



# Funkcje 2

Wprowadzenie do języka Python (XI)





W tym zostanie pokazane, jak zwrócić i otrzymać wiele wartości. Obliczymy również liczby pierwsze; zostaną obliczone obszary dla różnych kształtów i zostanie przedstawiona funkcja main().

Aby zwrócić wiele wartości, oddziel wartości zwracane przecinkami.

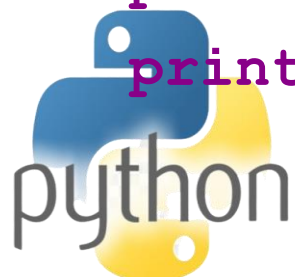
## KOD

```
def mno_dziel(num1, num2):  
    return (num1 * num2), (num1 / num2)
```

```
mnozник, dzielnik = mno_dziel(5, 4)
```

```
print("5 * 4 =", mnozник)
```

```
print("5/4 =", dzielnik)
```





## Zwróć listę liczb pierwszych

Liczba pierwsza może być podzielona tylko przez 1 i samą siebie. 5 jest liczbą pierwszą, ponieważ 1 i 5 to jedyne dzielniki liczby 5.

Liczba 6 jest złożona, ponieważ jest podzielna przez 1, 2, 3 i 6.

Otrzymamy zapytanie w programie o liczby pierwsze do wartości wejściowej. Następnie użyjemy pętli for i sprawdzimy, czy

moduł  $\neq 0$  dla każdej wartości aż do liczby do sprawdzenia. Jeśli moduł  $\neq 0$  oznacza to, że liczba nie jest pierwsza.





## KOD

```
def is_prime(liczba):  
    # Pętla przez liczby pierwsze od 2 do  
    # wartość do sprawdzenia  
    for i in range(2, liczba):  
        # Jeśli przy dzieleniu nie ma reszty, to wiemy o tym  
        # nie jest liczbą pierwszą  
        if (liczba % i) == 0:  
            return False  
    return True
```





```
def get_primes(maksymalna_liczba):  
    # Utwórz listę do przechowywania liczb pierwszych  
    lista_pierwszych = []  
    # Pętla przez liczby pierwsze od 2 do  
    # maksymalna żądana wartość  
    for liczba in range(2, maksymalna_liczba):  
        if is_prime(liczba):  
            lista_pierwszych.append(liczba)  
    return lista_pierwszych
```





```
zakres_do_sprawdzenia = int(input("Wyszukaj liczby pierwsze do :  
"))  
  
lista_pierwszych = get_primes(zakres_do_sprawdzenia)  
  
for pierwsza in lista_pierwszych:  
    print(pierwsza)
```





## Nieznana liczba argumentów

Możemy dostarczyć funkcji nieznaną liczbę argumentów za pomocą operatora (\*)[gwiazdka]

### KOD

```
def sumWszystko (*args) :
```

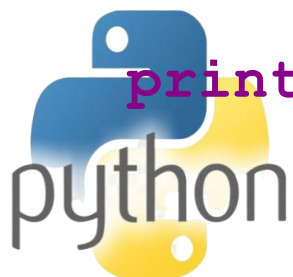
```
    suma = 0
```

```
    for i in args:
```

```
        suma += i
```

```
    return suma
```

```
print("Suma :", sumWszystko(1, 2, 3, 4))
```





## Droga do różnych funkcji i Main()

Tutaj pokierujemy do różnych funkcji w zależności od tego, jakiej figury pole chcemy obliczyć. Po raz pierwszy użyjemy również funkcji głównej.







## KOD

```
import math
```





```
# To prowadzi do właściwej funkcji związanej z obliczaniem pola  
figury
```

```
# Nazwa przekazanej wartości nie musi się zgadzać
```

```
def pobierz_obszar(ksztalt):  
    # Przełącz na małe litery, aby ułatwić porównanie  
    ksztalt = ksztalt.lower()  
    if ksztalt == "prostokat":  
        obszar_prostokat()  
    elif ksztalt == "kolo":  
        obszar_kolo()  
  
    else:  
        print("Proszę wprowadzić prostokąt lub okrąg")
```





```
# Utwórz funkcję, która oblicza obszar prostokąta
def obszar_prostokat():
    dlugosc = float(input("Podaj długość :"))
    szerokosc = float(input("Podaj szerokość:"))
    powierzchnia = dlugosc * szerokosc
    print("Powierzchnia prostokąta to", powierzchnia)
```





```
# Utwórz funkcję, która oblicza obszar okręgu
def obszar_kolo():
    promien = float(input("Wprowadź promień:"))
    powierzchnia = math.pi * (math.pow(promien, 2))
    # Sformatuj dane wyjściowe do 2 miejsc po przecinku
    print("Obszar okręgu to {:.2f}".format(powierzchnia))
```





```
# Często umieszczamy naszą główną logikę programowania w funkcji o  
nazwie main  
  
# Tworzymy to w ten sposób  
  
def main():  
    # Nasz program obliczy pole dla prostokątów lub okręgów  
    # Bez funkcji musielibyśmy stworzyć gigantyczną listę if, elif  
    # Zapytaj użytkownika jaki ma kształt  
    typ_kształtu = input("Pobierz obszar dla jakiego kształtu : ")  
    # Wywołaj funkcję, która przekieruje do właściwej funkcji  
    pobierz_obszar(typ_kształtu)
```





```
# Dzięki funkcjom bardzo łatwo jest zobaczyć, co się dzieje  
# Aby uzyskać więcej szczegółów, zapoznaj się z bardzo krótkimi  
określonymi funkcjami  
# Wszystkie definicje funkcji są ignorowane, a to wywołuje funkcję  
main()  
# do wykonania po uruchomieniu programu  
main()
```

