

Programowanie w JavaScript



WSEI
#szkoła programowania



Event – wydarzenie w aplikacji (np. kliknięcie, wciśnięcie klawisza, itp.)

Callback – funkcja wywoływana po spełnieniu warunku wystąpienia. W przypadku eventów po wystąpieniu eventu.

```
<button id="test">Klikij mnie i przetestuj</button>

const button = document.getElementById('test');

button.addEventListener('click', function(){
  console.log('Kliknąłeś mnie!', this); // this wskazuje na kliknięty element
})

button.addEventListener('click', () => {
  console.log('Kliknąłeś mnie!', this); // this wskazuje na obiekt globalny
})

// addEventListener('nazwa_eventu', callback)
```

Nie musimy przekazywać funkcji anonimowej jako callbacka. Możemy zdefiniować wcześniej taką funkcję i użyć jej w kilku różnych eventach. Zaletą takiego rozwiązania jest fakt że funkcji eventCallback możemy używać jeszcze w innych przypadkach niż tylko eventy.

```
<button id="test">Klikij mnie i przetestuj</button>
```

```
const button = document.getElementById('test');
```

```
let eventCallback = function(){  
  console.log('tutaj kod do eventu');  
}
```

```
button.addEventListener('click', eventCallback)
```

```
button.addEventListener('mouseover', eventCallback)
```

Nie musimy przekazywać funkcji anonimowej jako callbacka. Możemy zdefiniować wcześniej taką funkcję i użyć jej w kilku różnych eventach. Zaletą takiego rozwiązania jest fakt że funkcji eventCallback możemy używać jeszcze w innych przypadkach niż tylko eventy. Jak widzicie na jednym elemencie możemy nasłuchiwać wielu zdarzeń.

```
<button id="test">Klikij mnie i przetestuj</button>
```

```
const button = document.getElementById('test');
```

```
let eventCallback = function(){  
  console.log('tutaj kod do eventu');  
}
```

```
button.addEventListener('click', eventCallback)
```

```
button.addEventListener('mouseover', eventCallback)
```

Czasem chcemy usunąć nasłuchiwanie zdarzeń. Służy do tego metoda `removeEventListener`.

```
<button id="test">Klikij mnie i przetestuj</button>
```

```
const button = document.getElementById('test');
```

```
let eventCallback = function(){  
  console.log('tutaj kod do eventu');  
}
```

```
button.addEventListener('click', eventCallback);
```

```
button.removeEventListener('click', eventCallback);
```

Najpopularniejsze eventy w JS:

Mousedown, mouseup, click, dblclick, mousemove, mouseover, mouseout

Keydown, keypress, keyup

Resize, scroll, Focus, blur, change, submit

DOMContentLoaded jest specjalnym eventem uruchamiającym nasz skrypt w momencie załadowania się drzewa DOM

```
document.addEventListener("DOMContentLoaded", function () {  
    console.log("drzewo DOM załadowane - możesz działać!");  
});
```

Każdy event posiada swój specjalny obiekt w którym przekazywane są informacje na jego temat. Najpopularniejsze używane informacje to:

Event.currentTarget – zwraca element na którym zostało wywołane zdarzenie

Event.target - zwraca element który spowodował wywołanie eventu

Event.timeStamp – zwraca czas wywołania eventu w ms

Event.type – zwraca typ (nazwę) eventu

Event.preventDefault() – anuluje domyślną akcję elementu

```
button.addEventListener('click', function(event){  
  console.log('Object event', event);  
})
```

