

Sposoby dołączania skryptów JavaScript do pliku HTML

Skrypt wewnętrzny (inline, w pliku HTML)

- Kod JS można pisać **bezpośrednio w pliku HTML**, wewnątrz znacznika `<script>`.
- Najczęściej wrzucamy go do **sekcji `<head>`** (np. dla konfiguracji) albo na końcu **`<body>`** (żeby kod uruchomił się po wczytaniu strony).

```
<!DOCTYPE html>
<html lang="pl">
  <head>
    <meta charset="UTF-8" />
    <title>Przykład</title>
  </head>
  <body>
    <h1>Witaj!</h1>

    <script>
      alert("Skrypt wewnętrzny działa!");
    </script>
  </body>
</html>
```

👉 Plus: szybkie i proste.

👉 Minus: bałagan, jeśli kodu jest dużo.

Skrypt zewnętrzny (najczęściej używany)

- Kod JS zapisujemy w **osobnym pliku** (np. `script.js`).
- W HTML podłączamy go przez atrybut `src`:

```
<!DOCTYPE html>
<html lang="pl">
  <head>
    <meta charset="UTF-8" />
    <title>Przykład</title>
    <script src="script.js"></script>
  </head>
  <body>
    <h1>Witaj!</h1>
  </body>
</html>
```

👉 Plus: porządek, łatwiej testować i wielokrotnie używać.

👉 Minus: wymaga dodatkowego pliku.

Atrybuty defer i async (dla skryptów zewnętrznych)

- **Problem:** przeglądarka czyta HTML od góry, i gdy napotka <script>, **zatrzymuje się**, żeby go wczytać → strona może się opóźnić.
- Rozwiązania:
- defer
 - Skrypt ładuje się w tle, **ale wykona się dopiero po załadowaniu całego HTML-a**.
 - Idealny dla skryptów, które działają na elementach strony.

```
<script src="script.js" defer></script>
```

- async
 - Skrypt ładuje się w tle i **wykonuje od razu, gdy tylko się załaduje**.
 - Dobre do niezależnych rzeczy (np. reklamy, statystyki).

```
<script src="script.js" async></script>
```

Skrypt „inline” w atrybutach HTML (raczej unikać)

- Można napisać kod JS bezpośrednio w atrybutach HTML, np.:

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

👉 Plus: działa szybko.

👉 Minus: totalny chaos przy większych projektach – **niezalecane** w praktyce.