JS - sposoby osadzania, pierwsze zmienne, operatory i interakcje

Hello world!

- **<script>** - wewnątrz osadzamy kod JS
- **console.log("Cześć!")** - wypisujemy coś na tzw. konsolę
- kod JavaScript wykonuje się tylko raz
- **alert("Cześć!")**

Sposoby osadzania

wewnątrz <script>

zwykle na końcu dokumentu
przed zamykającym znacznikiem **</body>**

w oddzielnym pliku

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

rekomendowane rozwiązanie

atrybut defer

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

rekomendowany

odkłada wykonanie JavaScriptu do momentu załadowania drzewa DOM

nie mylić z **async**

w atrybutach np. **onclick**

```
<button onclick="alert('Akcja!')">Pokaż komunikat</button>
```

zła praktyka
zanieczyszczamy HTML-a podobnie tak jak w przypadku atrybutu style
nieefektywna metoda obsługi zdarzeń

Zmienne

kontenery / pojemniki, które przetrzymują wartości

Składnia

```
let nazwa = wartość;
```

Nazwa zmiennej

- wielkość liter ma znaczenie
- musi zaczynać się od litery, _ lub $
- może zawierać cyfry
- jest parę zarezerwowanych słów, które nie mogą być nazwą zmiennej, choćby **let**
- **https://mothereff.in/js-variables**
- konwencja: używamy liter alfabetu łacińskiego i cyfr oraz camelCase

Operatory

= operator przypisania

+ - * /

Zmienne z tekstem

```
let tekst = "Dzień dobry!"
```

rekomendacja: cudzysłów, nie apostrof

Zmienne z elementem HTML (węzłem, nodem)

```
let naglowek = document.querySelector(".naglowek");
```

Co mogę zrobić z elementem?

Dodać mu klasę

```
naglowek.classList.add("nowaKlasa");
```

Usunąć mu klasę

```
naglowek.classList.remove("nowaKlasa");
```

Przełączyć mu klasę

```
naglowek.classList.toggle("nowaKlasa");
```

Zmienić jego zawartość tekstową

```
naglowek.innerText = "Nowy tekst!"
```

Zmienić jego zawartość HTML

```
naglowek.innerHTML = "Nowa treść ze <span>znacznikiem HTML</span>.";
```

Usunąć go

```
naglowek.remove();
```

i wiele więcej...

Pierwsza interakcja

Potrzebujemy przycisku

```
<button class="przycisk">Usuń nagłówek</button>
```

Dodajemy obsługę zdarzenia

```
przycisk.addEventListener("click", () => {
  naglowek.remove();
});
```