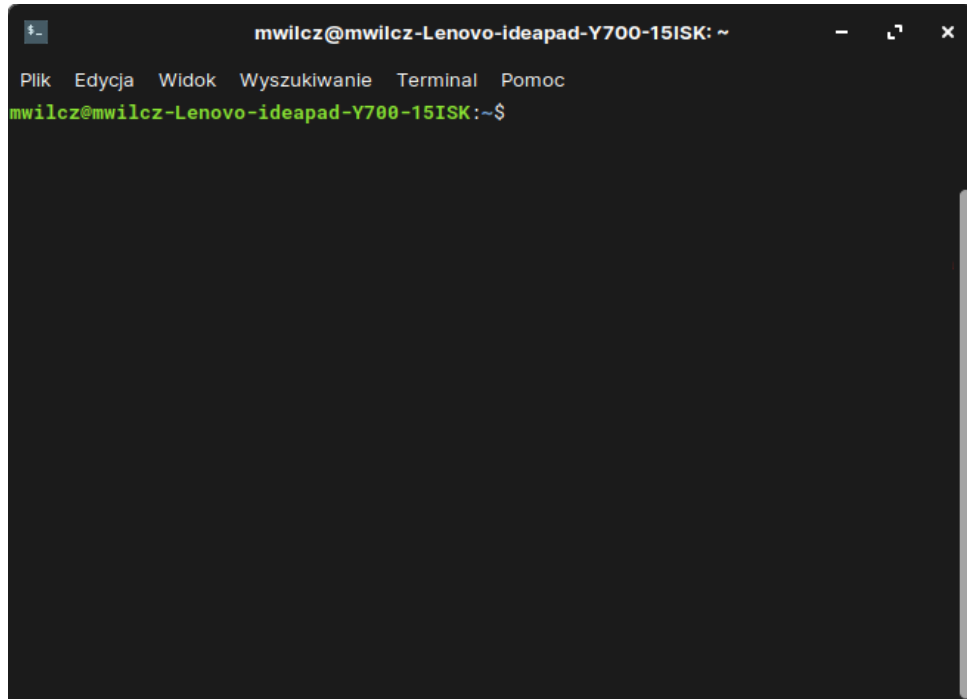
Deactivate

Ubuntu (19.10)

1. Włączamy terminal systemowy skrótem klawiszowym

Control + Alt + T

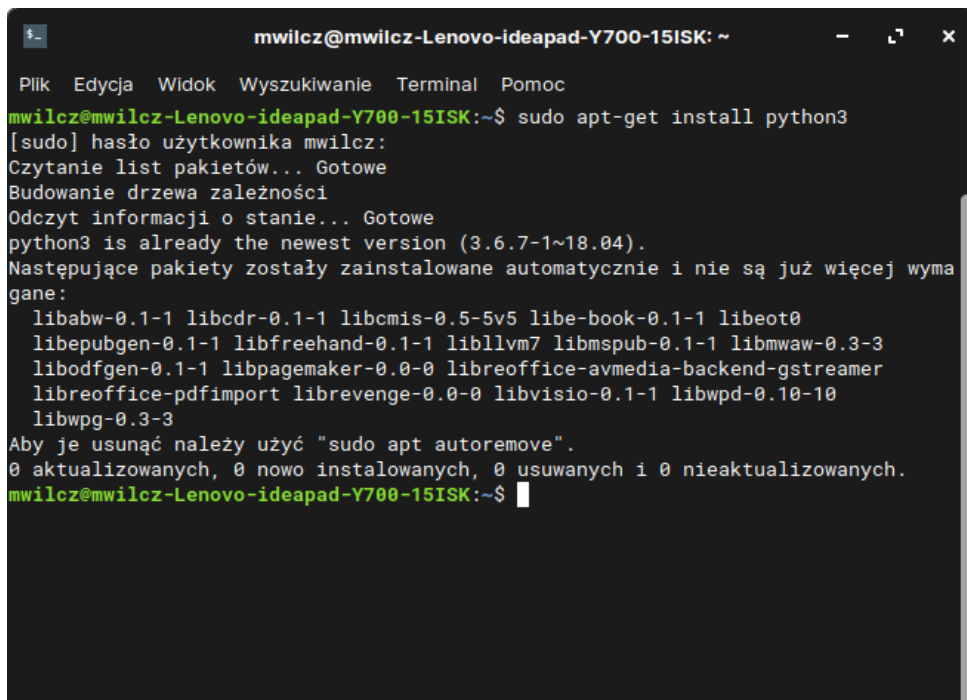
albo uruchamiamy go z menu.



