

Podstawy programowania (W11OPA-SI0072G)  
Wstęp do programowania (W11FTE-SI0141W)  
Wstęp do programowania (W11IKW-SI0080W)

### Laboratorium 3

1. Sprawdź działanie następujących poleceń w trybie interaktywnym interpretera:

```
In [1]: print("Jeden")
Jeden

In [2]: print("Jeden","dwa")
Jeden dwa

In [3]: print("Jeden","dwa","trzy")
Jeden dwa trzy

In [4]: print("Jeden","dwa","trzy",sep = "***")
Jeden***dwa***trzy

In [5]: print("Jeden","dwa","trzy",sep = "***", end = "$")
Jeden***dwa***trzy$
```

Przeanalizuj działanie wpisanych poleceń (w tym celu sięgnij do dokumentacji funkcji `print` <https://docs.python.org/3/library/functions.html#print>).

Zwróć uwagę na to, że funkcja `print()` może być wywoływana z różną liczbą argumentów. Ponadto, niektóre argumenty są przekazywane jako nazwane (`sep`, `end`).

2. Sprawdź działanie poniższego skryptu.

```
arguments_10.py X
1 def roznica(odjemna, odjemnik):
2     return odjemna - odjemnik
3
4 # argumenty przekazane jako pozycyjne
5 print(roznica(5,8))
6 # argumenty przekazane jako pozycyjne
7 print(roznica(8,5))
8
9 # pierwszy argumenty przekazany jako pozycyjny
10 # drugi jako argument nazwany
11 print(roznica(13, odjemnik = 8))
12 # argumenty przekazane jako nazwane
13 print(roznica(odjemna = 21, odjemnik = 13))
14 print(roznica(odjemnik = 21, odjemna = 44))
15
```

Zwróć uwagę na sposób przekazywania argumentów do funkcji (jako argumenty pozycyjne i/lub nazwane).

3. Sprawdź działanie poniższego skryptu.

```
arguments_11.py X
1 def roznica(odjemna, odjemnik = 0):
2     return odjemna - odjemnik
3
4 # odjemnik ma wartość domyślną
5 print(roznica(5))
6 # odjemnik ma wartość 3 (inną niż domyślną)
7 print(roznica(5,3))
8
```

Zwróć uwagę na wartości przyjmowane przez argumenty funkcji w zależności od sposobu wywołania.

4. NoneType jest typem wbudowanym, który przyjmuje tylko jedną wartość (None). Wielkość None może być wykorzystane do oznaczenia wielkości, które nie zostały zdefiniowane.

Przeanalizuj poniższy skrypt – obie funkcje: fun oraz fun2 zwracają wartość None.

```
return_none.py X
1 def fun():
2     print("Komunikat z funkcji fun()")
3
4 def fun2():
5     print("Komunikat z funkcji fun2()")
6     return
7
8 print(fun())
9 print(fun2())
10
```

Przeanalizuj poniższy skrypt, w którym funkcja jeden z argumentów funkcji f ma wartość domyślną None, co pozwala rozróżnić wywołanie z jednym i z dwoma argumentami.

Podstawy programowania (W11OPA-SI0072G)  
Wstęp do programowania (W11FTE-SI0141W)  
Wstęp do programowania (W11IKW-SI0080W)

```
arguments_09.py X
1  # -*- coding: utf-8 -*-
2  """
3  Definicja funkcji, w której opcjonalny argument
4  ma domyślną wartość None
5  """
6
7  def f(a, b = None):
8      if b is None:
9          print('Wywołanie z jednym argumentem', a)
10     else:
11         print('Wywołanie z dwoma argumentami', a, b)
12
13
14
15 f(1)
16 f(2,3)
```

5. Napisz program, który pozwala przeliczać na dżule wielkości wyrażone w kaloriach, ergach, elektronowoltach oraz kilowatogodzinach. W programie wydziel funkcję, która będzie dokonywała przeliczenia wielkości. W implementacji wykorzystaj wywołanie funkcji z argumentami nazwanymi.
6. Napisz program, który znajduje rozwiązania równania kwadratowego lub liniowego. W implementacji wykorzystaj funkcje opracowane w ramach rozwiązania zadań 4 i 5 poprzedniej listy zadań. Wykorzystaj argument o domyślnej wartości None do rozróżnienia wywołań o dwóch argumentach (równanie liniowej) oraz o trzech argumentach (równanie kwadratowe).
7. Zmodyfikuj implementację zadania z przeliczaniem temperatur tak, aby wykorzystywała argumenty nazwane i domyślne.

Karol Tarnowski  
Wrocław, 2024