

# Czym jest metoda i jak działa?

## Wprowadzenie

W programowaniu **metoda** (zwana również funkcją) to blok kodu, który wykonuje określone zadanie. Metody są używane do organizowania kodu, unikania powtórzeń i ułatwiania zarządzania programem. Dzięki metodom możemy podzielić program na mniejsze, logiczne części, co sprawia, że kod jest bardziej czytelny i łatwiejszy do utrzymania.

---

## Składnia metody

W języku C# (używanym w Unity) metoda składa się z kilku elementów:

```
[modyfikator_dostępu] [typ_zwracany] [nazwa_metody]([parametry])
{
    // Ciało metody - kod, który zostanie wykonany
}
```

Przykład:

```
public void WyświetlWiadomosc(string wiadomosc)
{
    Debug.Log("Wiadomość: " + wiadomosc);
}
```

## Wyjaśnienie elementów

1. Modyfikator dostępu (**public**)
  - Określa z jakich części programu można wywołać metodę
  - Przykłady: public, private(automatyczne), protected
2. Typ zwracany (**void**)
  - Określa, jaką wartość metoda zwraca po zakończeniu
  - Jeśli metoda nie zwraca wartości używamy słowa kluczowego **void**.
3. Nazwa metody (**WyświetlWiadomosc**)
  - Nazwa metody powinna opisywać jej zadanie.
  - Nazwy metod w C# zapisuje się z wielkiej litery - choć **nie** jest to wymagane.
4. Parametry (**string wiadomosc**)
  - Są to dane, które przekazujemy do metody.
  - Parametry są opcjonalne - metoda może nie przyjmować żadnych argumentów.
5. Ciało metody
  - Blok kodu, który zostanie wykonany po wywołaniu metody.

## Jak działa metoda?

1. Definiowanie metody:
  - metodę definiujemy raz w kodzie, aby można było ją wielokrotnie używać
2. Wywołanie metody:
  - Aby wykonać kod znajdujący się w tej metodzie, musimy ją **wywołać**.

- Wywołanie metody polega na podaniu jej nazwy oraz przekazaniu ewentualnych argumentów

Przykład:

```
// Definicja metody wyświetl wiadomość
public void WyświetlWiadomosc(string wiadomosc)
{
    Debug.Log("Wiadomość: " + wiadomosc);
}

private void Start()
{
    WyświetlWiadomosc("Witaj, świecie!"); // wywołanie metody "WyświetlWiadomosc"
}
```

Warto zaznaczyć, że metody definiujemy poza ciałem innych metod, a dopiero samo wywołanie metody może dzieć się w ciele innej metody.

\*Ciekawostka w przypadku w którym metoda wywołuje samą siebie, mamy do czynienia z [rekurencją](#).