

# Podstawy programowania Python

## *Wprowadzenie*

Jolanta Bachan

# Informacje do kontaktu

- Email: [jbachan@amu.edu.pl](mailto:jbachan@amu.edu.pl)
  - [jolabachan@gmail.com](mailto: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. 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: 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](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
```

# 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('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
```

# Drukowanie

```
>>> name = 'John'
```

```
>>> age = 19
```

```
>>> print(f"My name is {name} and I'm {age} years old.")
```

```
>>> x = 10
```

```
>>> y = 2.6
```

```
>>> print(f"The calculation is: {x} / {y} = {x/y:.2f}")
```

The calculation is: 10 / 2.6 = 3.85

```
>>> import datetime
```

```
>>> today = datetime.datetime.today()
```

```
>>> print(f"{today:%B %d, %Y}")
```

October 29, 2024

# 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

# 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
```

# Tablica - usuwanie

```
>> tab = ['laptop', 'monitor', 'computer']
>> del tab[1]      # ['laptop', 'computer']
>> tab = ['laptop', 'monitor', 'computer']
>> tab.pop(0)     # ['monitor', 'computer']
>> tab = ['laptop', 'monitor', 'computer', 'laptop', 'monitor']
>> tab.remove('laptop')
           # ['monitor', 'computer', 'laptop', 'monitor']
>> tab.remove('laptop')
           # ['monitor', 'computer', 'monitor']
```

# Ćwiczenia

- 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.

# input()

```
>>> moj_tekst = input("Wpisz tekst: ")
```

```
Wpisz tekst: Python jest fajny.
```

```
>>> print (moj_tekst)
```

```
Python jest fajny.
```

```
>>> type(moj_tekst)
```

```
<class 'str'>
```

# input()

```
>>> moja_liczba = input("Wpisz liczbę: ")
```

```
Wpisz liczbę: 15
```

```
>>> print (moja_liczba)
```

```
15
```

```
>>> type(moja_liczba)
```

```
<class 'str'>
```

```
>>> moja_liczba_integer = int(moja_liczba)
```

```
>>> type(moja_liczba_integer)
```

```
<class 'int'>
```

`raw_input()` lub `input()`

```
>>> moj_tekst = raw_input()
```

Ja sama napisalam ten tekst.

```
>>> print (moj_tekst)
```

Ja sama napisalam ten tekst.

raw\_input()   lub   input()

```
>>> x = input()
```

```
4
```

```
>>> type(x)
```

```
<type 'int'>
```

```
>>> y = raw_input()
```

```
5
```

```
>>> type(y)
```

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

# input()

```
>>> x = input('How old are you? ')
```

```
How old are you? 20
```

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

```
20
```

```
>>> _/2_
```

```
10
```

```
# To jest komentarz.
```

```
# Zmienna _ przechowuje ostatnio wydrukowaną  
wyrtość
```

# Zadania

- Zapytaj użytkownika o liczbę, a następnie wydrukuj za pomocą pętli for kolejno liczby od 0 do przedostatniej liczby od tej podanej przez Ciebie.

Twoja liczba: 3

Output:

0

1

2

- Zapytaj użytkownika o to, dokąd chce jechać oraz kiedy, a następnie wydrukuj informację:

You want to go to *Warsaw* on *Friday*.

# Zadania

- 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

# Definicja/Funkcja (koniec)

```
>>> def AskName():  
    name = input("What is your name? ")  
    print ("Hello " + name + "!")
```

```
>>> AskName()
```

# Moduły

- Open → New Window
- Run → Run Module

```
name = input("What is your name? ")  
print (f"Hello {name}!")
```

# math

```
>>> import math
```

```
>>> math.sqrt(4)
```

```
2.0
```

```
>>> math.sqrt(16)
```

```
4.0
```

```
>>> math.pow(4,2)
```

```
16.0
```

# division

```
>>> from __future__ import division
```

```
>>> 4/3
```

```
1.3333333333333333
```

```
>>> round (_,3)
```

```
1.333
```

# break

```
>>> suma = 0
>>> for x in range(10):
    suma = suma+x
    print (x, suma)
    if suma>4:
        break
```

```
>>> print (suma)
```

6

# if... elif... else

```
>>> def sprawdzaj():  
    x = int(input('Podaj liczbę: '))  
    if x > 10:  
        print ('Liczba jest większa od 10.')  
    elif x < 0:  
        print ('Liczba ujemna.')  
    else:  
        print ('Liczba z przedziału od 0 do 10.')
```

```
>>> sprawdzaj()  
Podaj liczbę: -3  
Liczba ujemna.
```

# if... elif... elif... else

```
>>> # from sys import exit
>>> def sprawdzaj2():
    x = int(input('Podaj liczbę: '))
    if x > 10:
        print ('Liczba jest większa od 10.')
    elif x < 0:
        print ('Liczba ujemna.')
    elif x == 0:
        print ('Do zobaczenia!')
        exit()
    else:
        print ('Liczba z przedziału od 1 do 10.')
```

# pętla **while**

```
>>> text = ""
```

```
>>> while text != "exit":
```

```
    text = input('Podaj tekst: ')
```

Podaj tekst. Jola

Podaj tekst. ma

Podaj tekst. kota

Podaj tekst. exit

# pętla **while**

```
>>> suma = 0
```

```
>>> while suma < 100:
```

```
    liczba = input('Podaj liczbę: ')
```

```
    suma = suma + int(liczba)
```

Podaj liczbę: 20

Podaj liczbę: 30

Podaj liczbę: 50

# pętla **while**

```
>>> def dodawanie():  
    suma = 0  
    x = 1  
    while x != 0:  
        x = int(input("Podaj liczbę: "))  
        suma = suma + x  
    print (suma)
```

