

# Podstawy programowania Python

## *Wprowadzenie*

Jolanta Bachan

# Informacje do kontaktu

- email: [jolabachan@gmail.com](mailto:jolabachan@gmail.com)
- strona internetowa: <http://bachan.speechlabs.pl/>
- dyżury w sali 312aB, 208 lub 209
  - środa, 9:00-9:45
  - środa, 14:45-15:30
- Proszę poinformować mnie wcześniej emailowo o Waszym przybyciu!

# 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

- van Rossum, Guido. 2004. Przewodnik po języku Python. Wydanie 2.3. PythonLabs. <<https://pl.python.org/docs/tut/tut.html>>
- 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>>
- Python – Dokumentacja. <http://www.python.org/doc/>
- Church, K.W. UnixTM for Poets.
- Graliński F., Junczys-Dowmunt M., Jassem K., PSI-Toolkit - A Natural Language Processing Pipeline. Computational Linguistics - Applications, Studies in Computational Intelligence. Heidelberg: Springer 2012.
- 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 dyżurach poprzez rozwiązywanie zadań lub odpowiedzi na pytania
  - Trzy **spóźnienia** 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

***POWODZENIA!***

# Termin zaliczenia

- Test 1: 2020-01-22
- Test 2: 2020-01-29



# 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
```



# 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

# Peçla for

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

3 4 5

Do zobaczenia za tydzień!