CYFROWA SZKOŁA
Wielkopolsk@ 2020

Funkcje

Wprowadzenie do języka Python (X)

Funkcje pozwalają na ponowne wykorzystanie kodu i ułatwiają zrozumienie kodu.

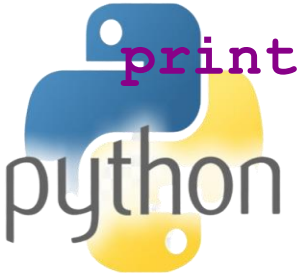
By stworzyć funkcję należy wpisać: def (zdefiniuj) nazwę funkcji, a następnie w nawiasach oddzieloną przecinkami listę wartości, które funkcja może zaakceptować.

Ta funkcja dodaje 2 wartości i zwraca sumę.

CYFROWA SZKOŁA
Wielkopolsk@ 2020

KOD

```
def dodaj_liczby(liczba_1, liczba_2):
    # Return zwraca wartość w razie potrzeby
    return liczba_1 + liczba_2



# Wywołujesz funkcję według nazwy, po której następuje przecinek
# wartości rozdzielonych w razie potrzeby, a wartość może lub nie może być zwrócone
print("5 + 4 =", dodaj_liczby(5, 4))
```



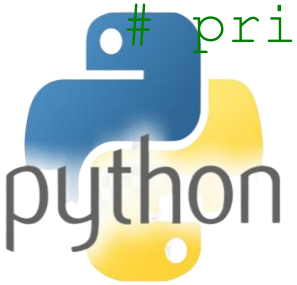
CYFROWA SZKOŁA
Wielkopolsk@ 2020

Zmienne lokalne w funkcji

Żadna zmienna zdefiniowana wewnątrz funkcji nie jest dostępna poza tą funkcją. Na przykład

KOD

```
def nazwa_przypisania():
    imie = "Ewa"



nazwa_przypisania()
# Zgłasza błąd nazwy
# print(imie)
```



CYFROWA SZKOŁA
Wielkopolsk@ 2020

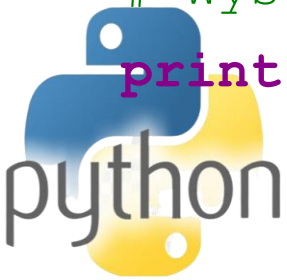
Zmienne globalne

Nie możesz zmienić zmiennej globalnej, nawet jeśli jest ona przekazywana do funkcji. To dlatego, że wartość a nie rzeczywista zmienna jest przekazywana do funkcji.

CYFROWA SZKOŁA
Wielkopolsk@ 2020

KOD

```
def zmien_nazwe(nazwa):
    # Próba zmiany globalnego
    nazwa = "Znak"


# Zmienna zdefiniowana poza funkcją nie może być zmieniona
# w funkcji korzystając z powyższego sposobu
imie = "Tomek"
# Spróbuj zmienić wartość
zmien_nazwe(imie)
# Wyświetla 'Tomek', mimo że funkcja próbuje to zmienić
print(imie)
```



Jeśli chcesz zmienić wartość, przekaż ją instrukcją ‘return’

KOD

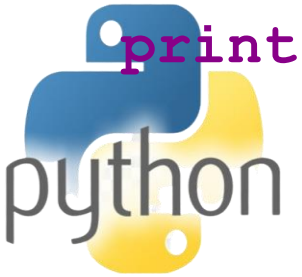
```
def zmien_nazwe_2():
    return "Znak"



nazwa = zmien_nazwe_2()
print(nazwa)
```

CYFROWA SZKOŁA
Wielkopolsk@ 2020

Możesz również użyć słowa kluczowego global, aby to zmienić.

KOD

```
gbl_nazwa = "Ewa"


def zmien_nazwe_3():
    global gbl_nazwa
    gbl_nazwa = "Ula"


zmien_nazwe_3()
print(gbl_nazwa)
```



CYFROWA SZKOŁA
Wielkopolsk@ 2020

Jeśli nie zwrócisz wartości, funkcja nie zwróci żadnej.

KOD

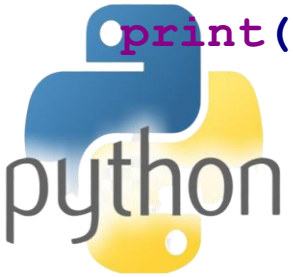
```
def podaj_sume(liczba1, liczba2):
    suma = liczba1 + liczba2



print(podaj_sume(5, 4))
```

CYFROWA SZKOŁA
Wielkopolsk@ 2020

WYKONAJ FUNKCJĘ is_float

Nie ma możliwości sprawdzenia, czy łańcuch zawiera liczbę zmiennoprzecinkową, więc zróbmy ją, definiując własną funkcję.

```
def is_float(str_val):
    try:
        # Jeśli ciąg nie jest float, float() wyrzuci
        # Błąd wartości
        float(str_val)
        # Jeśli istnieje wartość, którą chcesz zwrócić, użyj zwrotu
        return True
    except ValueError:
        return False
pi = 3.14
print("Czy Pi jest liczbą zmiennoprzecinkową :", is_float(pi))
```



CYFROWA SZKOŁA
Wielkopolsk@ 2020

Problem programistyczny do rozwiązania

W tym zadaniu otrzymasz równanie algebraiczne i rozwiążesz x. Wiedz, że x zawsze będzie otrzymywać pierwsze miejsce w wyrażeniu, a Ty zajmiesz się tylko dodawaniem. Oto próbka wywołania funkcji, a następnie jej wyjście.

```
# print(solve_eq("x + 4 = 9"))

# x = 5

# Rozwiązanie
```

CYFROWA SZKOŁA
Wielkopolsk@ 2020

```
def solve_eq(rownanie):
    x, dodaj, num1, równe, num2 = rownanie.split()
    # Konwertuj łańcuchy na liczby całkowite
    liczba1, liczba2 = int(num1), int(num2)
    # Konwertuj wynik na ciąg i połącz (konkatenuj)
    # to do ciągu "x = "
    return "x = " + str(liczba2 - liczba1)


print(solve_eq("x + 4 = 9"))
```