Wizualizacja danych - wykład 1

dr Piotr Jastrzębski

Sprawy ogranizacyjne

Sprawy ogranizacyjne

- Sylabus jest dostępny w systemie USOS.
- Regulamin zajęć dostępny jest na stronie prowadzącego zajęcia http://wmii.uwm.edu.pl/~piojas/.
- Forma zaliczenia: egzamin.
- Wykład - 15 godzin, zajęcia w ustalonych terminach.
- Terminarz, prezentacje i inne materiały związane z wykładem będą udostępniane w repozytorium Na Githubie https://github.com/pjastr/WizualizacjaDanychStac .

Wymagania wstępne

- Znajomość podstawowych konstrukcji programistycznych (ze wstępu do programowania).
- Matematyka z zakresu szkoły średniej/z przedmiotu repozytorium matematyki elementarnej.

Ewentualne braki należy opanować w samodzielnym zakresie.

W razie problemów zapraszam na konsultacje.

Wstęp to języka Python

Język Python

- Poprawna wymowa: pajton.
- Język Python stworzył we wczesnych latach 90. Guido van Rossum – jako następcę języka ABC.
- Nazwa języka pochodzi od serialu komediowego emitowanego w latach siedemdziesiątych przez BBC – „Monty Python's Flying Circus" (Latający cyrk Monty Pythona). Projektant, będąc fanem serialu i poszukując nazwy krótkiej, unikalnej i nieco tajemniczej, uznał tę za świetną.

Przełomowy rok - 2008

- Utworzenie drugiej gałęzi rozwoju 3.x. Początkowe obie gałęzie były rozwijane niezależnie, lecz na dziś zostało ogłoszone zakończenia wsparcia Pythona 2.x na rok 2020.
- Wykład będzie oparty o wersję 3.7.2 32-bitową (choć bardzo rzadko będzie korzystać z ostatnich nowości).

Podstawowe różnice między 2.x a 3.x

- funkcja `print`

Python 2:

```
print 'Hello, World!'
print('Hello, World!')
print "text", ; print 'print more text on the same line'
```

Python 3

```
print('Hello, World!')
print("some text,", end="")
print(' print more text on the same line')
```

Dzielenie zmiennych typu `int`

Python 2:

```
print '3 / 2 =', 3 / 2
print '3 // 2 =', 3 // 2
print '3 / 2.0 =', 3 / 2.0
print '3 // 2.0 =', 3 // 2.0
```

