

Podstawy programowania Python

Wprowadzenie

Jolanta Bachan

2020-10-15

2020-10-22

Informacje do kontaktu

- email: jbachan@amu.edu.pl
 - jolabachan@gmail.com
- strona internetowa: <http://bachan.speechlabs.pl/>
- Konsultacje na MS Teams po wcześniejszym umówieniu

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
- Kurek, K. 2019. Wyrażenia regularne w Pythonie. Kurs video. Kompletnie vademecum. Videopoint

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 online poprzez rozwiązywanie zadań lub odpowiedzi na pytania
 - Trzy **spóźnienia** liczone są jako 1 nieobecność
 - Trzy odmowy udostępnienia ekranu liczone są jako 1 nieobecność
- Zgromadzenie 5 punktów w semestrze z aktywności na zajęciach i z zadań domowych
- Test: Napisanie prostych programów z wykorzystaniem funkcji poznanych na zajęciach
 - Test 1 – dla wszystkich
 - Test 2 – dla osób, które nie zdały testu 1 i/lub przekroczyły limit dwóch nieobecności
- Rejestracja w USOSie – *To już chyba macie za sobą, tak?*

POWODZENIA!

Termin zaliczenia

- Test 1: przedostatnie zajęcia w semestrze, 2021-02-04
- Test 2: ostatnie zajęcia w semestrze, 2020-02-11

Zainstaluj i przetestuj Pythona

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

```
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
```


Do zobaczenia za tydzień!