

Wizualizacja danych

- wykład 1

dr Piotr Jastrzębski

Sprawy organizacyjne

Sprawy organizacyjne

- ▶ Syllabus jest dostępny w systemie USOS.
- ▶ Regulamin zajęć dostępny jest na stronie prowadzącego zajęcia <http://wmii.uwm.edu.pl/~piojas/>.
- ▶ Forma zaliczenia: egzamin.
- ▶ Wykład - 10 godzin.
- ▶ <https://github.com/pjastr/WizualizacjaDanychNStac2022>

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.

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
```

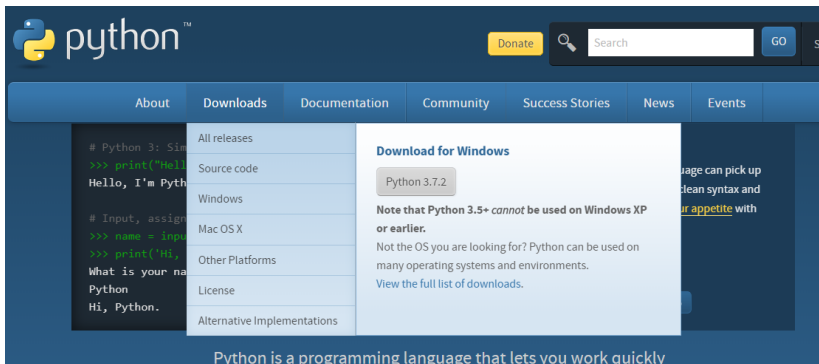
Python 3

```
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/>



The image shows a screenshot of the Python.org website. At the top left is the Python logo and the word "python" with a trademark symbol. To the right is a yellow "Donate" button and a search bar with a magnifying glass icon, a "GO" button, and a "Se" label. Below the header is a navigation menu with tabs for "About", "Downloads", "Documentation", "Community", "Success Stories", "News", and "Events". The "Downloads" tab is active, and a dropdown menu is open, listing options: "All releases", "Source code", "Windows", "Mac OS X", "Other Platforms", "License", and "Alternative Implementations". The "Windows" option is selected, and a white box titled "Download for Windows" is displayed. Inside this box, there is a button for "Python 3.7.2". Below the button, the text reads: "Note that Python 3.5+ cannot be used on Windows XP or earlier. Not the OS you are looking for? Python can be used on many operating systems and environments. View the full list of downloads." At the bottom of the page, a blue banner contains the text: "Python is a programming language that lets you work quickly".

Rysunek 1: Strona www

Linux

Sprawdzenie wersji na Ubuntu 18.04:

```
piotrekwd@piotrekwd-VirtualBox:~$ python3
Python 3.6.5 (default, Apr 1 2018, 05:46:30)
[GCC 7.3.0] on linux
Type "help", "copyright", "credits" or "license" for more information.
```

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...

Styl PEP8

- ▶ wymowa: pi-i-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

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.

Stringi:

- ▶ można używać pojedynczych apostrofów jak i podwójnych cudzysłówów
- ▶ ważne, aby stosować wybraną notację konsekwentnie
- ▶ jedyny wyjątek to gdy wewnątrz stringu chcemy użyć cudzysłów np.

```
print('Oglądam film "Player One"')
```

Cechy języka Python

- ▶ Python wspiera różne paradygmaty programowania: obiektowy, imperatywny oraz funkcyjny.
- ▶ Posiada w pełni dynamiczny system typów i automatyczne zarządzanie pamięcią (garbage collector).
- ▶ Często używany jako język skryptowy. Interpretery Pythona są dostępne na wiele systemów operacyjnych. Różne implementacje Pythona: CPython (język C), IronPython (platforma .NET), Jython (Java), PyPy (Python).
- ▶ Prosta i czytelna składnia ułatwiająca utrzymywanie, używanie i rozumienie kodu.

Zen

```
import this
```

The Zen of Python, by Tim Peters

Beautiful is better than ugly.

Explicit is better than implicit.

Simple is better than complex.

Complex is better than complicated.

Flat is better than nested.

...

PL - [https:](https://pl.python.org/)

[//pl.python.org/forum/index.php?topic=392.msg1844#msg1844](https://pl.python.org/forum/index.php?topic=392.msg1844#msg1844)

Zmienna

- ▶ najprościej: przechowuje pewną wartość:

```
x = 5
```

```
y = "John"
```

Wbudowane typy danych

- ▶ numeryczne (liczbowe): int, float, complex
- ▶ tekstowe: str
- ▶ sekwencji: list, tuple
- ▶ odwzorowania (mapping type): dict
- ▶ zestawów (set types): set, frozenset
- ▶ logiczne: bool
- ▶ binarne: bytes, bytearray

Int - typ całkowity

- ▶ bez kropki dziesiętnej
- ▶ może być dowolnie długi (ograniczenie ilość pamięci)

```
print(123123123123123123123123123123123123123123 + 1)
```

Jaki system liczbowy?