```
>>> dodawanie()
```

```
Podaj liczbę: 2
```

```
Podaj liczbę: 3
```

```
Podaj liczbę: 0
```

```
5
```

# pętla **while**

```
>>> def sprawdzaj3():  
    x = 1  
    while x != 0:  
        x = int(input('Podaj liczbę: '))  
        if x > 10:  
            print ('Liczba jest większa od 10.')  
        elif x < 0:  
            print ('Liczba ujemna.')  
        else:  
            print ('Liczba z przedziału od 0 do 10.')
```

```
def drukowanie_liczb():  
    for i in range(10):  
        if i < 5:  
            print (i)  
        else:  
            print ('The next number would be '+str(i)+'.')  
            break
```

```
>>> drukowanie_liczb()
```

```
0
```

```
1
```

```
2
```

```
3
```

```
4
```

```
The next number would be 5.
```

```
def drukowanie_liczb(x):  
    for i in range(10):  
        if i < x:  
            print (i)  
        else:  
            print ('The next number would be '+str(i)+'.')  
            break
```

```
>>> drukowanie_liczb(3)
```

```
0
```

```
1
```

```
2
```

```
The next number would be 3.
```

# try... except... else

```
>>> try:  
    print("To jest test.")  
except:  
    print("Coś poszło nie tak.")  
else:  
    print("Wszystko jest w porządku.")
```

To jest test.

Wszystko jest w porządku.

# try... except... else

```
>>> try:  
    print("To jest test.", name)  
except:  
    print("Coś poszło nie tak.")  
else:  
    print("Wszystko jest w porządku.")
```

Coś poszło nie tak.

# try... except NameError... except

```
>>> try:
```

```
    print('Test'-20)
```

```
except NameError:
```

```
    print("Variable y is not defined.")
```

```
except:
```

```
    print("Something else went wrong.")
```

Something else went wrong.

# try... except NameError... except

```
>>> try:
```

```
    print('Test'-y)
```

```
except NameError:
```

```
    print("Variable y is not defined.")
```

```
except:
```

```
    print("Something else went wrong.")
```

Variable y is not defined.

# try... except TypeError... except

```
>>> try:
```

```
    print('Test'+20)
```

```
except TypeError:
```

```
    print("Cannot concatenate string and integer.")
```

```
except:
```

```
    print("Something else went wrong.")
```

Cannot concatenate string and integer.

# try... except... finally

```
>>> try:
```

```
    print(test)
```

```
except:
```

```
    print("Zmienna 'test' nie jest zdefiniowana.")
```

```
finally:
```

```
    print("Próba 'try except' jest zakończona.")
```

```
>> def sumowanie():  
    suma = 0  
    while True:  
        try:  
            x = int(input("Podaj liczbę: "))  
            suma = suma + x  
        except ValueError:  
            print("Spróbuj jeszcze raz. To nie była liczba.")  
            break  
    print(suma)
```

pass

```
>>> for x in [0, 1, 2]:
```

```
    pass
```

```
>>> def myfunction():
```

```
    pass
```

# Tabliczka mnożenia *zagnieżdżona pętla for*

```
>>> for i in range(11):  
    for j in range(11):  
        print (i*j, end=' ')  
    print()
```

```
0 0 0 0 0 0 0 0 0 0  
0 1 2 3 4 5 6 7 8 9 10  
0 2 4 6 8 10 12 14 16 18 20  
0 3 6 9 12 15 18 21 24 27 30  
0 4 8 12 16 20 24 28 32 36 40  
0 5 10 15 20 25 30 35 40 45 50  
0 6 12 18 24 30 36 42 48 54 60  
0 7 14 21 28 35 42 49 56 63 70  
0 8 16 24 32 40 48 56 64 72 80  
0 9 18 27 36 45 54 63 72 81 90  
0 10 20 30 40 50 60 70 80 90 100
```

# Tabliczka mnożenia *zagnieżdżona pętla for*

```
>>> for i in range(11):
```

```
    for j in range(11):
```

```
        print ('%4d' % (i*j), end=' ')
```

```
    print()
```

```
0  0  0  0  0  0  0  0  0  0  0
0  1  2  3  4  5  6  7  8  9 10
0  2  4  6  8 10 12 14 16 18 20
0  3  6  9 12 15 18 21 24 27 30
0  4  8 12 16 20 24 28 32 36 40
0  5 10 15 20 25 30 35 40 45 50
0  6 12 18 24 30 36 42 48 54 60
0  7 14 21 28 35 42 49 56 63 70
0  8 16 24 32 40 48 56 64 72 80
0  9 18 27 36 45 54 63 72 81 90
0 10 20 30 40 50 60 70 80 90 100
```

# Sprawdź, czy wpisano liczbę

```
>>> def sprawdzaj():  
    x = input('Podaj liczbę: ')  
    if x.isdigit():  
        print ("Ok!")
```

```
>>> sprawdzaj()
```

Podaj liczbę: 5

Ok!

# Zadanie

Napisz funkcję, która zapyta użytkownika o podanie liczby oraz o tekst, a następnie wydrukuje tekst tyle razy, ile wynosi liczba podana przez użytkownika w jednym ciągu. Jeśli wydrukowany tekst będzie dłuższy lub równy 30 znakom, to wydrukuje "BRAWO!", a jeśli nie, to pojawi się komunikat "Spróbuj jeszcze raz!".

Skorzystaj z funkcji len, np. len(tekst)

Przykładowe działanie programu:

```
>>> dlugosc_tekstu()
```

Podaj liczbę: 5

Podaj tekst: Jolanta

JolantaJolantaJolantaJolantaJolanta

BRAWO!

Do zobaczenia za tydzień!