

Podstawy programowania Python

Wprowadzenie

Jolanta Bachan

Informacje do kontaktu

- Email: jbachan@amu.edu.pl
 - jolabachan@gmail.com
- Strona internetowa: <http://bachan.speechlabs.pl/>
- Dyżury: pokój 204B lub 218B (BOW)
 - wtorek 9:35-10:35
 - czwartek 11:20-12:20

Syllabus (1)

- analiza istniejących programów w celu zilustrowania zagadnień programistycznych
- wykorzystanie zdobytej wiedzy do modyfikacji istniejących programów
- wykorzystanie zdobytej wiedzy do tworzenia własnych programów

Syllabus (2)

- dane tekstowe, liczbowe, listowe, zbiory
- operatory, pętle, instrukcje warunkowe
- definiowanie funkcji
- działanie na plikach input/output
- wyrażenia regularne
- kodowanie znaków – UTF8, Latin-2, cp1250

Syllabus (3)

- analiza i przetwarzanie korpusów językowych za pomocą programów skryptowych
 - tokenizacja
 - lista wyrazowa
 - normalizacja tekstu
 - kolokacje i bigramy
 - konkordans
 - statystyki

Literatura

- Bird, S., Klein, E. Loper, E. 2009. Natural Language Processing with Python – Analyzing Text with the Natural Language Toolkit. O'Reilly Media, <<http://www.nltk.org/book>>
- Moura, R. 2020. The Python Guide for Beginners. Start coding in Python 3. <<https://www.freecodecamp.org/news/the-python-guide-for-beginners/>>
- van Rossum, Guido. 2004. Przewodnik po języku Python. Wydanie 2.3. PythonLabs. <<https://pl.python.org/docs/tut/tut.html>>
- Python – Dokumentacja. <http://www.python.org/doc/>
- Church, K.W. Unix™ for Poets.
- Jurafsky, Daniel, and James H. Martin. 2009. Speech and Language Processing: An Introduction to Natural Language Processing, Speech Recognition, and Computational Linguistics. 2nd edition. Prentice-Hall.
- Ruslan, M. (Ed.) 2002. The Oxford Handbook of Computational Linguistics. Oxford University Press
- Friedl, J. 2001. Wyrażenia regularne. Helion. O'Reilly

Zaliczenie

- Aktywność na zajęciach
- Obecność na zajęciach
 - dopuszczalne 2 nieobecności w semestrze
 - po przekroczeniu limitu 2 nieobecności, należy je usprawiedliwić (np. zwolnieniami lekarskimi) i zaliczyć wszystkie nieobecności na konsultacjach poprzez rozwiązywanie zadań lub odpowiedzi na pytania
 - 3 **spóźnienia** liczone są jako 1 nieobecność
- Zgromadzenie 5 punktów w semestrze z aktywności na zajęciach i z zadań domowych jest warunkiem dopuszczenia do zaliczenia
- Test: Napisanie prostych programów z wykorzystaniem funkcji poznanych na zajęciach (*należy zdobyć minimum 60% punktów*)
- Rejestracja w USOSie – *To już chyba macie za sobą, tak?*

POWODZENIA!

Termin zaliczenia

- Test zaliczeniowy: 28 stycznia 2025
- Test poprawkowy: 25 lutego 2025

Zainstaluj i przetestuj Pythona

- Instrukcje znajdziesz na naszej stronie
http://bachan.speechlabs.pl/en/teaching/win2024_2025/programming_LK

Zainstaluj i przetestuj Pythona

```
print ('Hello world!')
```

- operatory numeryczne: + - * / // % **
- operatory porównania: == != > < >= <=

Zmienne i typy

- język typowany dynamicznie, tzn. nie musisz deklarować typu danych wcześniej
 - >> tekst = 'Zmienne i typy'
 - >> liczba = 15
- każda zmienna jest obiektem i jest powiązana z metodami

Zmienne i typy

- łańcuchy znaków – **str** – 'tekst' "Don't worry"
- liczby całkowite – **int** – 1 10 300
- liczby rzeczywiste – **float** – 1.5 3.479
- tablica – **list** – ['to', 'jest', 'lista'] [1, 10, 300]
 - >> lista = ['to', 'jest', 'lista']
 - >> lista[0]
- zbiór **set**([1, 10, 300, 1, 10]) {1, 10, 300, 1, 10}
 - >> zbior = x = set([1, 10, 300, 1, 10])
 - >> if 1 in x:
 - print ('1 jest w zbiorze!')

