

# Funkcje

April 11, 2023

```
[ ]: Lab06 - Funkcje
```

```
[ ]: 1. Przykłady funkcji
```

```
"""  
Funkcje  
"""  
  
def wyswietl(a, b):  
    print(a, b)  
  
def dodaj(a, b):  
    return a + b  
  
wyswietl(5, 8)  
c = dodaj(4, 6)  
print(c)
```

```
[ ]: 2. Funkcje wewnętrzne
```

```
"""  
Funkcje wewnętrzne  
"""  
  
def funkcja1(a):  
    print(a)  
  
    def funkcja2():  
        print(a * 2)  
  
    funkcja2()  
  
funkcja1(5)
```

```

def oblicz(a, b):
    print(a * b)

test = oblicz
test(2, 4)

a = 10

if a < 5:
    def funkcja():
        print('liczba mniejsza od 5')
else:
    def funkcja():
        print('liczba większa lub równa 5')

funkcja()

```

### [ ]: 3. Argumenty funkcji

```

"""
Argumenty funkcji
"""

# Przykład 1
def wyswietl(a, b, c):
    print(a, b, c)

wyswietl(5, 10, 15)
wyswietl(2, c=6, b=4)

# Przykład 2
def wyswietl2(a, b=4, c=8):
    print(a, b, c)

wyswietl2(2)
wyswietl2(1, 2)
wyswietl2(3, c=6)

# Argumenty dowolne
def wypisz(*arg):

```

```
print(arg)

def wypisz2(**arg):
    print(arg)

wypisz(1, 2, 3)
wypisz2(a=1, c=3)
```

[ ]: 4. Zakres zmiennych

```
"""
Zakres zmiennych
"""

# Zakres lokalny
x = 5

def wypisz1(a):
    x = a
    print(x)

wypisz1(2)
print(x)

# Zakres globalny
y = 4

def wypisz2(b):
    global y
    y = b
    print(y)

wypisz2(8)
print(y)

# zakres nonlocal
def funkcja1():
    a = 5
    print(a)
```

```

def funkcja2():
    nonlocal a
    a = 10

    funkcja2()
    print(a)

funkcja1()

```

[ ]: 5. Wyrażenia `lambda`

```

"""
Wyrażenia lambda
"""

l = lambda a, b, c: a + b + c
print(l(2, 3, 4))

lista = [(lambda a: a * 2), (lambda a: a * 4), (lambda a: a * 6)]

for el in lista:
    print(el(2))

print(lista[1](3))

```

[ ]: 6. Generatory

```

"""
Generatory
"""

def generator(a):
    for i in range(a):
        yield i + 1

for i in generator(4):
    print(i, end=' ')

print('')
gen = generator(5)
print(next(gen))
print(next(gen))
print(next(gen))

```

## [ ]: 7. Zadania

Do każdego z podanych niżej zadań dołącz skrypt prezentujący działanie kodu

### Zadanie 1 (3 pkt)

-----

Napisz funkcję sprawdzającą, które z podanych w argumencie liczb są liczbami  
↪ doskonałymi.

Funkcja może przyjmować dowolną ilość argumentów.

W wyniku działania funkcji powinien powstać słownik w postaci {liczba: `True`,  
↪ liczba: `False`}

### Zadanie 2 (3 pkt)

-----

Napisz funkcję wypisującą liczby Catalana. Zmodyfikuj kod z zadania wykonanego  
↪ na jednym z poprzednich zajęć.

Funkcja powinna umożliwić użytkownikowi wyświetlenie liczb Catalana parzystych,  
↪ nieparzystych lub wszystkich

mniej niż od liczby podanej w argumencie. Domyślnie funkcja wyświetla  
↪ wszystkie liczby.

funckja (N, {p,n,a})

### Zadanie 3 (3 pkt)

-----

Napisz generator liczb pierwszych uzyskanych metodą sita Eratostenesa.  
Domyślnie generator wyświetla pierwszych 100 liczb pierwszych.

### Zadanie 4 (2 pkt)

-----

Napisz funkcję korzystającą z wyrażenia `lambda`. Funkcja powinna

- wypisywać N liczb podzielnych przez 2 , ale nie podzielnych przez 3
- umożliwić sortowanie liczb rosnąco i malejąco

### Zadanie 5 (2 pkt)

-----

Napisz skrypt działający jak prosta baza produktów spożywczych.

Zmodyfikuj kod z zadania wykonanego na jednym z poprzednich zajęć.

Zastosuj funkcje.

Zadanie 6 (3 pkt)

-----

Napisz skrypt z implementacją algorytmu Dijkstry. Wykorzystaj wiedzę z zajęć.

Zadanie 7 (4 pkt)

-----

Napisz funkcję, która zwraca sumę pierwszych  $N$  wyrazów rozwinięcia  $\exp$  w szereg [Maclaurina](http://www.pg.gda.pl/snm/pracownicy/jolanta.dymkowska/papers/szereg_M.pdf) funkcji  $e^x$  ([http://www.pg.gda.pl/snm/pracownicy/jolanta.dymkowska/papers/szereg\\_M.pdf](http://www.pg.gda.pl/snm/pracownicy/jolanta.dymkowska/papers/szereg_M.pdf)).  
Wartości  $x$  i  $N$  niech będą argumentami funkcji.