2. Wpisujemy komendę

sudo apt-get install python3

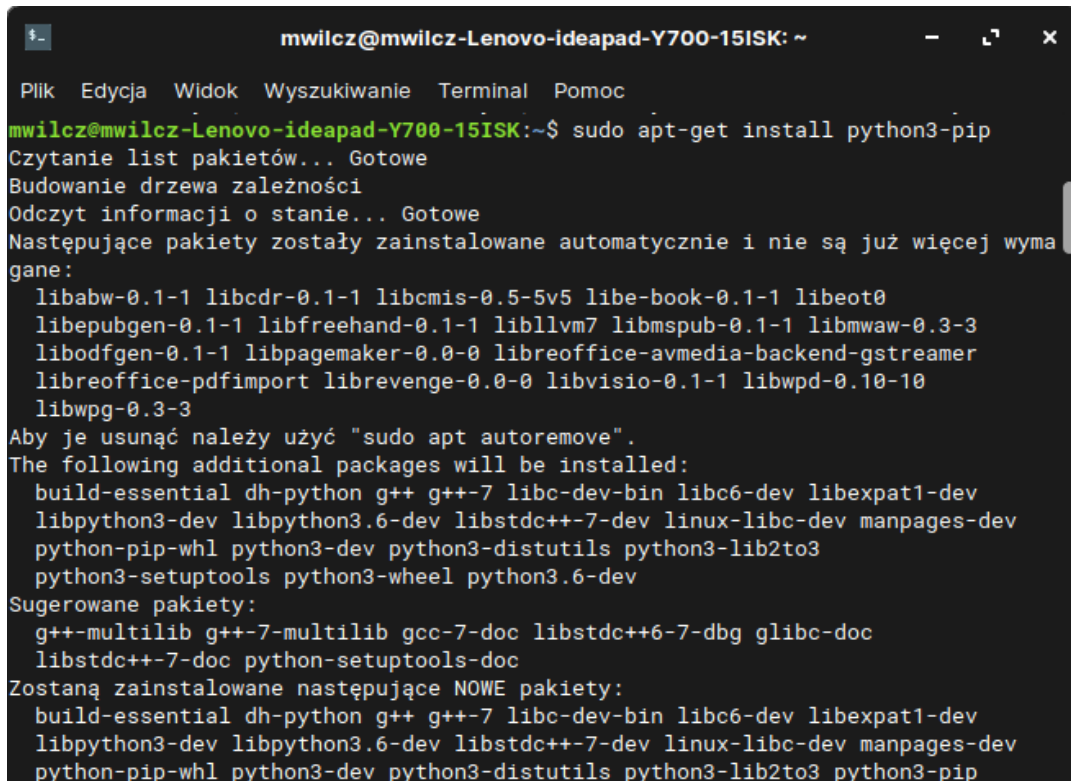
i wciskamy klawisz **ENTER**.



3. Po zainstalowaniu **pythona3** wpisujemy w terminalu komendę

sudo apt-get install python3-pip

i wciskamy klawisz **ENTER**.



```
mwilcz@mwilcz-Lenovo-ideapad-Y700-15ISK: ~  
Plik Edycja Widok Wyszukiwanie Terminal Pomoc  
mwilcz@mwilcz-Lenovo-ideapad-Y700-15ISK:~$ sudo apt-get install python3-pip  
Czytanie list pakietów... Gotowe  
Budowanie drzewa zależności  
Odczyt informacji o stanie... Gotowe  
Następujące pakiety zostały zainstalowane automatycznie i nie są już więcej wymagane:  
libabw-0.1-1 libcdr-0.1-1 libcmis-0.5-5v5 libe-book-0.1-1 libeot0  
libepubgen-0.1-1 libfreehand-0.1-1 libllvm7 libmspub-0.1-1 libmwaw-0.3-3  
libodfgen-0.1-1 libpagemaker-0.0-0 libreoffice-avmedia-backend-gstreamer  
libreoffice-pdfimport libvenge-0.0-0 libvisio-0.1-1 libwpd-0.10-10  
libwpg-0.3-3  
Aby je usunąć należy użyć "sudo apt autoremove".  
The following additional packages will be installed:  
build-essential dh-python g++ g++-7 libc-dev-bin libc6-dev libexpat1-dev  
libpython3-dev libpython3.6-dev libstdc++-7-dev linux-libc-dev manpages-dev  
python-pip-whl python3-dev python3-distutils python3-lib2to3  
python3-setuptools python3-wheel python3.6-dev  
Sugerowane pakiety:  
g++-multilib g++-7-multilib gcc-7-doc libstdc++6-7-dbg glibc-doc  
libstdc++-7-doc python-setuptools-doc  
Zostaną zainstalowane następujące NOWE pakiety:  
build-essential dh-python g++ g++-7 libc-dev-bin libc6-dev libexpat1-dev  
libpython3-dev libpython3.6-dev libstdc++-7-dev linux-libc-dev manpages-dev  
python-pip-whl python3-dev python3-distutils python3-lib2to3 python3-pip
```

4. Po zainstalowaniu **pip'a** możemy przejść do instalacji wirtualnego środowiska. Jednak dalsza instalacja różni się tylko jedną komendą. Mianowicie musimy wpisywać

python3

zamiast **python3.5**.

Odsyłam więc do instrukcji dla Ubuntu 12.04

Punkt 12

Przykładowy program do testowania

1. Ściągamy zipa ze strony GitHub

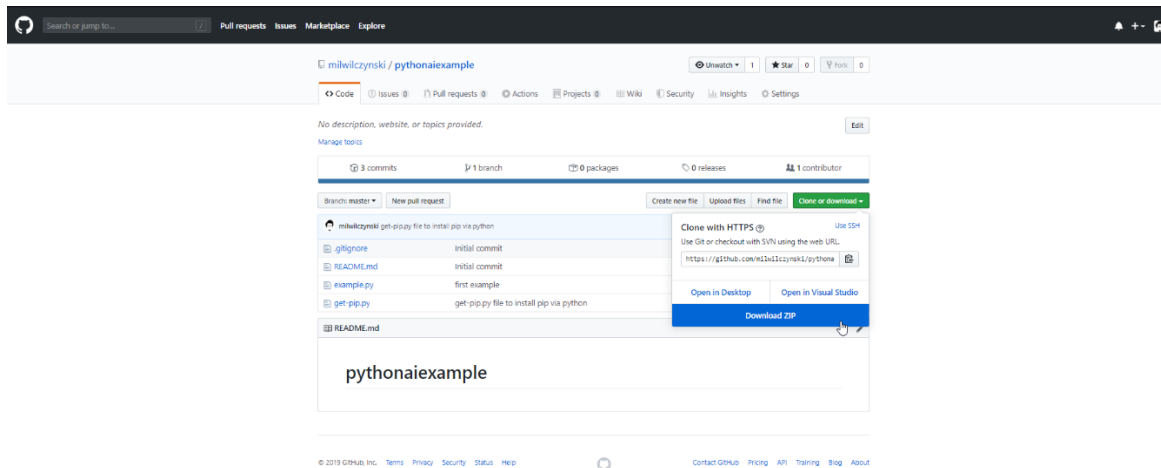
<https://github.com/milwilczynski/pythonaexample>