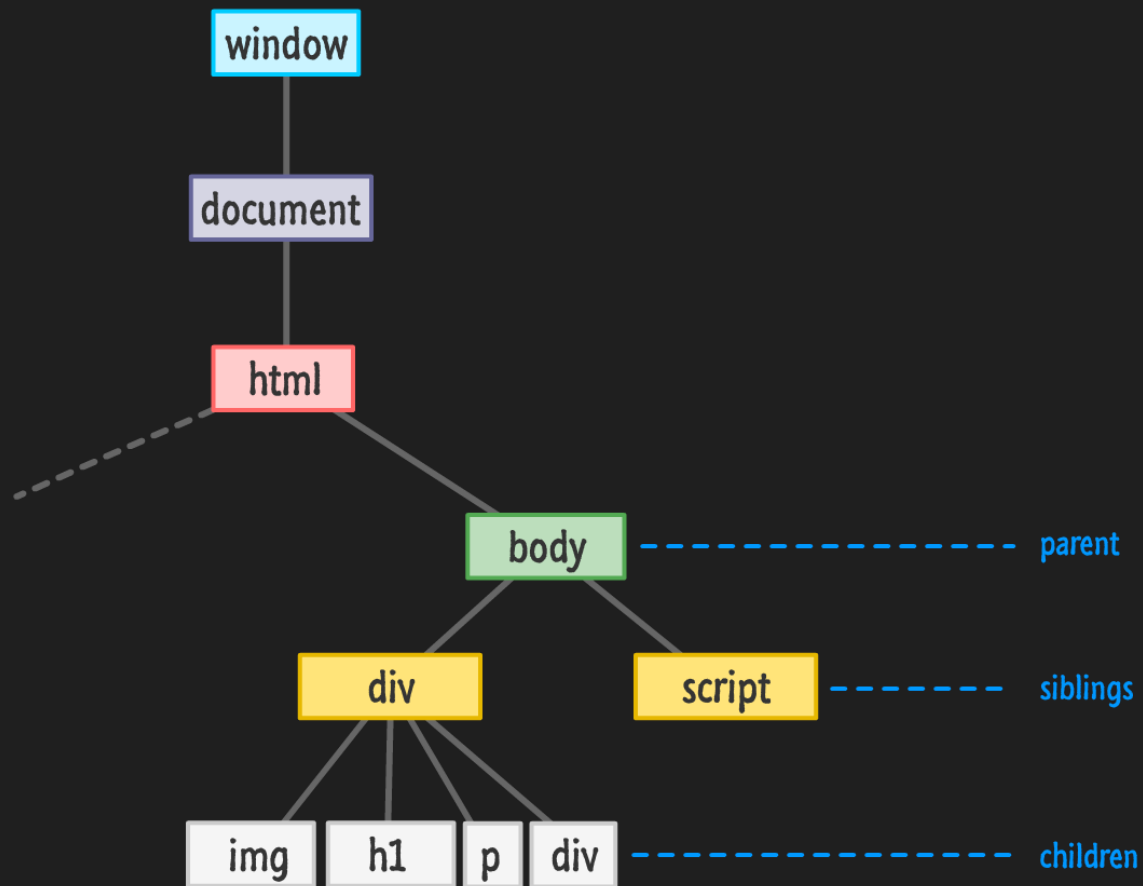



DOM – Poruszanie się po drzewie DOM

Po drzewie DOM możemy się dowolnie poruszać zaczynając od dowolnego elementu.

Rozróżniamy 3 relacje pomiędzy elementami:

1. Parent
2. Sibling
3. Child



Poruszanie się w górę:

```
<div id="foo">  
|   <div id="bar">Wewnętrzny div</div>  
</div>
```

```
const bar = document.querySelector('#bar');
```

```
const parent = bar.parentElement;
```

Poruszanie się w dół:

```
<div id="foo">  
  <div id="bar">Wewnętrzny div</div>  
  <div id="baz">Baz</div>  
  <div id="buz">Buz</div>  
</div>
```

```
const foo = document.querySelector('#foo');
```

```
const all = foo.children;  
const bar = foo.firstChild;  
const buz = foo.lastElementChild;
```

Poruszanie się na boki:

```
<div id="foo">  
  <div id="bar">Bar</div>  
  <div id="baz">Baz</div>  
  <div id="buz">Buz</div>  
</div>
```

```
const baz = document.querySelector('#baz');
```

```
const bar = baz.previousElementSibling;
```

```
const buz = baz.nextElementSibling;
```

