

Akcja i reakcja

Mikołaj Dyczkowski
Frontend 4 Beginners

Czym jest JavaScript?

- Skryptowy język programowania
- Dynamicznie i słabo typowany
- Jeden z najpopularniejszych języków programowania na świecie.

Co możemy zrobić przy pomocy JS?

(Prawie) wszystko.

- Dynamicznie zmieniać zawartość strony
- Tworzyć skomplikowane animacje
- Gry przeglądarkowe
- Reagować na interakcje użytkownika ze stroną
- Zaciągać dane z zewnętrznego API (źródła danych)

Jak załączyć JS do strony

```
<script>  
    ...kod...  
</script>
```

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

defer i async - more info

First dive

Codepen

Obiekt `window`

reprezentuje otwartą kartę w przeglądarce, w której znajduje się nasza strona.

`window.location`

`window.alert`

`window.console`

i wiele innych. Do zmiennych zawartych w obiekcie `window` możemy się odwoływać również bezpośrednio (np. `console.log` zamiast `window.console.log`). Tak samo w drugą stronę - globalne zmienne w naszym kodzie są dostępne przez obiekt `window`.

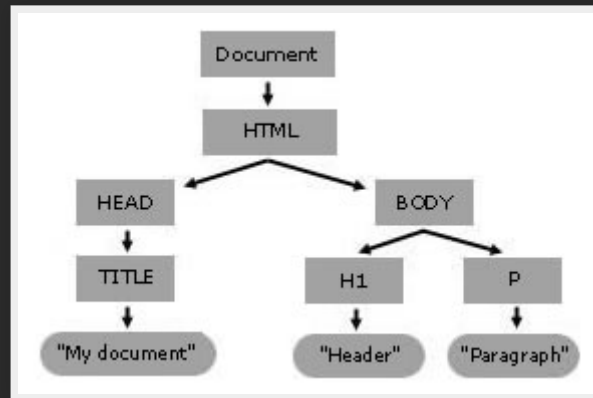
Document Object Model

Przedstawienie struktury strony w sposób zrozumiały dla języków programowania (najczęściej JavaScriptu)

Pozwala nam manipulować stroną i jej zawartością.

Obiekt `document`

reprezentuje samą stronę i jest pierwszym elementem drzewa DOM.



Daje nam dostęp do wielu informacji na temat naszej strony i oferuje API do manipulacji DOMem (przykład dokumentu HTML).

`document.title`

`document.head`

`document.body`

`document.createElement`

`document.addEventListener`

i wiele innych

Wyszukiwanie elementów na stronie

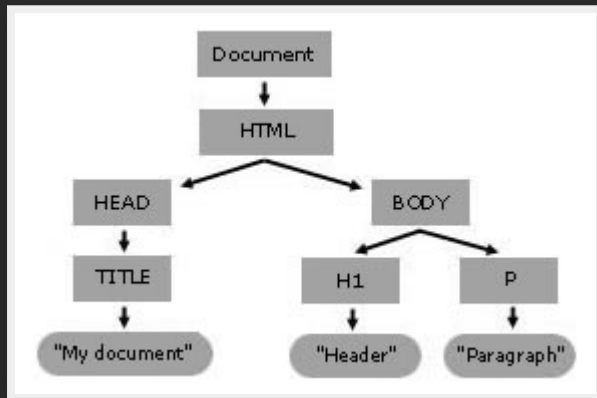
```
// get a single element
// accepts any valid CSS selector like '.some-class > a'
document.querySelector('#some-button')

// get all elements matching the given selector
document.querySelectorAll('.some-class')

// self-descriptive
document.getElementById('some-id')

// get all elements having ALL of the given classes
document.getElementsByClassName('class1 class2')
```

Relacje między elementami



- Element bezpośrednio zawierający inne elementy jest ich rodzicem (parent), a one jego dziećmi (children)
- Elementy mające tego samego rodzica są rodzeństwem (siblings)
- Rodzic rodzica itd. danego elementu jest jego przodkiem (ancestor)
- Dzieci dzieci itd. danego elementu to potomkowie (descendants)

Odwiedziny u rodziny.

Codepen

Manipulowanie elementem Codepen

Manipulowanie elementem cz. 2

Codepen

Manipulowanie stylami elementu

Codepen

Dodawanie elementów do strony

Codepen

Eventy

Pozwalają reagować na wydarzenia na stronie, tj. kliknięcie w przycisk, przescrollowanie strony itp.

```
// Składnia: element.addEventListener(eventType, function  
  
element.addEventListener('click', (e) => {  
    console.log(e.target);  
});
```

Event bubbling

Eventy idą od documentu w dół (faza capturing) aż do elementu w którym faktycznie wystąpił event (target), a potem znowu do góry (faza bubblingu)

Codepen

Stop propagation

Codepen

Prevent default

Powstrzymuje domyślną akcję odpalaną na danym evencie (np nawigację do nowej strony po kliknięciu na link, albo submit formularza)

Codepen

`e.target` i event delegation

Aby nie przypinać wielu listenerów do powtarzających się elementów możemy wykorzystać event delegation - przypięcie listenera do wspólnego rodzica i wykorzystanie `e.target` do określenia który element faktycznie wywołał event.

Codepen

Usuwanie przypiętych listenerów

Codepen

Further reading
javascript.info

Praca domowa

- Dodaj hamburger menu do strony dla ekranu mniejszego niż 360px - przycisk pokazujący/ukrywający nawigację na szerokości. Menu i przycisk otwierający może na razie pozostać nieostylowane, ostylujemy je po kolejnych zajęciach.
- Napisz funkcję, która po kliknięciu w przycisk "Kup bilet" w sekcji "Concerts" usunie przycisk i w jego miejscu wyświetli tekst "Have fun!"
- Napisz funkcję, która na evencie "submit" formularza kontaktowego wyloguje wpisane wartości do konsoli. (Podpowiedź: użyj funkcji `Object.fromEntries` w połączeniu z `FormData`)

Pytania?