Ubuntu

Strona 29

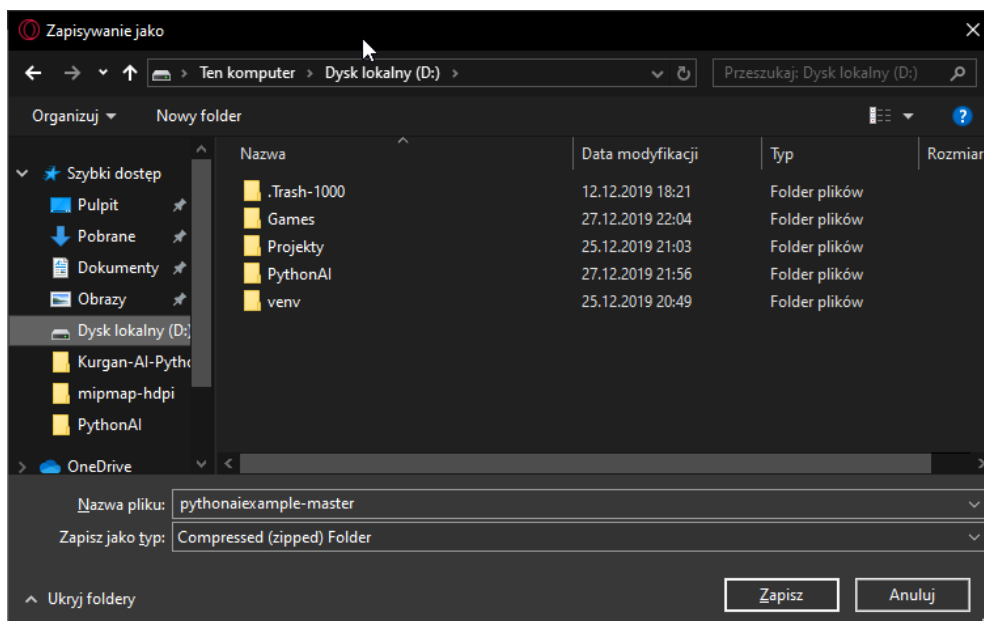
Windows

Aby pobrać paczkę klikamy na zielony przycisk **Clone or download**, a następnie klikamy **Download ZIP**.

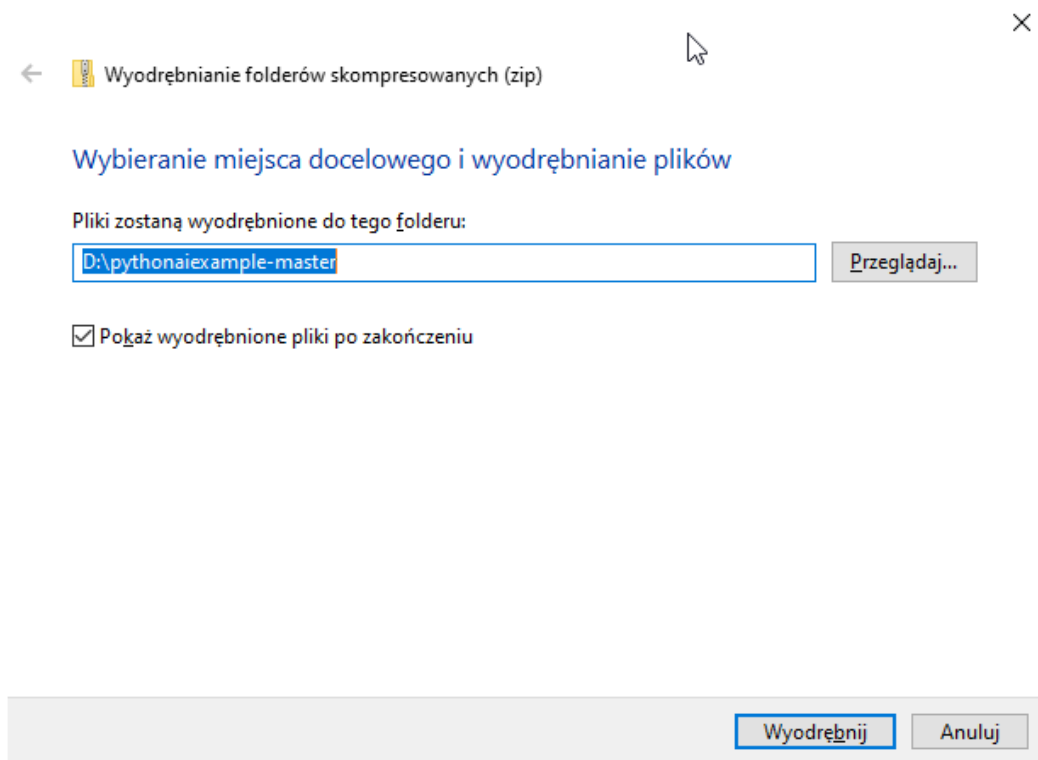


<https://github.com/milwilczynski/pythonaexample/archive/master.zip>

2. Plik najlepiej zapisać w miejscu, gdzie znajdują się nasze środowisko wirtualne. Na przykład, jeżeli Twoje środowisko znajdują się w folderze **/venv**, na partycji **D:** (**D:/venv**) to zapisz paczkę „przed” folderem z zainstalowanym środowiskiem, tj. **D:**