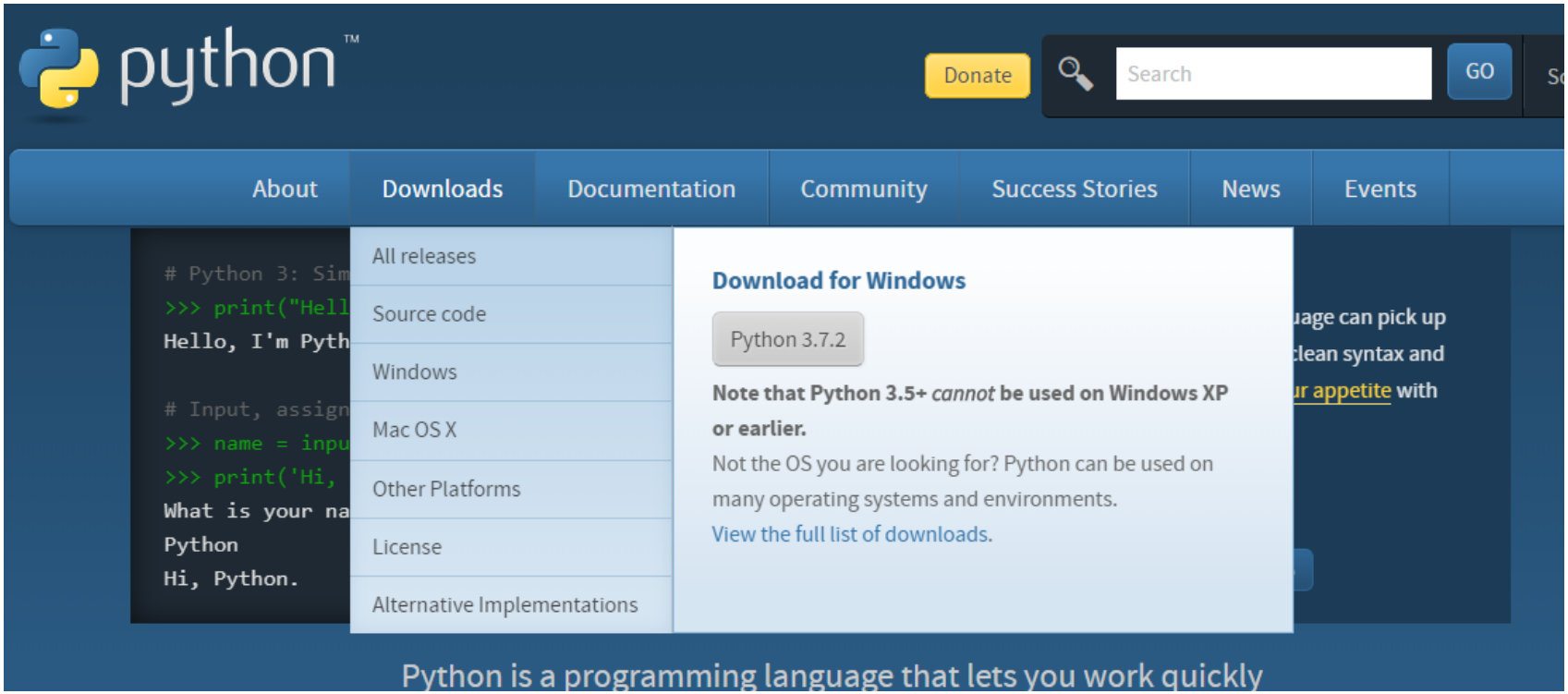
Python 3

```
print('3 / 2 =', 3 / 2)
print('3 // 2 =', 3 // 2)
print('3 / 2.0 =', 3 / 2.0)
print('3 // 2.0 =', 3 // 2.0)
```

Warto doczytać np. tutaj.

Instalacja - Windows

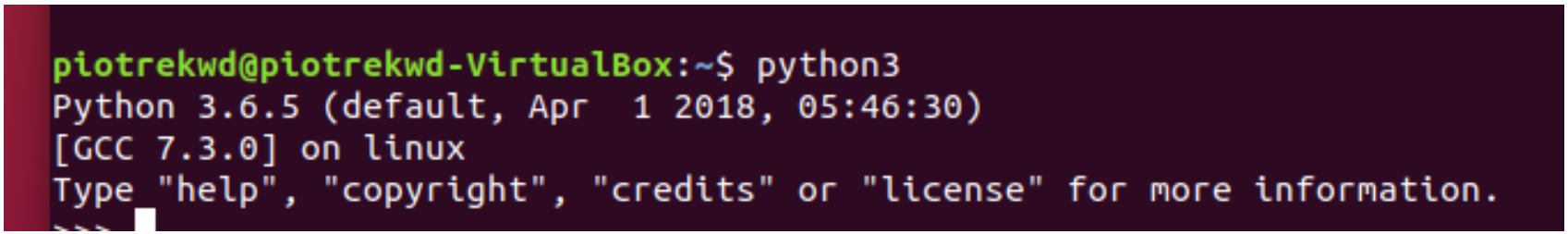
- https://python.org/



Rysunek 1: Strona www

Linux

Sprawdzenie wersji na Ubuntu 18.04:



Ręczna instalacja:

```
sudo apt install python3
```

Wybór IDE do Pythona

- IDLE (domyślny)
- PyCharm https://www.jetbrains.com/pycharm/ (na ćw. i wykład)
- SPyder IDE https://www.spyder-ide.org/
- Visual Studio https://visualstudio.microsoft.com/pl/vs/features/python/
- Visual Studio Code + odpowiednie rozszerzenia https://code.visualstudio.com/
- Atom + ide-python https://atom.io/packages/ide-python
- i wiele innych...

