PYTHON Funkcje

Funkcja jest wyodrębnionym fragmentem kodu, zamkniętym w pewnej strukturze posiadającej swoją nazwę. Odwołując się poprzez tę nazwę wykonujemy dany fragment kodu w dowolnym miejscu programu bez konieczności powielania tych samych treści.

Składnia:

```
def nazwafunkcji ( opcjonalne_parametry_wejściowe ) :
     instrukcje
```

Aby zdefiniować funkcję musimy użyć słowa kluczowego def, określić nazwę funkcji, użyć pary nawiasów okrągłych – mówiących nam, że mamy do czynienia właśnie z funkcją, następnie linię zakończyć symbolem dwukropka. Elementy te są wymagane. Pomiędzy nawiasami okrągłymi mogą się znaleźć opcjonalne parametry wejściowe, o których w kolejnej lekcji.

Wszystkie instrukcje należące do naszej funkcji muszą być oczywiście poprzedzone tabulatorem.

Przykład:

```
def witaj() :
   print("Witaj Świecie!")
```

Stworzyliśmy właśnie funkcję o nazwie witaj, której jedynym zadaniem jest wypisanie na ekran jednej linii tekstu.

Wprowadzając kod powyższej funkcji do pliku .py i uruchamiając go na ekranie nie zobaczymy nic. Dlaczego? Jak już wspomniałem wcześniej, aby wykorzystać / uruchomić funkcję należy się do niej odwołać po nazwie. Dodajmy zatem do naszego przykładu wywołanie funkcji:

```
def witaj() :
   print("Witaj Świecie!")
witaj()
```

Po uruchomieniu na ekran wypisze się oczywiście tekst: Witaj Świecie!

Przejdźmy do jeszcze jednego krótkiego przykładu pokazującego zalety funkcji, a mianowicie wykonajmy funkcję wielokrotnie, dla przykładu wykorzystując pętlę for:

```
def witaj() :
```

```
   print("Witaj Świecie!")
for x in range (1,11):
   witaj()
```

Efektem 10 linii z tekstem: Witaj Świecie!

Funkcja może przyjmować parametry wejściowe oraz zwracać parametry wyjściowe.

Parametry wejściowe

Parametry wejściowe, to wartości, które chcemy przekazać do wnętrza funkcji, na których następnie wykonywane są jakieś czynności czy działania.

Dla przykładu weźmy sobie funkcję, która przywita użytkownika po imieniu:

```
def witaj(imie) :
   print("Witaj", imie)
x = input("Podaj swoje imię: ")
witaj(x)
```

Jak widzimy, w definicji funkcji znalazł się parametr wejściowy o nazwie imie, który następnie pod tą samą nazwą został wykorzystany w wypisaniu.

Podczas wywołania funkcji również musiał zostać podany parametr wejściowy. Liczba parametrów wejściowych w wywołaniu funkcji oraz jej definicji musi być taka sama.

Od razu wspomnę, że od poprzedniego zdania jest wyjątek. Parametr wejściowy może mieć ustaloną wartość domyślną. Wówczas jeśli przy wywołaniu funkcji nie podamy wartości tego parametru, zostanie użyta właśnie wartość domyślna. Zobaczmy to na przykładzie:

```
def witaj(imie = "Jacek") :
   print("Witaj", imie)
witaj()
```

Jak widzimy definicja funkcji zawiera jeden parametr wejściowy o nazwie imie oraz o domyślnie ustawionej wartości (=) “Jacek”.

Funkcję wywołaliśmy nie podając wartości parametru wejściowego, zatem została wykorzystana wartość domyślna, a na ekran wypisany został tekst: Witaj Jacek.

Parametrów wejściowych może nie być, natomiast może ich być również więcej jak jeden. Zobaczmy kolejny przykład:

```
def witaj(imie, nazwisko) :
   print("Witaj", imie, nazwisko)
x = input("Podaj swoje imię: ")
y = input("Podaj swoje nazwisko: ")
witaj(x,y)
```

Tym razem mamy dwa parametry wejściowe imie oraz nazwisko.

Parametry wyjściowe

Przejdźmy od razu do kolejnego przykładu, program obliczający pole kwadratu po długości boku:

```
def pole_kwadratu(dlugosc_boku) :
   pole = dlugosc_boku**2
   print(pole)
x = int(input("Podaj długość boku kwadratu: "))
pole_kwadratu(x)
```

Mamy zdefiniowaną funkcję pole_kwadratu, która przyjmuje jeden parametr wejściowy, dokonuje na nim obliczeń, następnie wypisuje wynik na ekran. Musimy pamiętać, żeby do funkcji przekazać parametr o właściwym typie. W tym przypadku musi być to liczba.

Funkcja nie musi wypisywać wyniku na ekran, może go zwracać. Zmieńmy zatem powyższy program tak, aby funkcja zwracała wynik:

```
def pole_kwadratu(dlugosc_boku) :
   pole = dlugosc_boku**2
   return pole
x = int(input("Podaj długość boku kwadratu: "))
pole = pole_kwadratu(x)
print("Pole kwadratu o długości boku", x, "wynosi:", pole)
```

Jak widzimy, pojawiła się nowa linia wewnątrz funkcji w miejscu wypisania. Pojawiło się słowo kluczowe return, następnie nazwa wyliczonej zmiennej – parametr wyjściowy funkcji.

Skoro funkcja zwraca wynik, możemy go przypisać do dowolnej zmiennej podczas jej wywołania:

```
pole = pole_kwadratu(x)
```

Od tej chwili zmienna pole zawiera wynik działania funkcji, wartość, którą funkcja zwróciła.