

Python. Laboratorium 12. Funkcje. Part 1.

Wprowadzenie:

Funkcje w Pythonie to sposób na nadanie nazwy fragmentowi swojego skryptu(kodu), i możliwość późniejszego, wielokrotnego odwoływania się do niego z użyciem tej nazwy.

Na definicję funkcji składają się słowa:

- **Nazwa funkcji poprzedzona słowem def, która definiuje funkcje.**
- **Lista argumentów funkcji**, otoczona w nawiasach i oddzielona przecinkami. □
Instrukcje, które definiują funkcje, otoczone w nawiasach klamrowych, { }.
- **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**.

Ogólnie mówiąc funkcja może pobierać parametry i zwracać wartości, ale też nie musi. Kiedy używamy funkcji wówczas mówimy o wywołaniu funkcji, która nakazuje programowi wykonanie kodu zdefiniowanego w bloku kodu, a następnie wykonanie kodu znajdującego się po wywołaniu funkcji.

Przykład:

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

```
    c = a + b
```

```
    return c
```

```
def minus(a,b):
```

```
    c = a - b
```

```
    return c
```

```
print (suma(1,4))
```

```
print (minus(4,1))
```

Aby użyć funkcji to poza jej stworzeniem musimy ją jeszcze wywołać, inaczej będzie tak jakbyśmy nic nie zrobili. Wywołanie funkcji jak na powyższym przykładzie wygląda tak: `print (suma(1,4))`

W tym wywołaniu do zmiennej `a` zostanie przypisane 1, a do zmiennej `b` zostanie przypisane 4.

Funkcja `return` zwraca wartość `c` czyli sumę dwóch zmiennych `a` i `b`.

Przykład:

```
def dzielenie(dzielna, dzielnik):
```

```
    if(dzielnik == 0):
```

```
        return # zakoncz funkcje nic nie zwracajac
```

```
    else:
```

```
        return dzielna/ dzielnik
```

```
print (dzielenie(5, 1))
```

```
print (dzielenie(5, 0))
```

W tym wywołaniu do zmiennej dzielna zostanie przypisana wartość 5 a do zmiennej dzielnik 1 oraz w drugim wywołaniu 0. Pierwsze wywołanie zwróci wartość 5, drugie 0.

Przykład

```
def max(x, y):
```

```
    if x > y:
```

```
        return x
```

```
    else:
```

```
        return y
```

```
max(2,3); //wywołanie funkcji max z dwoma parametrami 2 i 3
```

Funkcja zwróci wartość 3 ponieważ 3 jest większe od 2.

Lista zadań:

1. Napisz skrypt pozwalający obliczyć sumę, różnicę, mnożenie, dzielenie oraz pierwiastkowanie liczb podanych z formularza.
2. Napisz skrypt zawierający formularz pozwalający na obliczenie pola i objętości różnych brył geometrycznych (kula, stożek, sześcian).
3. Napisz skrypt pozwalający na przeliczanie wysokości podanej w stopach na wysokość w metrach centymetrach, milimetrach i kilometrach.
4. Napisz program w którym zdefiniujesz dwie zmienne typu int. Program zapyta użytkownika: „Na jakiej wysokości lecimy? [w metrach]:”.
Jeśli będzie wysokość poniżej 5 km to program ma wypisać wartość podaną przez użytkownika „ km to za nisko!” czyli np. 2.5 km to za nisko!. W przeciwnym razie „Na tej wysokości jesteś już bezpieczny”.
5. Napisz program o nazwie BMI, obliczającą indeks masy ciała z dwóch wartości: masy i wzrostu przekazanych jako argumenty. Do obliczeń użyj następującego wzoru: $BMI = \frac{masa(kg)}{(wzrost(m))^2}$. Funkcja ma wyświetlać trzy komunikaty zależne od indeksu BMI:
 - BMI < 18,5: "Masz niedowagę"
 - 18,5 < BMI < 24,99: "Twoja waga jest prawidłowa"
 - BMI > 25,00: "Masz nadwagę!!!"

6. Napisz funkcję, która wyświetli postać binarną oraz szesnastkową podanej liczby całkowitej zapisanej w systemie dziesiętnym. Zaproponuj rozwiązanie pozwalające skonwertować postać binarną do szesnastkowego i w drugą stronę. Jak zmieni się program, kiedy będziemy chcieli dołączyć przeliczanie na system ósemkowy?
7. Napisz program, który pozwoli wyświetlić wpisane liczby dziesiętne w postaci liczb rzymskich np. liczba 900 w zapisie rzymskim wynosi CM. Program powinien również umożliwić odwrócić tą operację.