

# Algorytmy i programowanie w języku Python - 04

Informatyka

# Listy w Pytonie

# Listy w języku Python

Aby utworzyć zmienne indeksowane w języku Python, możemy zdefiniować specjalną strukturę danych – **listę**, przechowującą one wiele wartości.

```
nazwa_listy = [element1, element2, ..., elementn]
```

```
moja_lista = [2, 4, 5, 6]
```

# Listy w języku Python

Do elementów listy odwołujemy się, podając nazwę listy i indeks elementu umieszczony w nawiasach kwadratowych, np.

$a[0]$ ,  $a[1]$ , ...,  $a[n - 1]$  – dla listy  $n$ -elementowej o nazwie  $a$ .

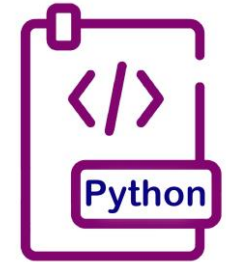
W języku Python pierwszy indeks jest zawsze równy 0.

Przykład:

```
moja_lista = [0] * 10
```

– oznacza zdefiniowanie listy o nazwie `moja_lista` składającej się z dziesięciu elementów, z których każdy jest równy zero

# Ćw-12 Python



1. Napisz program w którym utwórz listę `moja_lista`
  - a) Na utworzonej liście wyszukaj najmniejszy i największy element w zbiorze korzystając z funkcji `min()` i `max()`
  - b) Wypisz wyniki na ekran.
  - c) Zapisz program w pliku pod nazwą `cw-12_ImieNazwisko`

# Funkcje w Pytonie

# Funkcje w języku Python

W języku Python **funkcje** definiujemy słowem kluczowym `def` następnie podaje się nazwę funkcji (bez polskich znaków i spacji) i wstawia nawiasy, w których można umieścić parametr lub parametry, albo nie umieszczać żadnego.

Deklarację należy zakończyć dwukropkiem, który zagwarantuje, że po naciśnięciu klawisza **Enter** w następnym wierszu powstanie wcięcie.

Kolejne wiersze zawierają instrukcje do wykonania.

Polecenie `return`, które przekazuje szukaną wartość, kończy działanie funkcji.

# Funkcje w języku Python

W języku Python wszystkie podprogramy nazywane są **funkcjami**, dzielimy je na dwie grupy:

- funkcje ***niezwracające wartości***

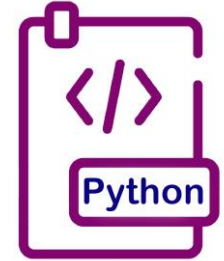
```
def nazwa_funkcji(lista_parametrów):  
    lista_instrukcji
```

- funkcje ***zwracające wartość***

```
def nazwa_funkcji(lista_parametrów):  
    lista_instrukcji  
    return wartość
```



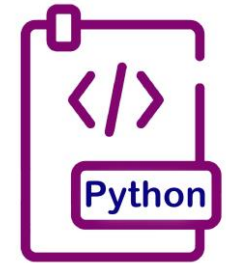
# Ćw-13 Python



Utwórz funkcję o nazwie "pole", która będzie wypisywać wartość pola kwadratu o boku 5.

- a) zacznij od słowa kluczowego `def`, w drugim wierszu zadeklaruj zmienną oznaczającą bok kwadratu,
- b) w trzecim wierszu napisz polecenie `return`, które przekazuje wartość i kończy działanie funkcji
- c) wypisz wyniki na ekran.
- d) Zapisz program w pliku pod nazwą `cw-13_ImieNazwisko`

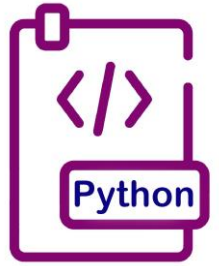
# Ćw-14 Python



Zmodyfikuj funkcję 'pole' z cw-13 - tak aby:

- a) pobierała wartość boku z klawiatury z pomocą funkcji `input()` oraz wypisała wartość pola kwadratu na ekran.
- b) Zapisz program w pliku pod nazwą `cw-14_ImieNazwisko`

# Ćw-15 Python



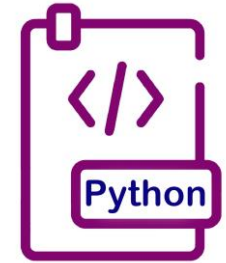
Napisz program w którym utworzysz listę `x`, do listy dodaj funkcję `def` 'najmniejszy' przyjmującą jako parametr listę: `def najmniejszy(lista):`

- a) Zadeklaruj w funkcji zmienną `y`, która przechowuje aktualnie najmniejszą liczbę, na początku powinna przyjmować pierwszy element z listy: `y = lista[0]`
- b) Teraz porównujemy wartość zmiennej `y` z kolejnymi elementami listy, z pomocą pętli `for`, która pod zmienną `i` będzie umieszczać w kolejnych przejściach elementy listy. Wewnątrz pętli instrukcja warunkowa sprawdzi czy `i` jest mniejsze niż `y`: `if i < y:`

Jeśli warunek jest spełniony nową wartością `y` powinna być obecna wartość `i`

- a) Po przejściu pętli zwracana jest wartość zmiennej `y` jako najmniejszej wartości na liście z pomocą polecenia `return y`,
- b) Należy wywołać funkcję podając jej nazwę w funkcji `print`, wartość zostanie wypisana na ekran.

# Ćw-16 Python



1. Zmodyfikuj ćw-15 tak aby funkcja `def` wyszukiwała największy element listy. W tym celu:
  - a) utwórz funkcję 'największy',
  - b) w instrukcji warunkowej zmień sprawdzanie czy kolejny element listy jest *większy* od zapisanego jako y elementu,
  - c) W ten sposób zostanie wyświetlony ten największy.

# Legenda

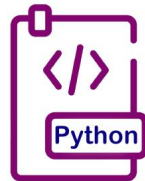
1. Wykład



2. Notatka z wykładu



3. Ćwiczenie 1



5. Pytanie



6. Informacje