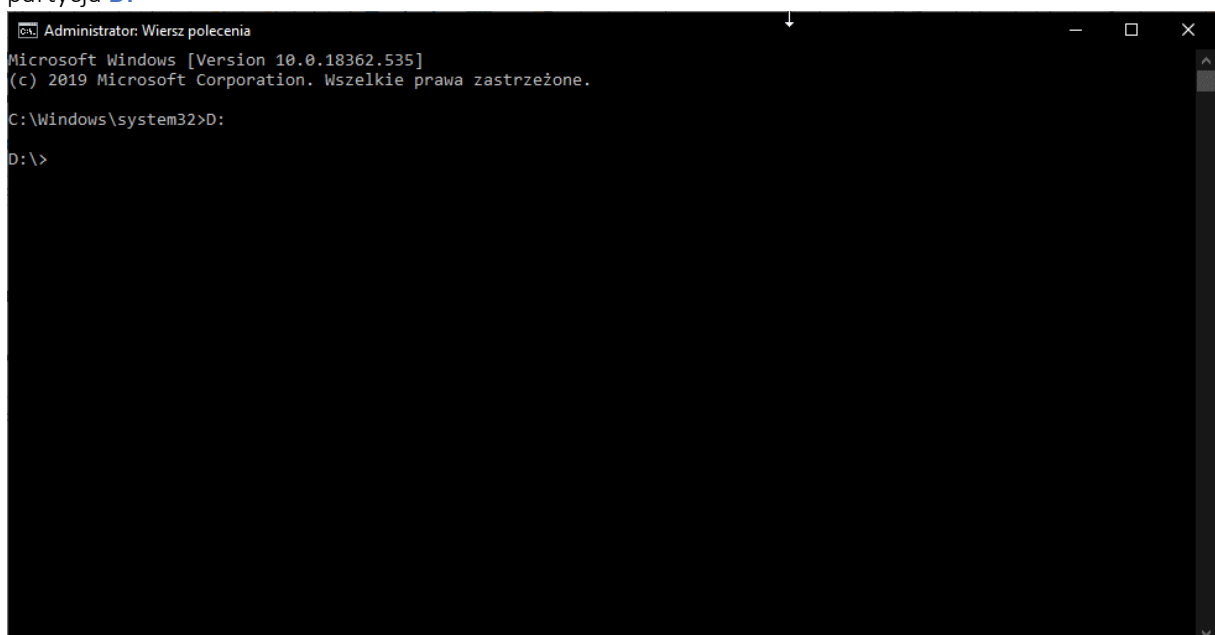
3. Następnie wypakowujemy plik pythonaexample-master.zip. Klikamy na plik **PPM (prawy przycisk myszy)**, a następnie wybieramy opcję **Wyodrębnij wszystkie...** Jeżeli posiadasz program **WinRar** bądź **7Zip** również możesz ich użyć.



4. Klikamy Wyodrębnij. Utworzy się katalog **pythonaexample-master**.

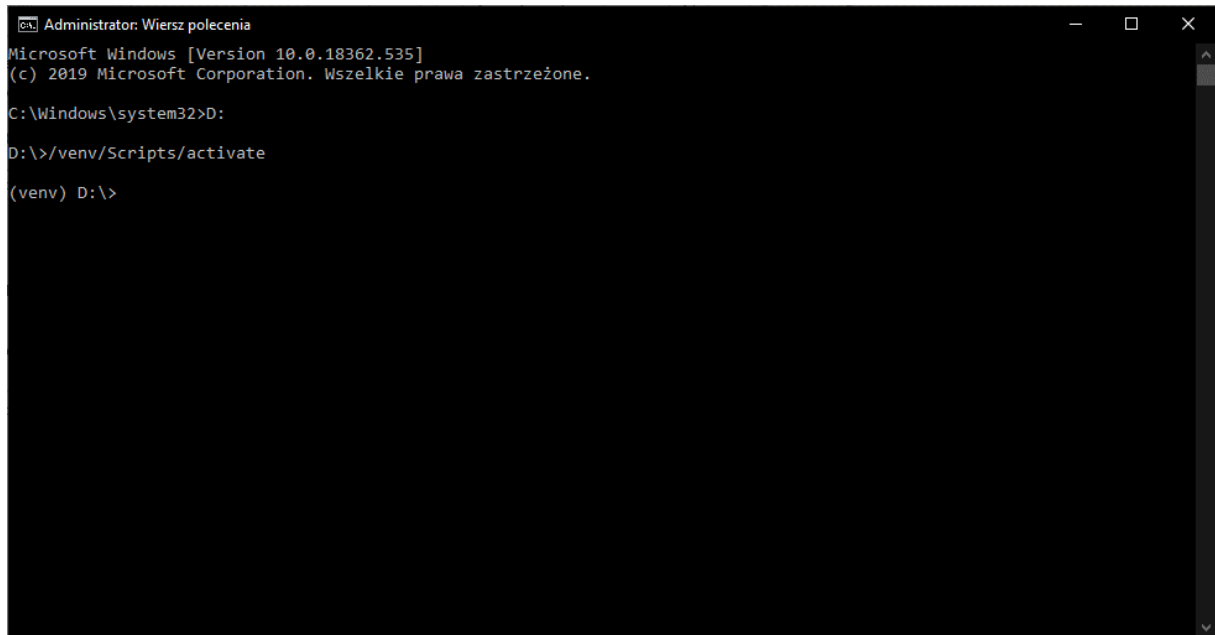
5. Odpalamy naszą środowisko wirtualne. Włączamy **Wiersz polecenia** najlepiej w **trybie administratora** – patrz [Strona 7](#).

6. Przechodzimy do folderu, gdzie znajdują się środowisko wirtualne, czyli **folder „venv”**. U mnie jest do partycja **D:**



