



DOM manipulacija

DOM manipulacija

Mijenjanje stilizacije

Povezana literatura

JavaScript: The Definitive Guide

- Part II. Client-side JavaScript, 16. Scripting CSS

JavaScript Patterns

- 9. DOM and Browser Patterns

<https://developer.mozilla.org/en-US/docs/Web/API/HTMLElement/style>

Inline stilovi

Atribut stila HTML elementa je njegov inline stil i poništava sve specifikacije stila u prije definiranom stylesheetu. Svojstvo stila ima **najviši prioritet** u CSS kaskadi kao inline deklaracija stila postavljena putem atributa stila.

```
document.getElementById(id).style.property = new style;  
element.style.position = "relative";
```

CSSStyleDeclaration

Svojstvo stila se razlikuje od drugih atributa HTML elementa po tome što njegova vrijednost nije niz, već **CSSStyleDeclaration** objekt, i ima posebna pravila imenovanja CSS svojstava.

`element.style.font-family = "Arial";` ❌

`element.style.fontFamily = "Arial";` ✓

`element.style.float = "left";` ❌

`element.style.cssFloat = "left";` ✓

Izračunati stilovi

Teže je doći do već predefiniranih vrijednosti stilova primijenjenih na elementu, ali možemo koristiti window metodu **getComputedStyle()**.

Ta nam metoda vraća stil koji je na kraju primijenjen na elementu i read-only je.

```
var element = document.getElementById(id);  
var styles = window.getComputedStyle(element, null);  
var style = getPropertyValue(styles, "font-size");
```

Mijenjanje CSS klasa

Ponekad želimo umjesto inline stila promijeniti CSS klasu elementa. Za to koristimo **className** i **classList** svojstva elemenata.

```
var element = document.getElementById(id);  
element.className = "containerElement containerElement--left";  
  
console.log(element.classList);  
element.classList.add("hide");  
element.classList.remove("hide");
```

Animiranje CSS-a

Zbog svoje dinamičke i interaktivne prirode, JavaScript se često koristi za animiranje CSS svojstava.

```
var elem = document.getElementById("cube");  
var pos = 0;  
var id = setInterval(frame, 10);  
  
function frame() {  
    pos++;  
    elem.style.left = pos + 'px';  
}
```


Mijenjanje stilizacije - vježba

Pratite upute u Upute.txt

Trajanje vježbe: 25min

DOM manipulacija

Događaji

Povezana literatura

JavaScript: The Definitive Guide

- Part II. Client-side JavaScript, 17. Handling events

JavaScript Patterns

- 9. DOM and Browser Patterns

Događaj

Kad god se nešto dogodi na našoj web stranici,
možemo koristiti JavaScript u reakciji na taj događaj.

Događaj

Pojmovi

event type	Tip događaja
event target	Objekt nad kojim se događaj dogodio ili s kojim je događaj povezan
event handler / listener	Funkcija koja reagira na događaj
event object	Objekt koji sadrži detalje o događaju (npr. type i target)
event propagation	Proces koji browser koristi da odredi event target
event capturing	Akcija koja se može poduzeti prije nego događaj dođe do mete

Tipovi događaja

Form	blur, change, focus, input, invalid, reset, submit, ...
Window	load, offline, line, popstate, resize, storage, ...
Mouse	click, mousedown, mouseup, mousewheel, wheel, ...
Key	keydown, keypress, keyup
HTML5	drag, dragstart, scroll, play, pause, progress, ...
Mobile	touchstart, touchend

addEventListener()

Ime event handlera ima prefix **on-**

3 su načina postavljanja:

- 1) **inline**
- 2) kao svojstvo na elementu
- 3) addEventListener metoda

```
<input type="button" onblur="blur();" value="Click" />
```

addEventListener()

Ime event handlera ima prefix **on-**

3 su načina postavljanja:

- 1) inline
- 2) **kao svojstvo na elementu**
- 3) addEventListener metoda

```
<input type="button" value="Click" />
```

```
var button = document.querySelector('button');  
button.onblur = function() {  
    ...  
}
```


addEventListener()

3 su načina postavljanja:

- 1) inline
- 2) kao svojstvo na elementu
- 3) **addEventListener metoda**

```
<input type="button" value="Click" />
```

```
var button = document.querySelector("button");  
function handleBlur() { ... }
```

```
button.addEventListener("blur", handleBlur);  
button.removeEventListener("blur", handleBlur);
```

Handler funkcija

Funkcija koja reagira na događaj uvijek dobiva objekt događaja kao parametar.

Također možemo i otkazati defaultnu reakciju na taj događaj, te spriječiti propagaciju događaja na druge elemente.

```
function handleBlur(event) {  
    var target = event.target;  
  
    if (target == element) {  
        event.preventDefault();  
        event.stopPropagation();  
    }  
    return false;  
}
```

Bubbling

