

# Kurs rozszerzony języka Python

## Wykład 1.

Marcin Młotkowski

12 października 2022

# Plan wykładu

- 1 O języku
- 2 Organizacja zajęć
- 3 Praca z Pythonem: narzędzia
  - Narzędziownia
- 4 Podstawowe typy w Pythonie
  - Typy całkowite
  - Typy zmiennoprzecinkowe
  - Inne wartości

# Plan wykładu

- 1 O języku
- 2 Organizacja zajęć
- 3 Praca z Pythonem: narzędzia
  - Narzędziownia
- 4 Podstawowe typy w Pythonie
  - Typy całkowite
  - Typy zmiennoprzecinkowe
  - Inne wartości

# Początki języka Python

Lata 90 — CWI Amsterdam, Guido van Rossum



Źródło: [wikimedia.org](https://commons.wikimedia.org/wiki/File:Guido_van_Rossum_-_PyCon_2009.jpg)

# Stan obecny

Python Software Foundation (PSF)



# Aktualna wersja (11 X 2022)

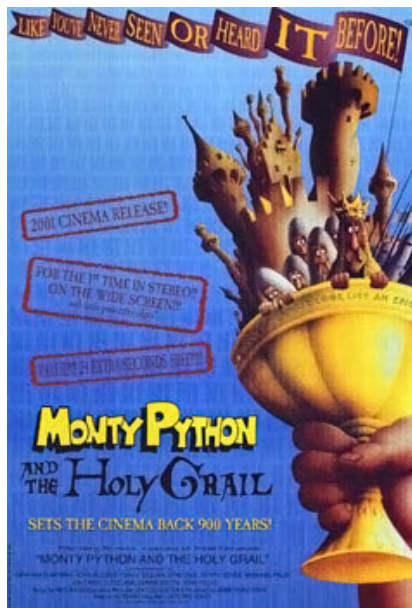
- seria 2.7: zakończony rozwój
- 3.10.7

## Aktualna wersja (11 X 2022)

- seria 2.7: zakończony rozwój
- 3.10.7

Ranking popularności języków programowania TIOBE: 1 pozycja  
<https://www.tiobe.com/tiobe-index/python/>

# Skąd pochodzi nazwa?





# Plan wykładu

- 1 O języku
- 2 Organizacja zajęć
- 3 Praca z Pythonem: narzędzia
  - Narzędziownia
- 4 Podstawowe typy w Pythonie
  - Typy całkowite
  - Typy zmiennoprzecinkowe
  - Inne wartości

# Wykłady

**Forma wykładu:** Wykład stacjonarny, s. 119;

**Slajdy, pliki, listy zadań** Kurs w serwisie

`http://skos.ii.uni.wroc.pl/`

**Oddawanie zadań:** SKOS, pracownia, do ustalenia przez prowadzącego;

**Moje konsultacje:** pok. 303, termin w systemie zapisy i USOS, zdalne (Teams).

# Komunikacja: pytania, uwagi, etc.

- podczas wykładu;
- forum na portalu skos.

# Organizacja pracowni

- Listy zadań będą na Skosie;
- Oddawanie list: na Skosie;
- Punktacja: sprawdziany na USOSWeb.

# Materiały pomocnicze



<https://docs.python.org/3/>



Mark Pilgrim, [Dive into Python](#)



Eric Matthes, [Python Crash Course](#)



Guido van Rossum, [The Python Language Reference](#)



Kenneth Reitz, Tanya Schlusser, [The Hitchhiker Guide to Python](#)



Bruce Eckel, [Python3 Patterns & Idioms Book](#)



Dusty Phillips, [Python 3 Object Oriented Programming](#)



Tarek Ziadé, [Expert Python Programming](#)

## Inne fajne książki



Bob Gregory, and Harry Percival, [Architecture Patterns with Python](#)



Ch. Durr & Jill-Jenn Vie, [Competitive Programming in Python](#)



Kristian Rother, [Pro Python Best Practices](#)



Bruce Eckel, [Thinking in Python](#)

# Plan kursu

- 1 Język Python  
typy podstawowe, wbudowane struktury danych, obiekty,  
programowanie funkcjonalne
- 2 Zastosowania Pythona  
przetwarzanie tekstu, bazy danych, interfejsy graficzne, I/O,  
protokoły sieciowe, wątki, SciPy, NumPy
- 3 Zagadnienia różne  
testowanie i dokumentowanie, refleksje, współpraca z innymi  
językami

# Pracownia

- listy z zadaniami do zaprogramowania

## Zaliczenie

Zdobycie przynajmniej połowy punktów z maksymalnej liczby punktów do zdobycia.



# Plan wykładu

- 1 O języku
- 2 Organizacja zajęć
- 3 Praca z Pythonem: narzędzia
  - Narzędziownia
- 4 Podstawowe typy w Pythonie
  - Typy całkowite
  - Typy zmiennoprzecinkowe
  - Inne wartości

# Python

- CPython 3.8;
- wersja 64 i 32-bitowa;
- MS Windows/Linux/macOS/FreeBSD.

