

# Wprowadzenie do JavaScript

---

## Wprowadzenie do JavaScript

---

### 1. Co to jest JavaScript i do czego służy?

JavaScript (JS) to język programowania, który jest kluczowym elementem tworzenia dynamicznych stron internetowych. Pozwala na interakcję z użytkownikami i manipulację treścią strony bez konieczności przeładowywania jej.

- **W skrócie:** HTML tworzy strukturę strony, CSS nadaje jej styl, a JavaScript dodaje funkcjonalność i interaktywność.
- **Przykłady użycia JavaScript:**
  - Walidacja formularzy na stronie
  - Tworzenie animacji i efektów wizualnych
  - Obsługa kliknięć, przesunięć i innych działań użytkownika
  - Przetwarzanie i wyświetlanie danych pochodzących z API (np. dane pogodowe)

JavaScript jest wszechobecny w świecie internetu, co czyni go jednym z najważniejszych języków dla programistów front-endowych i full-stack.

### 2. Historia i rozwój języka JavaScript

- **Powstanie:** JavaScript został stworzony w 1995 roku przez Brendana Eichę dla firmy Netscape. Jego początkowa nazwa to **Mocha**, później zmieniona na **LiveScript**, a ostatecznie na JavaScript.
- **Szybki rozwój:** Został stworzony w zaledwie 10 dni jako narzędzie do dodawania interakcji na stronach internetowych. JavaScript szybko zdobył popularność, a jego standardy zaczęły być rozwijane przez organizację **ECMA International**.
- **ECMAScript (ES):** Oficjalna nazwa standardu JavaScript. Kolejne wersje ECMAScript (ES) wprowadzały nowe funkcje i usprawnienia, z czego najważniejsze to:
  - **ES5** (2009): Stabilna wersja, którą wspierają wszystkie przeglądarki.
  - **ES6** (2015) i nowsze: Znanie jako ES2015 i późniejsze. Wprowadziły wiele nowoczesnych funkcji, takich jak `let`, `const`, klasy, funkcje strzałkowe (`arrow functions`), moduły oraz asynchroniczne `async/await`.

### 3. Wprowadzenie do ES6+ (najnowsze wersje ECMAScript)

JavaScript rozwija się dynamicznie, a najnowsze wersje ES (ECMAScript) przynoszą szereg nowych możliwości, które ułatwiają pisanie bardziej czytelnego i wydajnego kodu.

- **let i const**: Nowe sposoby deklaracji zmiennych. Są bardziej przewidywalne i bezpieczne niż tradycyjne `var`.
- **Funkcje strzałkowe** (`arrow functions`): Krótsza i bardziej elegancka forma zapisu funkcji.
- **Moduły ES6**: Umożliwiają podział kodu na mniejsze pliki, co ułatwia zarządzanie dużymi projektami.
- **Promisy i async/await**: Wprowadzone do obsługi asynchroniczności w kodzie, co ułatwia współpracę z API i obsługę danych.

## 4. Sposoby dodawania JavaScript do strony

JavaScript może być dodawany do strony na kilka sposobów, w zależności od potrzeb i struktury projektu:

### 4.1. Wewnątrz HTML (`<script>`)

Najprostszym sposobem dodania JavaScriptu jest umieszczenie skryptu bezpośrednio w pliku HTML za pomocą elementu `<script>`:

```
<!DOCTYPE html>
<html lang="pl">
<head>
  <meta charset="UTF-8">
  <title>Moja Pierwsza Strona</title>
</head>
<body>
  <h1>Witaj na mojej stronie!</h1>
  <script>
    console.log("Witaj świecie! To jest mój pierwszy skrypt
JavaScript!");
  </script>
</body>
</html>
```

W ten sposób kod JavaScript jest interpretowany bezpośrednio w przeglądarce użytkownika, kiedy strona jest ładowana.

Plusy: Proste do szybkiego testowania i małych projektów.

Minusy: Nie nadaje się do dużych projektów – utrudnia organizację kodu i utrzymanie.

### 4.2. Zewnętrzny plik .js

Dla większych projektów lepszym rozwiązaniem jest oddzielenie JavaScriptu od HTML-a i zapisanie go w osobnym pliku .js. W pliku HTML odnosi się do niego poprzez:

```
<!DOCTYPE html>
<html lang="pl">
```

```
<head>
  <meta charset="UTF-8">
  <title>Moja Strona</title>
  <script src="skrypt.js" defer></script>
</head>
<body>
  <h1>Witaj ponownie!</h1>
</body>
</html>
```

Zalety:

Łatwiejsza organizacja kodu, szczególnie w większych projektach.

Ułatwia ponowne użycie kodu na różnych stronach.

defer i async – parametry przyspieszające ładowanie strony:

defer: Skrypt łąduje się równocześnie z HTML-em, ale wykonuje dopiero po załadowaniu całej strony.

async: Skrypt wykonuje się jak najszybciej, ale może to powodować problemy z ładowaniem elementów.

### 4.3. Wewnątrz atrybutów HTML

JavaScript można również osadzić bezpośrednio w atrybutach elementów HTML, np.:

```
<button onclick="alert('Kliknąłeś przycisk!')">Kliknij mnie</button>
```

Plusy: Przydatne dla bardzo prostych interakcji.

Minusy: Trudne do utrzymania w większych projektach i niezalecane dla współczesnych aplikacji.

## Podsumowanie

JavaScript to fundamentalny język programowania dla każdego, kto chce tworzyć dynamiczne strony internetowe. Jest to niezbędne narzędzie w pracy każdego programisty front-endowego, a jego znajomość otwiera drzwi do bardziej zaawansowanych technologii i frameworków takich jak React, Vue, czy Angular.