

Funkcja w Kotlinie:

- to fragment kodu, który wykonuje określone zadanie
- może być wywołany w dowolnym miejscu programu.
- umożliwiają organizowanie kodu w logiczne bloki, które mogą być wielokrotnie używane bez konieczności ich powtarzania.
- może przyjmować argumenty (dane wejściowe) i zwracać wartość (wynik).

Argumenty

- to, co przekazujemy do funkcji.
- argument (czyli wartość taka jak 2 lub "Pizza") ląduje prosto w parametrze.

Parametr

- to nic innego jak zmienna lokalna:
 - ma swoją nazwę,
 - ma typ,
 - używana wewnątrz ciała funkcji.

Podstawowa składnia funkcji w Kotlinie

Funkcja w Kotlinie jest definiowana za pomocą słowa kluczowego `fun`, a jej składnia wygląda następująco:

```
fun nazwaFunkcji(parametr1: Typ, parametr2: Typ): TypZwracany {  
    // ciało funkcji  
  
    return wartość  
}
```

Funkcja zwraca **Int**

```
fun printSum(int1: Int, int2: Int) : Int {  
    val result = int1 + int2  
  
    return result  
}
```

Funkcja nic nie zwraca **Unit**

```
fun printSum(int1: Int, int2: Int): Unit {  
    val result = int1 + int2  
    println(result)  
}
```

Funkcje jako wyrażenia

```
fun max(a: Int, b: Int): Int = if (a > b) a else b
```

a nawet uprościć do

```
fun max(a: Int, b: Int) = if (a > b) a else b
```

Domyślne wartości parametrów

```
fun powitajImie(imie: String = "Świecie"): String {  
    return "Cześć, $imie!"  
}
```

Przykłady

```
fun pomnoz(a: Int, b: Int) = a * b  
  
fun main() {  
    println(pomnoz(4, 6)) // Wypisze: 24  
}
```

```
fun powitanie(imie: String = "Gość"): String {  
    return "Witaj, $imie!"  
}  
  
fun operujNaLiczbach(a: Int, b: Int, operacja: (Int, Int) -> Int):  
Int {  
    return operacja(a, b)  
}  
  
fun dodaj(a: Int, b: Int): Int {  
    return a + b  
}  
  
fun main() {  
    val suma = operujNaLiczbach(3, 7, ::dodaj) // :: to jest operator  
function reference  
    println(suma) // Wypisze: 10  
  
    val mnozenie = operujNaLiczbach(3, 7) { x, y -> x * y }  
    println(mnozenie) // Wypisze: 21  
}
```

 [Edit this page](#)