

# Podstawy programowania w JavaScript

---

## Podstawy programowania w JavaScript

---

### Wprowadzenie do JavaScript

JavaScript jest jednym z najważniejszych języków programowania w web development. Powstał w 1995 roku jako język skryptowy do rozszerzania możliwości stron internetowych. JavaScript działa po stronie klienta, co oznacza, że jest wykonywany w przeglądarce użytkownika, a nie na serwerze. Dzięki temu możliwe jest tworzenie dynamicznych, interaktywnych stron internetowych.

### Zmienne i typy danych

Zmienne w JavaScript służą do przechowywania danych. Możemy tworzyć zmienne za pomocą słów kluczowych `var`, `let` lub `const`. Każda zmienna może przechowywać różne typy danych, takie jak:

- Liczby (Number): np. `42`, `3.14`
- Łańcuchy znaków (String): np. `"Hello, world!"`
- Boolean: `true` lub `false`
- Obiekty (Object): np. `{name: "Jan", age: 30}`
- Tablice (Array): np. `[1, 2, 3, 4]`
- Undefined: wartość zmiennej, która nie została zainicjalizowana.

### Operatory

JavaScript posiada szeroki wachlarz operatorów, które służą do wykonywania różnych operacji na danych:

- Operatory arytmetyczne: `+`, `-`, `*`, `/`, `%` (modulo), `**` (potęgowanie)
- Operatory przypisania: `=`, `+=`, `-=`, `*=`, `/=`
- Operatory porównania: `==`, `===`, `!=`, `!==`, `>`, `<`, `>=`, `<=`
- Operatory logiczne: `&&` (AND), `||` (OR), `!` (NOT)

### Instrukcje warunkowe

Instrukcje warunkowe pozwalają na podejmowanie decyzji w kodzie na podstawie warunków logicznych. Najczęściej używane są:

- `if`: Sprawdza, czy warunek jest prawdziwy.
- `else`: Wykonywane, gdy warunek `if` jest fałszywy.

- else if: Sprawdza kolejny warunek, jeśli poprzedni był fałszywy.
- switch: Wykorzystuje wiele możliwych warunków do wyboru jednej z kilku ścieżek.

## Pętle

Pętle umożliwiają wielokrotne wykonywanie fragmentu kodu. W JavaScript mamy różne typy pętli:

- for: Używana, gdy znana jest liczba iteracji.
- while: Wykonuje się, dopóki warunek jest prawdziwy.
- do...while: Podobna do while, ale warunek sprawdzany jest po wykonaniu bloku kodu.

## Funkcje

Funkcje to bloki kodu, które mogą być wielokrotnie wywoływane. Są zdefiniowane za pomocą słów kluczowych function. Funkcje mogą przyjmować argumenty i zwracać wartość.

Przykład definicji funkcji:

```
function suma(a, b) {  
  return a + b;  
}
```

Można ją wywołać w następujący sposób:

```
let wynik = suma(5, 3); // wynik wynosi 8
```

## Przykłady zastosowań zmiennych i typów danych

1. Stworzenie zmiennej przechowującej nazwę użytkownika i jego wiek:

```
let imie = "Anna";  
let wiek = 25;
```

2. Operacje arytmetyczne:

```
let a = 10;  
let b = 20;  
let suma = a + b; // 30  
let roznica = a - b; // -10
```

## Ćwiczenie z instrukcjami warunkowymi

1. Napisz program, który sprawdzi, czy liczba jest dodatnia, ujemna, czy równa zero:

```
let liczba = -5;  
if (liczba > 0) {  
  console.log("Liczba jest dodatnia");  
} else if (liczba < 0) {  
  console.log("Liczba jest ujemna");  
}
```

```
} else {  
console.log("Liczba jest równa zero");  
}
```

2. Ćwiczenie z użyciem switch:

```
let kolor = "czerwony";  
switch (kolor) {  
case "czerwony":  
console.log("Kolor jest czerwony");  
break;  
case "zielony":  
console.log("Kolor jest zielony");  
break;  
default:  
console.log("Inny kolor");  
}
```

## Ćwiczenie z pętlami

1. Wyświetlanie liczb od 1 do 10 za pomocą pętli for:

```
for (let i = 1; i <= 10; i++) {  
console.log(i);  
}
```

2. Tworzenie tablicy liczb i ich sumowanie:

```
let liczby = [10, 20, 30, 40];  
let suma = 0;  
for (let i = 0; i < liczby.length; i++) {  
suma += liczby[i];  
}  
console.log("Suma: " + suma); // Suma: 100
```

## Ćwiczenie z funkcjami

1. Stwórz funkcję, która obliczy obwód prostokąta:

```
function obwodProstokata(dlugosc, szerokosc) {  
return 2 * (dlugosc + szerokosc);  
}  
console.log(obwodProstokata(5, 10)); // 30
```

2. Napisz funkcję, która sprawdzi, czy liczba jest parzysta:

```
function czyParzysta(liczba) {  
return liczba % 2 === 0;  
}
```

```
}  
console.log(czyParzysta(4)); // true  
console.log(czyParzysta(7)); // false
```

Zadania do samodzielnego wykonania

## Zadanie 1

Napisz program, który będzie przyjmować liczbę od użytkownika i wyświetli, czy liczba ta jest dodatnia, ujemna, czy równa zero. Następnie, w zależności od wyniku, funkcja powinna zwrócić odpowiedni komunikat.

## Zadanie 2

Stwórz tablicę liczb całkowitych, a następnie napisz funkcję, która obliczy sumę liczb znajdujących się w tej tablicy. Użyj pętli for lub while.

## Zadanie 3

Napisz program, który dla żądanej liczby zwróci jej wartość silni, czyli iloczynu wszystkich liczb naturalnych od 1 do tej liczby. Użyj pętli for i funkcji.