7. Uruchamiamy środowisko wirtualne poleceniem

`/venv/Scripts/Activate`

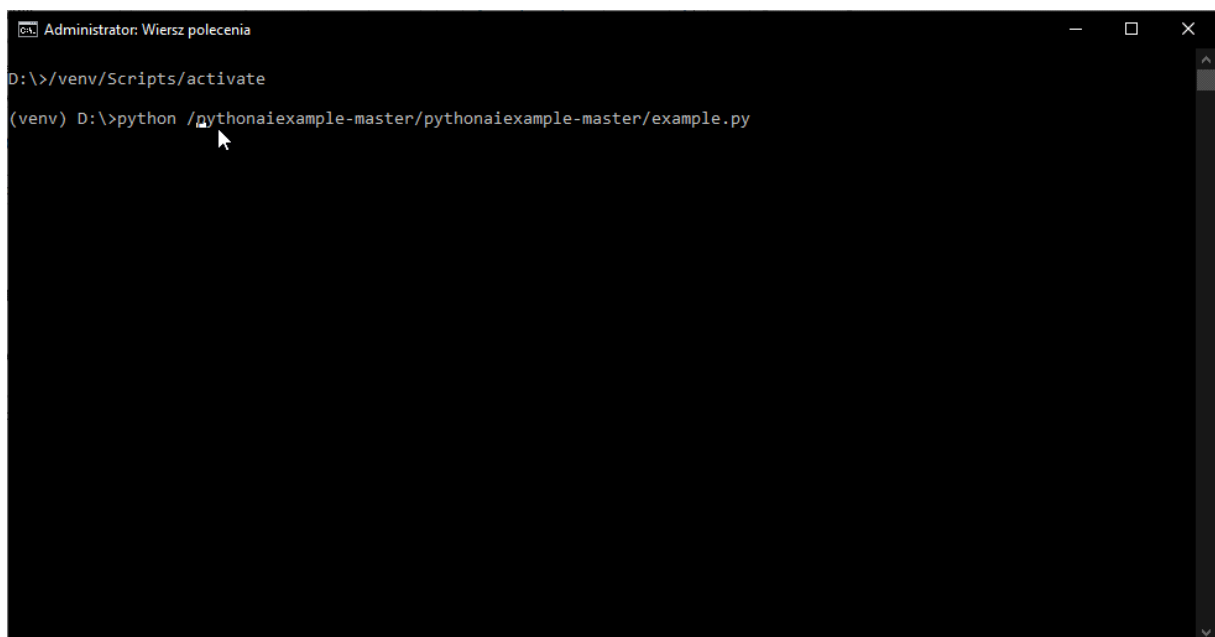


```
Administrator: Wiersz polecenia
Microsoft Windows [Version 10.0.18362.535]
(c) 2019 Microsoft Corporation. Wszelkie prawa zastrzeżone.

C:\Windows\system32>D:
D:\>/venv/Scripts/activate
(venv) D:\>
```

8. I przystępujemy do uruchomienia naszego programu komendą

`python /pythonaiexample-master/pythonaiexample-master/example.py`



```
Administrator: Wiersz polecenia

D:\>/venv/Scripts/activate
(venv) D:\>python /pythonaiexample-master/pythonaiexample-master/example.py
```

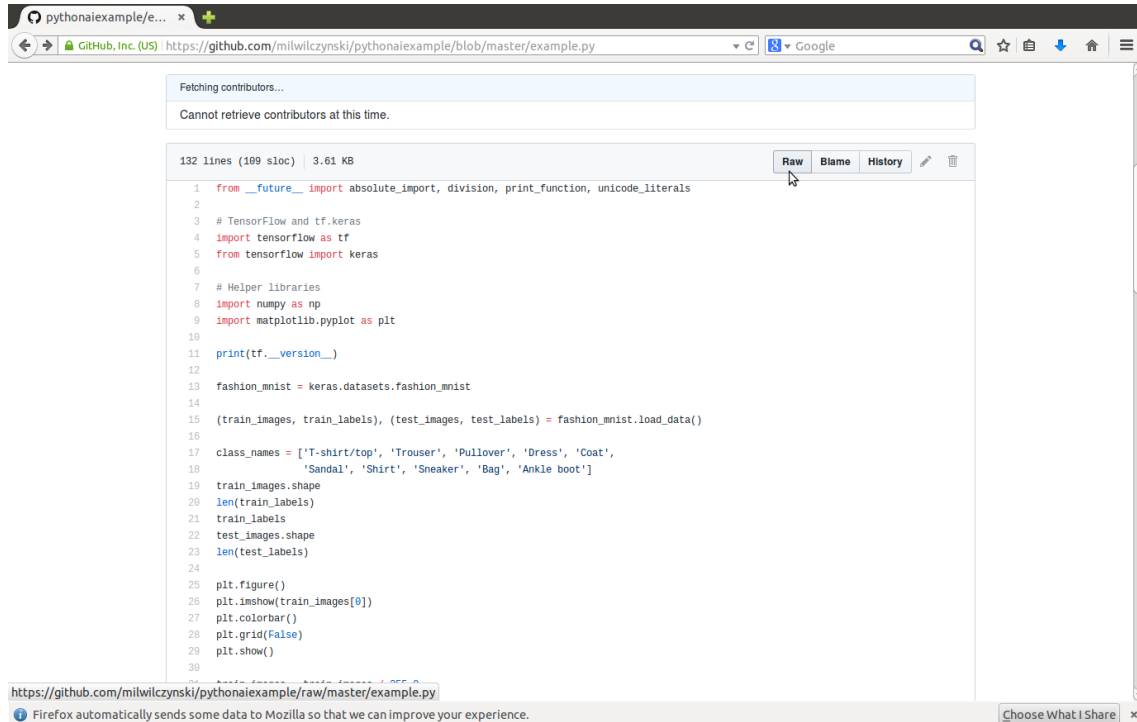
Podczas uruchomienia tego programu za pierwszym razem będą ściągać się obrazy potrzebne do nauki programu. Download pochodzi ze strony google.

Dalsza część instrukcji znajduje się w [punkcie 9](#)

Ubuntu: Ubuntu 12.04 bez aktualizacji może mieć problem. Mianowicie Firefox, może nie obsługiwać wszystkich skryptów zawartych na stronie GitHub. W tym celu proszę kliknąć na [example.py](#) na stronie