- Linia poleceń
- idle3

## Uruchamianie programów

```
$ python3 plik.py
```

Co się dzieje

- 1 Kompilacja programu

## Uruchamianie programów

```
$ python3 plik.py
```

Co się dzieje

- 1 Kompilacja programu
- 2 Czasem tworzą się pliki \*.pyc w katalogu \_\_pycache\_\_

## Uruchamianie programów

```
$ python3 plik.py
```

Co się dzieje

- 1 Kompilacja programu
- 2 Czasem tworzą się pliki \*.pyc w katalogu \_\_pycache\_\_
- 3 Program jest wykonywany

## Uruchamianie programów

```
$ python3 plik.py
```

Co się dzieje

- 1 Kompilacja programu
- 2 Czasem tworzą się pliki \*.pyc w katalogu \_\_pycache\_\_
- 3 Program jest wykonywany

```
$ python3 plik.zip
```

# Narzędzia

- Edytory z podświetleniem składni: vim, gedit, geany, emacs, atom, sublime



# Narzędzia

- Edytory z podświetleniem składni: vim, gedit, geany, emacs, atom, sublime
- Zintegrowane narzędzia
  - idle3
  - PyCharm
  - Visual Studio Code

# Narzędzia

- Edytory z podświetleniem składni: vim, gedit, geany, emacs, atom, sublime
- Zintegrowane narzędzia
  - idle3
  - PyCharm
  - Visual Studio Code
- Jupyter

# Narzędzia

- Edytory z podświetleniem składni: vim, gedit, geany, emacs, atom, sublime
- Zintegrowane narzędzia
  - idle3
  - PyCharm
  - Visual Studio Code
- Jupyter
- publiczne dostępne serwisy webowe, np:  
<https://skulpt.org/>

# Plan wykładu

- 1 O języku
- 2 Organizacja zajęć
- 3 Praca z Pythonem: narzędzia
  - Narzędziownia
- 4 Podstawowe typy w Pythonie
  - Typy całkowite
  - Typy zmiennoprzecinkowe
  - Inne wartości

# bool

Stałe

`True`, `False`

Jako *fałsz* interpretuje się zero, `None`, puste stringi czy puste listy.

# bool

Stałe

`True`, `False`

Jako *fałsz* interpretuje się zero, `None`, puste stringi czy puste listy.

Operatory

`and`, `or`, `not`

Operatory te działają też z innymi typami.

# Przykład

```
lista = [1,2,3]
while lista:
    print(lista.pop())
```

# Liczby całkowite

```
googol = 10**100
```

```
0xAA
```

```
0o777
```

```
0b1001
```



# Specjalistyczne biblioteki

```
import random
```

```
random.choice([1,2,3,4,5,6,7,8,9])
```

```
random.random()
```

```
random.randint(2, 12)
```

# float

1.23

```
liczba_Avogadro = 6.02214076e23
```

```
srednica_atomu = 10E-10
```

# Moduł math

```
import math
```

```
math.sin(math.radians(180))
```

```
math.pi
```

```
math.gcd(9, 12)
```

## Dodatkowe wartości

NaN, +inf, -inf

Przykład: szukanie minimum

```
lista = [1.0, 2.0, 5.0, -12.0]  
m = float('+inf')
```

```
for x in lista:  
    if x < m:  
        m = x
```

# Jawne konwersje

```
int(3.14)
```

```
int("-3")
```

```
float(2)
```

```
float("-1024")
```

# Kłopoty z floatem

Czy prawdą jest

$$0.1 + 0.1 + 0.1 - 0.3 == 0.0$$

# Kłopoty z floatem

Czy prawdą jest

```
0.1 + 0.1 + 0.1 - 0.3 == 0.0
```

Remedium

```
import decimal as d
d.Decimal('0.1') + d.Decimal('0.1') + d.Decimal('0.1') -
    d.Decimal('0.3') == d.Decimal(0)
```

# Liczby zespolone

$1 + 2j$

$1 + 2j + 4 - 2j$



# Ułamki

## Moduł fractions

# Wartość 'pusta'

## None

Wartość ta jest interpretowana jako pusta/brak wartości. Jest to domyślna wartość argumentów, brak wartości przy odczycie z bazy danych, domyślna wartość zwracana przez funkcje.

# Wartość 'pusta'

## None

Wartość ta jest interpretowana jako pusta/brak wartości. Jest to domyślna wartość argumentów, brak wartości przy odczycie z bazy danych, domyślna wartość zwracana przez funkcje.

Idiomatyczne sprawdzanie 'pustości':

```
if x is None:
```

```
if y is not None:
```

# Inne wartości

NotImplemented

# Inne wartości

`NotImplemented`

Ellipsis:

...

Pytania?