

# PODSTAWY PROGRAMOWANIA W PYTHON

Dzień 2



# AGENDA

## DAY 2

Pamięć w komputerze

Zmienne, typy danych

operatory

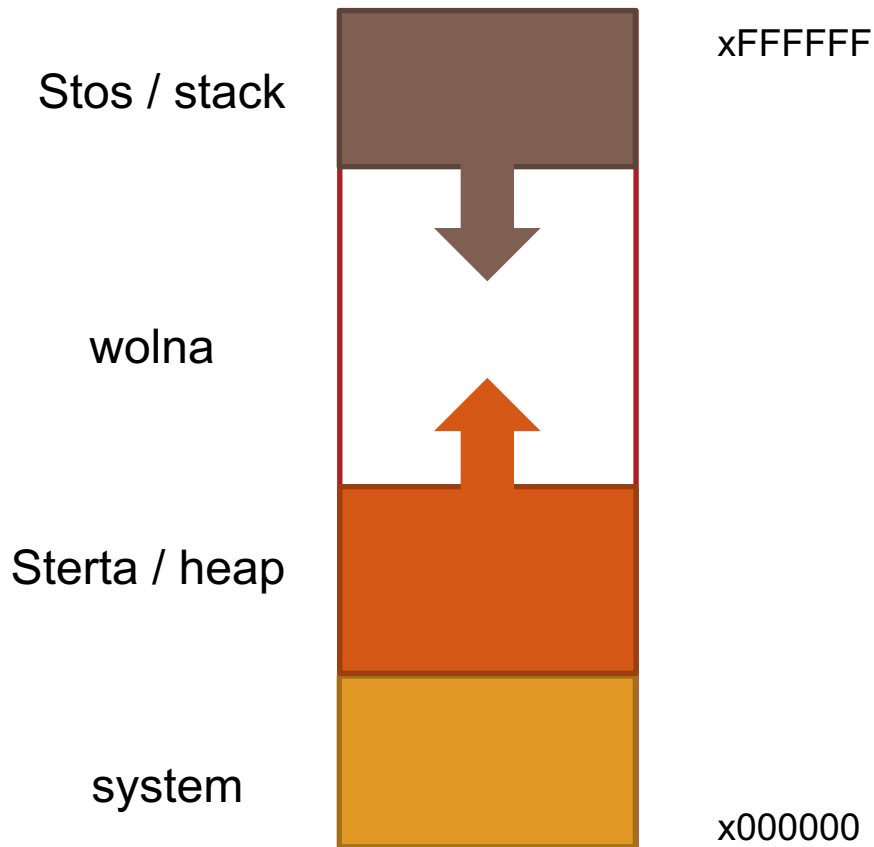
Instrukcje warunkowe

Code style

# 1. Pamięć

Typy danych, zmienne

# PAMIĘĆ A PROGRAMY



# PAMIĘĆ W PYTHON



# TYPY DANYCH

|            |         |                               |
|------------|---------|-------------------------------|
| 123        | - int   | - liczby całkowite            |
| 54.45      | - float | - liczby zmiennie-przecinkowe |
| "Ala"      | - str   | - łańcuchy znaków (string)    |
| True/False | - bool  | - prawda fałsz                |
| None       |         |                               |

listy, słowniki, tuple  
pliki, własne typy

# ZMIENNA

nazwany obszar pamięci, w którym znajduje się jakaś wartość  
pozwala na ponowne użycie wartości w innym miejscu w kodzie

```
moja_liczba = 124  
nazwisko = "Kowalski"  
czy_obecny = True
```

**= to jest znak przypisania**

# OPERATORY

Matematyczne:

`+, -, *, /, //, %, ^`

Logiczne:

`==, !=, <, >, <=, >=, in, is, and, or, not`



# OPERATOR PRZYPISANIA

=

najpierw wykonywane (obliczane) jest wyrażeniem, które znajduje się po prawej stronie znaku, następnie ta wartość jest przypisywana do zmiennej po lewej stronie znaku

```
wynik = 5 != 4 and 'a' not in 'Andrzej'
```

# METODY WBUDOWANE TYPÓW

Każdy typ danych posiada zdefiniowane metody (funkcje), które pozwalają na wykonanie różnych działań.