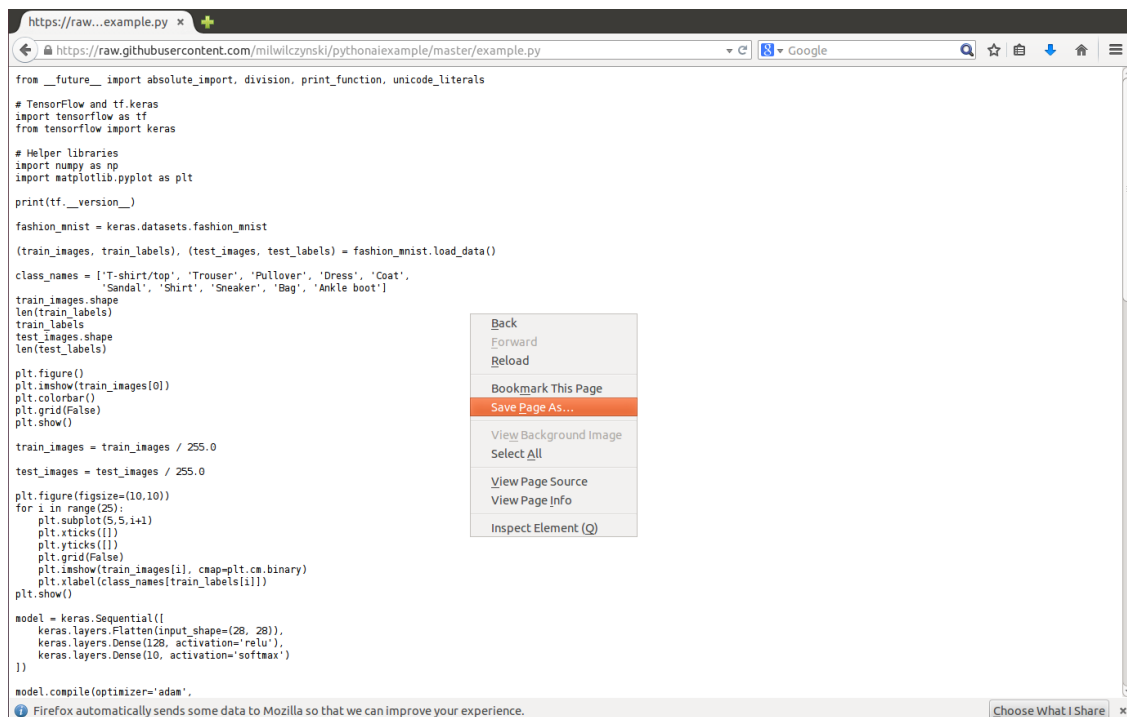
<https://github.com/milwilczynski/pythonaexample>

Przeniesie nas to do kodu źródłowego pliku.

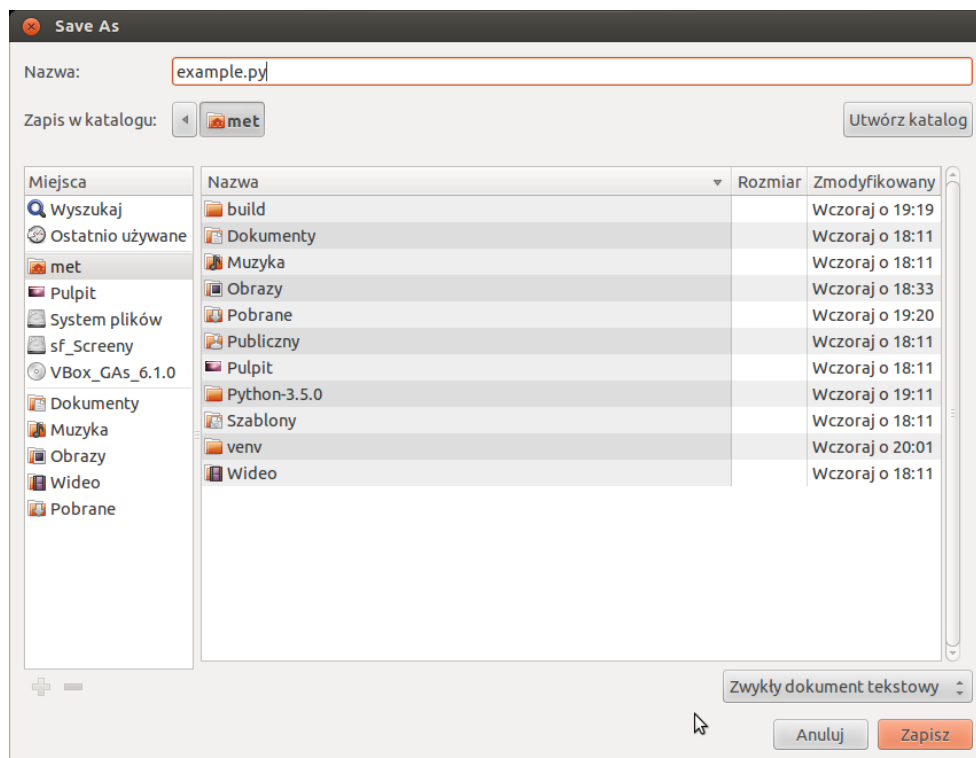


Po przeniesieniu klikamy **Raw**. Przeniesie nas to na stronę, gdzie znajduje się czysty kod programu.

Klikamy **PPM (prawym przyciskiem myszy)** na białe tło, a następnie klikamy **Zapisz jako**, u mnie jest to **Save Page As...**



Wybieramy folder, gdzie znajdują się nasz folder ze środowiskiem wirtualnym, u mnie jest to katalog `$HOME`, czyli folder z nazwą użytkownika po lewo (**patrz folder met**).