Zmienne i typy

- logiczny/boolowski – **bool** – True False

```
>> x = True
```

```
>> if x == True:
```

```
    print ('I am right')
```

```
else:
```

```
    print ('I am wrong')
```

- słownik `a = dict(one=1, two=2, three=3)`

```
>> mydictionary = {'car' : 'auto', 'cat' : 'kot', 'house' : 'dom'}
```

```
>> mydictionary['car']
```

```
'auto'
```

Tekst

```
>>> x = "Don't worry"
```

```
>>> print(x)
```

```
Don't worry
```

```
>>> x = 'Don\'t worry'
```

```
>>> print(x)
```

```
Don't worry
```

```
>>> x = """Don't  
worry"""
```

```
>>> print(x)
```

```
Don't
```

```
worry
```

Konkatenacja:

```
>>> x = 'Hello'
```

```
>>> y = 'world'
```

```
>>> print(x + ' ' + y)
```

```
Hello world
```

Konwersja typów

```
>>> tekst = 'hello world '
```

```
>>> rzeczywista = 2.0
```

```
konkatenacja = tekst + rzeczywista
```

```
Traceback (most recent call last):
```

```
File "<pysHELL#130>", line 1, in <module>
```

```
    konkatenacja = tekst + rzeczywista
```

```
TypeError: cannot concatenate 'str' and 'float' objects
```

```
>>> konkatenacja = tekst + str(rzeczywista)
```

```
>>> print (konkatenacja)
```

```
hello world 2.0
```

Sprawdź typ danych

```
>>> type(tekst)
```

```
<type 'str'>
```

```
>>> type(rzeczywista)
```

```
<type 'float'>
```


Drukowanie

```
>>> print ("To jest mój napis: %s" % tekst)
```

```
To jest mój napis: hello world
```

```
>>> calkowita = 35
```

```
>>> print ("To jest moja liczba całkowita: %d" % calkowita)
```

```
>>> print ("To jest moja liczba rzeczywista: %.3f" % rzeczywista)
```

```
To jest moja liczba rzeczywista: 2.000
```

```
>>> print ("To jest mój napis: %s i moja liczba rzeczywista %.3f"  
% (tekst, rzeczywista))
```

```
To jest mój napis: hello world i moja liczba rzeczywista 2.000
```

Drukowanie

```
>>> print('komputer', 'klawiatura', 'myszka', sep=' -> ')
```

```
komputer -> klawiatura -> myszka
```

```
>>> print('komputer', 'klawiatura', 'myszka', sep=' -> ', end='.')
```

```
komputer -> klawiatura -> myszka.
```

```
>>> print('{} -> {} -> {}'.format('komputer','klawiatura','myszka'))
```

```
komputer -> klawiatura -> myszka
```

```
>>> print('{}'.format(3))
```

```
3
```

```
>>> print('{:.3f}'.format(3.438736374))
```

```
3.439
```

```
>>> print('from {1} to {0}'.format('A', 'B'))
```

```
from B to A
```

Pętlą for

```
>>> for x in 'hello':  
    print (x)
```

h

e

l

l

o

```
>>> for x in range(5):  
    print (x)
```

0

1

2

3

4

Peçta for

```
>>> for x in range(3,6):  
    print (x, end=' ')
```

3 4 5

Pełna for + enumerate

```
>>> word = 'Python'
```

```
>>> for i, letter in enumerate(word):  
    print (i, letter)
```

```
0 P
```

```
1 y
```

```
2 t
```

```
3 h
```

```
4 o
```

```
5 n
```

pętla for i sumowanie liczb

```
>>> suma = 0          0
>>> for x in range(20): 1
    suma = suma + x    3
                        6
                        10
>>> print (suma)      15
190                    ...
                        190
```

Tablica

```
>> tablica = []  
>> tablica.append(1)  
>> tablica.append(2)  
>> tablica.append(3)  
>> print (tablica[0])  
>> print (tablica[1])  
>> print (tablica[2])
```

- pętla for
for x **in** tablica:
 print (x)

Tablica jednowymiarowa

```
tab1 = [1, 2, 3, 4, 5, 6, 7, 8, 9]
```

```
print (tab1[0]) # jeden indeks
```

```
1
```

1	2	3	4	5	6	7	8	9
---	---	---	---	---	---	---	---	---

```
for i in tab1:
```

```
    print (i, end=' ')
```

```
1 2 3 4 5 6 7 8 9
```

Tablica dwuwymiarowa

```
tab2 = [[1,2,3],[4,5,6],[7,8,9]]
```

```
print (tab2[0][0]) # dwa indeksy
```

```
1
```

1	2	3
4	5	6
7	8	9

```
for item in tab2:
```

```
    for number in item: # pętla zagnieżdżona
```

```
        print (number, end=' ')
```

```
    print ()
```

```
1 2 3
```

```
4 5 6
```

```
7 8 9
```


Ćwiczenie

- Stwórz tablicę z imionami 4 osób oraz wydrukuj imię, które jest w drugiej komórce w tablicy.
- Wydrukuj tylko pierwsze litery imion z tablicy w pętli for.

Zadania domowe

- Wydrukuj kwadraty liczb od 0 do 9. Skorzystaj z pętli **for**.

0 1 4 9 16 25 36 49 64 81

- Wydrukuj iloczyn liczb z zakresu od 10-15 pomnożonych przez 2. Skorzystaj z pętli **for**.

20 22 24 26 28 30

20

&

22

&

24