typ.funkcja()

„ala ma kota”.capitalize()

zdanie = „ala ma kota”

zdanie.capitalize()

# STRING

```
nazwisko = "Kowalski"
```

```
# długość
```

```
len(nazwisko) -> 8
```

```
# Indeksowanie
```

```
nazwisko[0] -> K
```

```
nazwisko[3] -> a
```

```
nazwisko[8] -> błąd, nie ma takiego indeksu!
```

# PROBLEM Z FLOAT

## Office Space



# PYTHONTUTOR.COM

[Start shared session](#)[What are shared sessions?](#)

Python 3.6

```
1 a = 'Hello world'
2 b = 34
3 b *= 3
4
5 print(b)
```

[Edit code](#) | [Live programming](#)

→ line that has just executed

→ next line to execute

**NEW!** Click on a line of code to set a breakpoint. Then use the Forward and Back buttons to jump there.

&lt;&lt; First

&lt; Back

Step 3 of 4

Forward &gt;

Last &gt;&gt;

Print output (drag lower right corner to resize)

Frames

Objects

Global frame

|   |               |
|---|---------------|
| a | "Hello world" |
| b | 34            |

Generate permanent link

Generate shortened link

Click the button above to create a permanent link to your visualization. To report a bug, paste the link along with a brief error description in an email addressed to [philip@pgbovine.net](mailto:philip@pgbovine.net)

Generate embed code

To embed this visualization in your webpage, click the 'Generate embed code' button above and paste the resulting HTML code into your webpage. Adjust the height and width parameters and change the link to **https://** if needed.

2.

# Instrukcje warunkowe

if – elif - else

# BLOK KODU

Instrukcja/wyrażenie :

Dwukropek rozpoczynający blok

Indentacja 1  
poziom (4 spacje)

Instrukcja

Instrukcja

Instrukcja :

instrukcja

Instrukcja/wyrażenie :