2. Włączamy **Terminal** skrótem klawiszowym

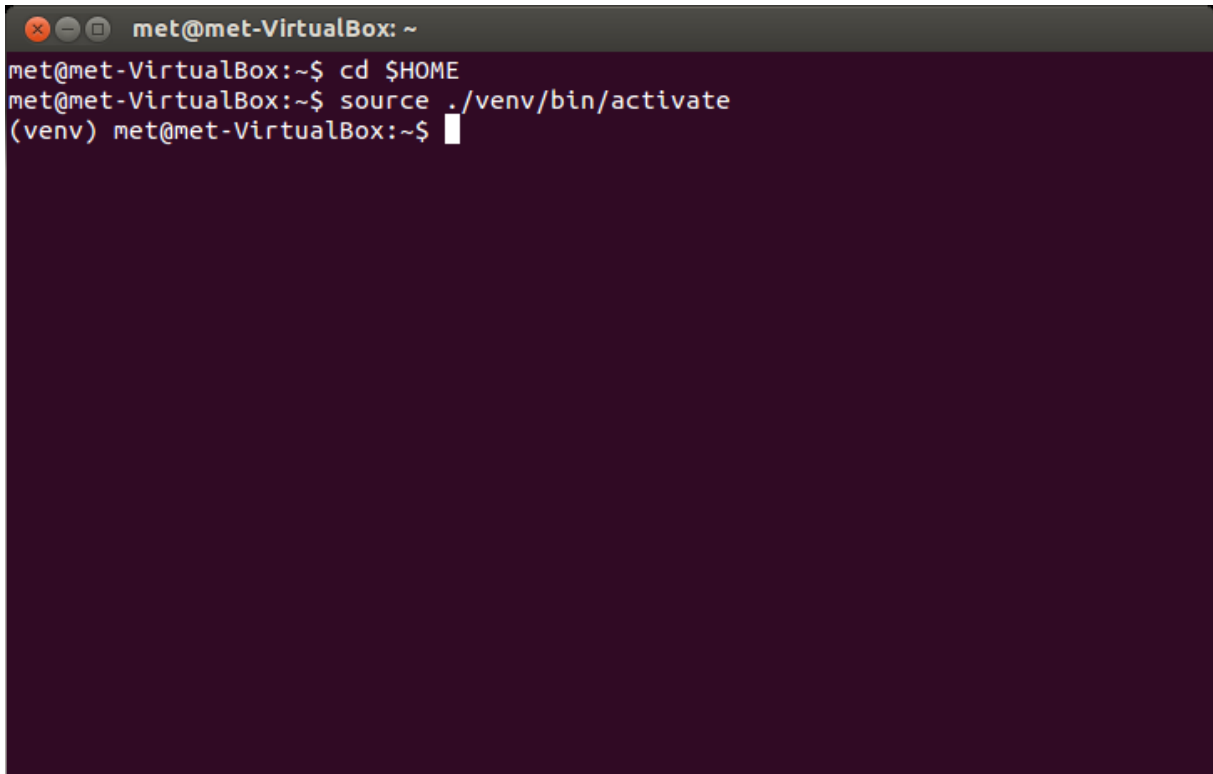
Control + Alt + T

albo otwieramy terminal z menu. I przechodzimy do katalogu, gdzie znajdują się folder „**venv**”. U mnie jest to katalog `$HOME`.

```
met@met-VirtualBox: ~  
met@met-VirtualBox:~$ cd $HOME  
met@met-VirtualBox:~$
```

3. Włączamy nasze środowisko wirtualne poleceniem

source ./venv/bin/activate

A terminal window titled 'met@met-VirtualBox: ~' with a dark purple background. The window shows the following commands and output: 'met@met-VirtualBox:~\$ cd \$HOME', 'met@met-VirtualBox:~\$ source ./venv/bin/activate', and '(venv) met@met-VirtualBox:~\$' with a cursor at the end of the last line.

```
met@met-VirtualBox:~$ cd $HOME
met@met-VirtualBox:~$ source ./venv/bin/activate
(venv) met@met-VirtualBox:~$
```

4. Włączamy program komendą

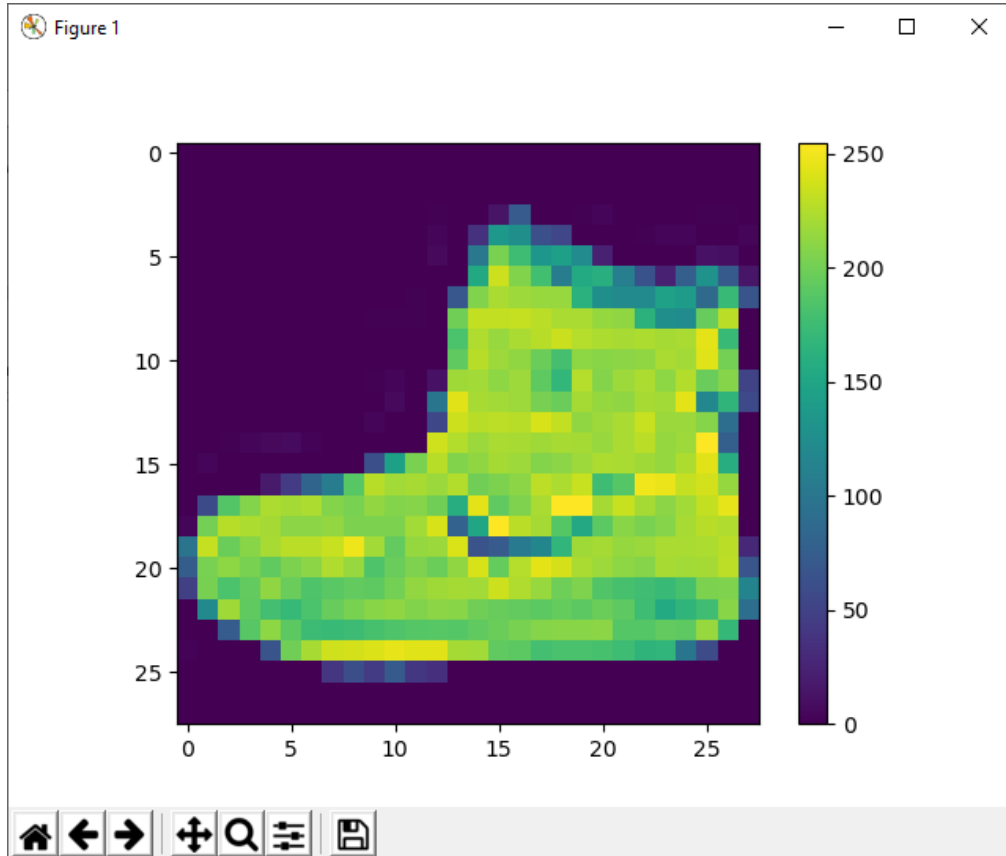
python ./Pobrane/example.py

Podczas uruchomienia tego programu za pierwszym razem będą ściągać się obrazy potrzebne do nauki programu.

