

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 13:00-14:00
 - czwartek 11:15-12:00

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. UnixTM 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: 4 lutego 2026
- Test poprawkowy: 26 lutego 2026

Zainstaluj i przetestuj Pythona

- Instrukcje znajdziesz na naszej stronie

https://bachan.speechlabs.pl/en/teaching/win2025_2026/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
```

Do zobaczenia za tydzień!