Instrukcja

Instrukcja

Instrukcja

# PSEUDOKOD

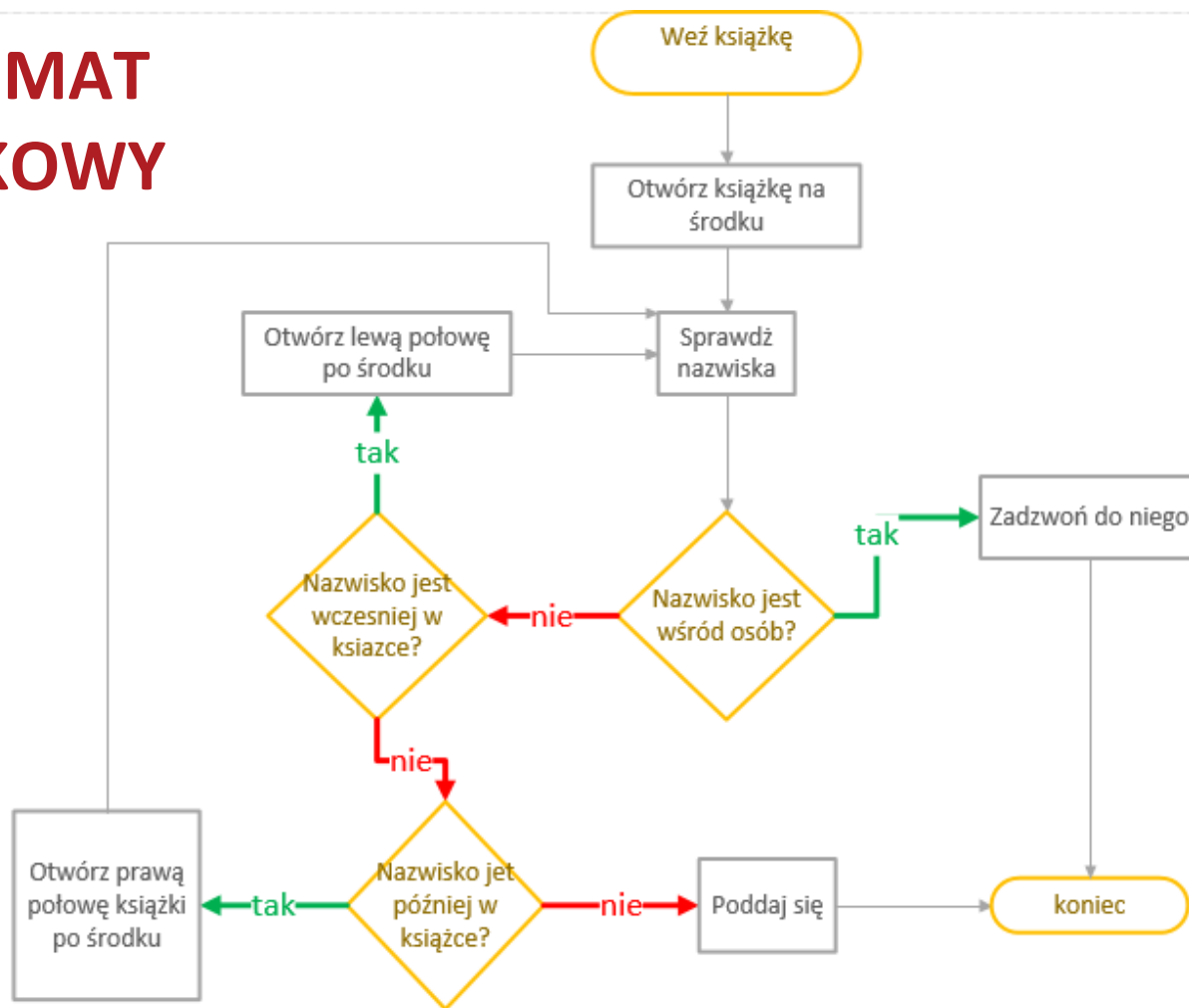
1. weź książkę telefoniczną
2. otwórz książkę na środku
3. zobacz nazwiska
4. jeśli "Wojtkowiak" jest wśród osób
5.       zadzwoń do niego
6. w przeciwnym razie jeśli "Wojtkowiak" jest wcześniej w książce
7.       otwórz lewą połowę po środku
8.       Idź do kroku 3
9. w przeciwnym razie jeśli "Wojtkowiak" jest później w książce
10.      otwórz prawą połowę po środku
11.      idź do kroku 3
12. w przeciwnym razie
13.      poddaj się



# Instrukcje warunkowe

1. weź książkę telefoniczną
2. otwórz książkę na środku
3. zobacz nazwiska
4. **jeśli "Wojtkowiak" jest wśród osób**
5.       zadzwoń do niego
6. **w przeciwnym razie jeśli " Wojtkowiak" jest wcześniej w książce**
7.       otwórz lewą połowę po środku
8.       Idź do kroku 3
9. **w przeciwnym razie jeśli " Wojtkowiak" jest później w książce**
10.      otwórz prawą połowę po środku
11.      idź do kroku 3
12. **w przeciwnym razie**
13.      poddaj się

# SCHEMAT BLOKOWY



# IF-ELIF-ELSE

## **if warunek:**

# jakiś kod wykonany gdy warunek prawdziwy

## **elif inny warunek:**

# kod wykonany gdy warunek w if był fałszywy

# warunek w tym elif musi być prawdziwy aby ten kod wykonać

## **elif inny warunek:**

# elif-ów może być wielu. lub żadnego, kod wew. elif

# wykona się tylko gdy wszystkie wyższe warunki były fałszywe

## **else:**

# przypadek domyślny, tu nie sprawdzamy warunku, kod w else

# będzie wykonany gdy wszystkie w if- elif były fałszywe

# else może być tylko jeden lub wcale

# Tablica logiczna

| A     | B     | A and B | A or B |
|-------|-------|---------|--------|
| True  | True  | True    | True   |
| True  | False | False   | True   |
| False | True  | False   | True   |
| False | False | False   | False  |

## 2. Code Style

# ZEN OF PYTHON

import this

*Beautiful is better than ugly.*

*Explicit is better than implicit.*

*Simple is better than complex.*

*Complex is better than complicated.*

*Flat is better than nested.*

*Sparse is better than dense.*

*Readability counts*

## PEP 8





# Thanks!!