Domyślnie dziesiętny. Więcej za kilka zajęć przy algorytmach liczbowych.

```
print(101)
print(101)
print(0x101) # zero-x
print(0o101) # zero-litera o
print(0b101) # zero-b
print(0X101) # zero-x
print(0O101) # zero-litera o
print(0B101) # zero-b
```

Sprawdzenie typu

```
print(type(234))
```

<class 'int'> - wszystko jest obiektem

Float - typ zmiennoprzecinkowy

```
print(4.2)
print(4.)
print(.5)
print(.3e4)
print(3e4)
print(2e-2)
print(1.79e308)
print(1.8e308)
print(5e-324)
print(1e-325)
```

Complex - typ zespolony (dot. liczb zespolonych)

```
print(2+3j)  
print(2+5*1j)
```

Uwaga: wyrażenie

```
print(4+5*j)
```

skutkuje błędem w większości wypadków.

str - string, napisy, łańcuchy znaków

Obecnie odchodzi się od określenia “tablica znaków”

```
print("ABC")  
print('abc')
```

bool - typ logiczny

```
print(True)  
print(False)
```

Operator przypisania

= przypisuje prawą stronę do lewej (!), często mylony z operatorem logicznym równa się ==

```
x = 5
```

```
x = "Piotr"
```

```
a = 4.5
```

```
A = 56
```

```
x, y, z = "Orange", "Banana", "Cherry"
```

```
x = y = z = "Orange"
```

input - operacja wejścia

```
num = input ("Wprowadź liczbę :")  
print(num)  
name1 = input("Wprowadź imię : ")  
print(name1)
```

- ▶ zawsze przyjmuje napis, w razie potrzeby trzeba zrzutować

```
x = str(num)  
y = int(num)  
z = float(num)
```

print - instrukcja wyjścia

```
print(4.2)
# print('Mój wiek to' + 36)
print('Mój wiek to', 36)
a = 36
print('Mój wiek to', a)
print('hello', 'world')
print('hello', 'world', sep='')
print('hello', 'world', sep='\n')
```

```
print(*objects, sep=' ', end='\n', file=sys.stdout,  
      flush=False)
```

- ▶ `objects` - to co ma być wyświetlone
- ▶ `sep` - separator, domyślnie znak spacji
- ▶ `end` - co co ma być wyświetlone na końcu, domyślnie znak końca linii
- ▶ `file` - określa gdzie mają być `objects` wyświetlone, domyślnie `sys.stdout` (domyślny ekran)
- ▶ `flush` - określa czy “wyjście” ma być buforowane przed przekazaniem do `file`, domyślne `False`

```
print(1, 2, 3, 4)
```

```
## 1 2 3 4
```

```
print(1, 2, 3, 4, sep='*')
```

```
## 1*2*3*4
```

```
print(1, 2, 3, 4 ,sep='#', end='&')
```

```
## 1#2#3#4&
```

```
print('x', 'y', 'z', sep='', end='')  
print('a', 'b', 'c', sep='', end='')
```

```
## xyzabc
```

```
print('a', 'b', '\n', 'c')
```

```
## a b
```

```
## c
```


\t - przesunięcie do następnego "tab"=8 spacji

```
print('sdf', 3456, -2, sep='\t')
```

```
## sdf    3456    -2
```

Formatowanie napisów będzie później.

Bibliografia

- ▶ <https://pl.wikipedia.org/wiki/Python>, dostęp online 12.02.2019.
- ▶ <https://bulldogjob.pl/news/264-java-php-ruby-jak-wlasciwie-wymawiac-nazwy-technologiei>. 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.

Bibliografia - cd2

- ▶ <https://www.flynerd.pl/2017/05/python-4-typy-i-zmienne.html>, dostęp online 14.02.2019.
- ▶ <http://pytolearn.csd.auth.gr/p0-py/01/print.html>, dostęp online 15.02.2019.
- ▶ https://www.tutorialspoint.com/python3/python_lists.htm, dostęp online 17.02.2019.

Bibliografia - cd3

- ▶ <https://realpython.com/python-data-types/>, dostęp online 5.01.2022
- ▶ https://www.w3schools.com/python/python_variables.asp, dostęp online 5.01.2022
- ▶ https://www.w3schools.com/python/python_variables_multiple.asp, dostęp online 5.01.2022
- ▶ <https://realpython.com/python-print/>, dostęp online 5.01.2022
- ▶ <https://www.programiz.com/python-programming/operators>, dostęp online 5.01.2022
- ▶ <https://realpython.com/python-conditional-statements/>, dostęp online 5.01.2022
- ▶ <https://realpython.com/python-for-loop/>, dostęp online 5.01.2022
- ▶ <https://realpython.com/python-while-loop/>, dostęp online 5.01.2022