PyCharm Community Edition

- IDE używany na wykładzie i sugerowany do użycia w trakcie ćwiczeń, na egzaminie można użyć dowolny IDE zainstalowany w pracowni (choć zalecane jest użycie PyCharm),
- edytor z tzw. "inteligentnymi podpowiedziami"
- graficzny debugger
- inspekcja kodu, refaktoryzacja, wsparcie dla systemów kontroli wersji

Dla m.in. studentów dostępna jest za darmo wersja Professional. Informacje o niej dostępne są tutaj: https://www.jetbrains.com/student/.

Sugerowane jest użycie uczelnianego maila w domenie @student.uwm.edu.pl (więcej informacji o niej jest tutaj: http://www.uwm.edu.pl/studenci/uslugi-informatyczne) lub karty ISIC.

Styl PEP8

- wymowa: pi-p-pi-ejt
- standaryzacja kodu używana m.in. przy rozwijaniu nowych funkcjonalności
- używanie daje lepszą organizację i czytelność kod
- pełna wersja https://www.python.org/dev/peps/pep-0008/

Znaki odstępu:

- we wcięciach stosujemy spacje (a nie tabulatory)
- każdy poziom wcięcia powinien składać się z 4 spacji
- wiersz powinien składać się z maksymalnie 79 znaków

Dobrze:

```
foo = long_function_name(var_one, var_two,
                         var_three, var_four)

def long_function_name(
        var_one, var_two, var_three,
        var_four):
    print(var_one)

foo = long_function_name(
    var_one, var_two,
    var_three, var_four)
```

Źle:

```
foo = long_function_name(var_one, var_two,
    var_three, var_four)

def long_function_name(
    var_one, var_two, var_three,
    var_four):
    print(var_one)
```

Instrukcje warunkowe:

```
if (this_is_one_thing and
    that_is_another_thing):
    do_something()

if (this_is_one_thing and
    that_is_another_thing):
    # dodatkowy komentarz
    do_something()

if (this_is_one_thing
        and that_is_another_thing):
    do_something()
```

Listy:

```
my_list = [
    1, 2, 3,
    4, 5, 6,
    ]
result = some_function_that_takes_arguments(
    'a', 'b', 'c',
    'd', 'e', 'f',
    )
```

Listy - druga wersja:

```
my_list = [
    1, 2, 3,
    4, 5, 6,
]
result = some_function_that_takes_arguments(
    'a', 'b', 'c',
    'd', 'e', 'f',
)
```

Operatory arytmetyczne a przenoszenie:

Źle:

```
income = (gross_wages +
          taxable_interest +
          (dividends - qualified_dividends) -
          ira_deduction -
          student_loan_interest)
```

Dobrze:

```
income = (gross_wages
          + taxable_interest
          + (dividends - qualified_dividends)
          - ira_deduction
          - student_loan_interest)
```

Puste linie:

- dwie linie między funkcjami najwyższego poziomu i między klasami.
- pojedyncza linia między funkcjami w klasie

Kodowanie: * dla Pythona 3 sugerowane i domyślne to UTF-8.

Importowanie bibliotek

Dobrze:

```
import os
import sys
```

Źle:

```
import sys, os
```

Ale dobrze też:

```
from subprocess import Popen, PIPE
```

Bibliografia

- https://pl.wikipedia.org/wiki/Python, dostęp online 12.02.2019.
- https://bulldogjob.pl/news/264-java-php-ruby-jak-wlasciwie-wymawiac-nazwy-technologii. dostęp online 12.02.2019.
- https://sebastianraschka.com/Articles/2014_python_2_3_key_diff.html, dostęp online 14.02.2019.
- K. Ropiak, Wprowadzenie do języka Python, http://wmii.uwm.edu.pl/~kropiak/wd/Wprowadzenie%20do%20j%C4%99zyka%20Python.pdf, dostęp online 14.02.2019.
- B. Slatkin, Efektywny Python. 59 sposobów na lepszy kod, Helion 2015.
- https://www.python.org/dev/peps/pep-0008/, dostęp online 14.02.2019.