Punkt 9

znajdujący się niżej jest wspólny dla Windowsa i Ubuntu

9. Po uruchomieniu (i ewentualnym czekaniu) pierwsze okienko jakie powinno się nam pokazać to:



10. Aby pokazać następne okienko wyłączamy obecne klikając na krzyżyk.

Jeżeli jesteśmy ciekawi to oglądamy wyniki uczenia!

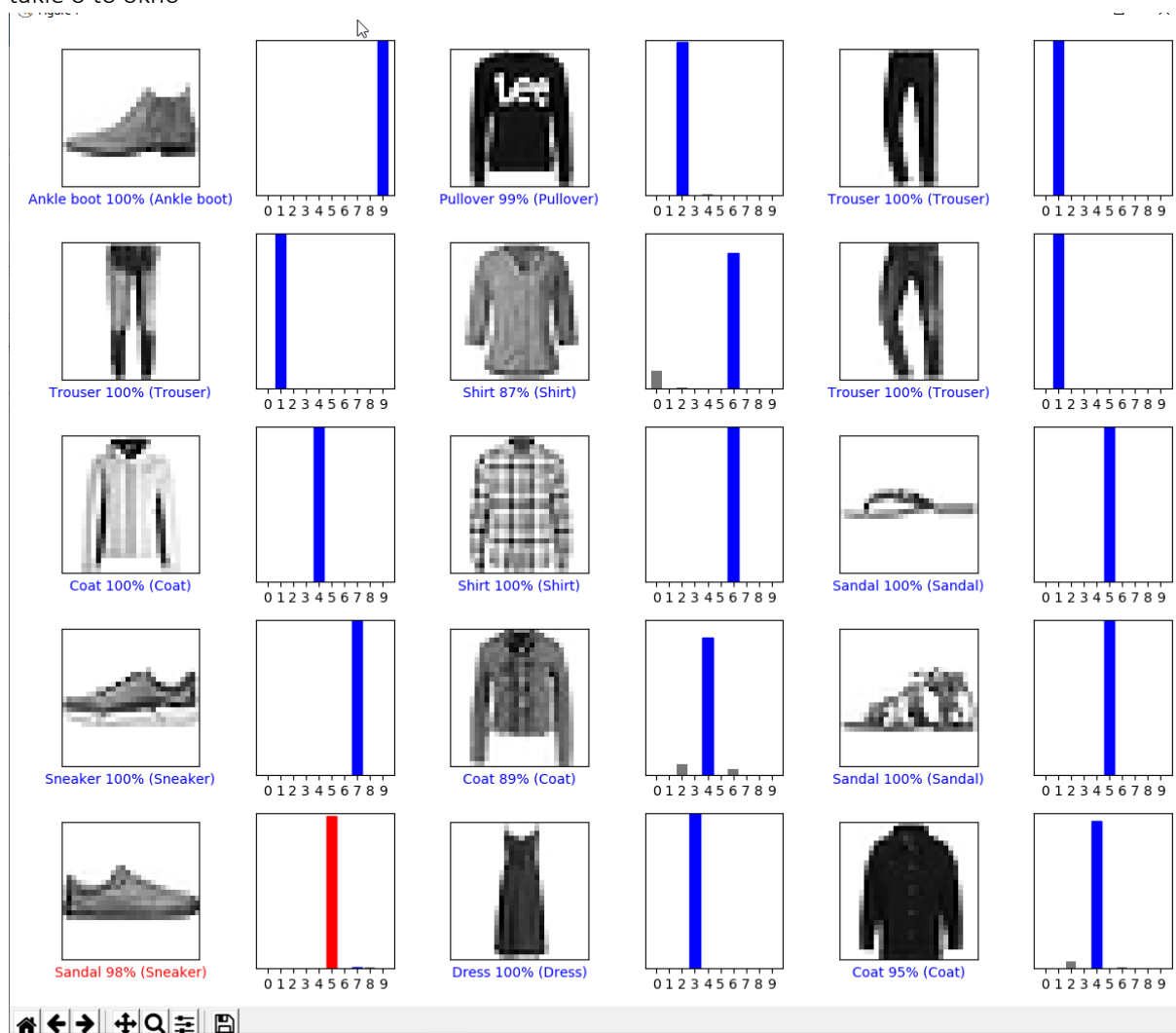
W pewnym momencie odpalania programu może nastąpić symulacja instrukcji. U mnie jest to instrukcja AVX2.

```
Administrator: Wiersz polecenia - python /pythonaiexample-master/pythonaiexample-master/example.py
D:\>venv/Scripts/activate

(venv) D:\>python /pythonaiexample-master/pythonaiexample-master/example.py
2.0.0
2019-12-28 12:17:41.340286: I tensorflow/core/platform/cpu_feature_guard.cc:142] Your CPU supports instructions that this TensorFlow binary was not compiled to use: AVX2
Train on 60000 samples
Epoch 1/10
60000/60000 [=====] - 4s 59us/sample - loss: 0.4969 - accuracy: 0.8270
Epoch 2/10
60000/60000 [=====] - 3s 46us/sample - loss: 0.3780 - accuracy: 0.8643
Epoch 3/10
60000/60000 [=====] - 3s 45us/sample - loss: 0.3395 - accuracy: 0.8781
Epoch 4/10
13056/60000 [=====>.....] - ETA: 2s - loss: 0.3227 - accuracy: 0.8829
```

Nie ma się czym przejmować. Są to instrukcje procesora, które są zawarte w listach rozkazów SSE. Niektóre procesory mogą najzwyczajniej w świecie ich nie posiadać, dlatego następuje symulacja.

11. Pojawi się jeszcze kilka okienek, które możemy bez problemu zamykać. Ostatnim okienkiem jest takie o to okno



Gratuluję! Program testowy przebiegł pomyślnie!