

FUNKCJE ZWRACAJĄCE WARTOŚĆ W JĘZYKU PYTHON

Funkcja w języku Python - wyodrębniona część programu, mająca unikatową nazwę oraz ustalony sposób wymiany danych z innymi częściami programu.

Funkcja, podprogram realizujący określone zadanie może być wielokrotnie wykorzystany w tym samym programie lub w innym. Stosowanie podprogramów zwiększa przejrzystość programu.

Główne funkcje w języku Python: **print()**, **input()**. Istnieje również możliwość tworzenia własnych funkcji.

Funkcje dzielimy na zwracające wartość i niezwracające wartości. Aby zastosować funkcję należy ją wcześniej zdefiniować oraz w odpowiednim momencie wywołać aby zadziałała.

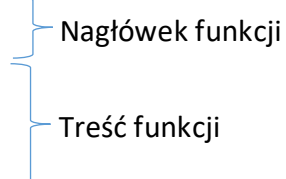
W nazwie funkcji można używać tylko liter, znaków podkreślenia i cyfr.

Funkcje powstały po to aby uniknąć powtarzalności kodu czyli zwiększyć jego czytelność.

Funkcję zwracającą wartość stosujemy, jeśli celem funkcji jest obliczenie i zwrócenie pewnej wartości do programu. Funkcja musi zawierać instrukcję **return** z wartością zwracaną do miejsca wywołania (np. do programu głównego)

Budowa funkcji zwracającej wartość:

```
def nazwa_funkcji(lista parametrów_formalnych) :  
    lista_instrukcji return wartość
```



Funkcje definiujemy w dowolnym miejscu programu (zazwyczaj na początku), ale przed pierwszym użyciem.

Treść funkcji (lista_instrukcji oraz instrukcja **return**) musi być wcięta względem nagłówka.

Wyrażenia **return** używamy do wyjścia z funkcji. Możemy opcjonalnie zwrócić w tym momencie jakąś wartość.

Aby wykonać treść instrukcji należy wywołać funkcję (podać nazwę funkcji a w nawiasach wartość parametrów).

Przykład 1. Definiowanie i wywołanie funkcji zwracającej wartość bez parametru

```
def hello():
    print("Hello world!!")
    return
print(hello())
```

def - służy właśnie do definiowania funkcji w Pythonie.

return - służy do zakończenia wykonywania kodu zawartego w ciele funkcji i pozwala na zwrot dowolnej wartości przy czym tylko 1 wartość można zwrócić jako rezultat wykonania funkcji.

W Pythonie, jeśli żadna wartość nie jest zwrócona to domyślnie przyjmowana jest wartość **None**.

W przeciwieństwie do **return** ze zwracaną wartością:

```
def hello():
    return "Hello world!"
print(hello())
```

Przykład 2. Definiowanie i wywołanie funkcji zwracającej wartość bez parametru.

Zdefiniuj funkcję *obwod* bez parametrów, zwracającą do programu głównego obwód kwadratu o boku *bok*. Wywołaj funkcję w programie głównym i wyprowadź wartość obwodu. Wartość zmiennej *bok* wprowadzaj z klawiatury.

```
def obwod():
    return 4 * bok

bok = int(input("Podaj bok kwadratu: "))
print("Obwód kwadratu o boku", bok, "wynosi", obwod())

input("Aby zakończyć program naciśnij ENTER")
```

Ćwiczenie 1.

1. Przepisz powyższy kod.
2. Zapisz plik pod nazwą **obwod_kwadratu**

Ćwiczenie 2.

1. Bazując na poprzednim ćwiczeniu, utwórz program obliczający pole kwadratu.
2. Zapisz plik pod nazwą **pole_kwadratu**

Ćwiczenie 3.

1. Otwórz plik `obwod_kwadratu` oraz `pole_kwadratu` jako jeden plik (program)
2. Zamień program na funkcję z parametrem.
3. W programie głównym wywołaj funkcję z **parametrem aktualnym** `x`, wprowadzanym z klawiatury jako liczba całkowita
4. Zapisz plik pod nazwą `obwod_i_pole_kwadratu_z_par`

Przykład 3. Definiowanie i wywołanie funkcji zwracającej wartość z parametrem

Podaj dwie liczby. Program ma wyświetlić większą z nich.

```
def max(x, y):
    if x > y:
        return x
    elif x == y:
        return "Liczby sa sobie równe"
    else:
        return y
x = int(input("Podaj 1 liczbę: "))
y = int(input("Podaj 2 liczbę: "))
print("Liczba większa:", max(x, y))
```

Przykład 4. Definiowanie i wywołanie funkcji zwracającej wartość z parametrem

Program sumujący dwie podane przez użytkownika liczby.

```
def suma(a, b):
    return a + b

a = int(input("Podaj 1 liczbę: "))
b = int(input("Podaj 2 liczbę: "))

print("Suma liczb wynosi:", suma(a, b))
```

Inny sposób zapisu:

```
def suma(a, b):
    s = a + b
    return s

a = int(input("Podaj 1 liczbę: "))
b = int(input("Podaj 2 liczbę: "))
wynik = suma(a, b)
print("Suma liczb wynosi:", wynik)
```

Przykład 5. Definiowanie i wywołanie funkcji zwracającej wartość z parametrem

Napisz program informujący czy liczba jest parzysta czy nieparzysta.

```
def parzysta(a):
    if a%2 == 0:
        return "parzysta"
    else:
        return "nieparzysta"
a = int(input("Podaj liczbę: "))
print("Podana liczba jest", parzysta(a))
```

Ćwiczenie 4.

1. Napisz program obliczający kwadrat dowolnej liczby n wprowadzonej przez użytkownika
2. Zapisz plik pod nazwą **kwadrat_n**

Ćwiczenie 5.

1. Zamień poniższe zadanie (realizujące obliczenie silni z liczby podanej przez użytkownika) na zadanie realizujące to samo ale za pomocą funkcji zwracającej wartość.

```
wynik = 1
n = int(input("podaj z jakiej liczby liczymy silnię: "))
for i in range(1, n + 1):
    wynik = wynik * i
print("Silnia wynosi", wynik)
```

2. Zapisz plik pod nazwą **silnia_funkcja**

Ćwiczenie 6.

Napisz funkcję o nazwie „kobieta”, której użytkownik przekaże jako argument znak wprowadzony z klawiatury („k” lub „m” - duże lub małe). Znak powinien odpowiadać płci użytkownika. Zadaniem funkcji o nazwie „kobieta” ma być sprawdzenie czy znak będący odpowiedzią użytkownika a więc i argumentem wywołania funkcji kobieta(a) jest równy znakowi „K” albo „k”.

- ✓ Jeśli tak („K” lub „k”) to funkcji zwróci rezultat „t”.
- ✓ Jeśli użytkownik wpisze „m” lub „M” to funkcji zwróci rezultat „m”.
- ✓ W przeciwnym przypadku zwróci „n”.

Funkcja jest wywołana w instrukcji warunkowej IF, ELIF, ELSE.

- ✓ Jeśli funkcja zwróci jako rezultat „t” to na ekranie pojawi się komunikat „Jesteś kobietą”.
- ✓ Jeśli rezultatem będzie „m” to pojawi się komunikat „Jesteś mężczyzną.”
- ✓ W przeciwnym przypadku (jeśli ktoś wpisze np. „x”) pojawi się komunikat „Podaj prawdziwe dane!”

Zapisz plik pod nazwą **k_czy_m**