



DOM manipulacija

Dragi polaznici, ova prezentacija nije primarno namijenjena za učenje već služi kao **pomoćni materijal**. Kao takva, ne može zamijeniti predavanja, literaturu kao niti vašu (pro)aktivnost na nastavi i konzultiranje s predavačem u slučaju potencijalnih pitanja i/ili nejasnoća.

DOM manipulacija

Ključne riječi

HTML DOM, Window objekt, `getElementsByClassName`, `innerHTML`

Cjeline

Selektiranje DOM elemenata

Mijenjanje DOM strukture

Mijenjanje sadržaja i atributa

Mijenjanje stila

Događaji

Vježba

DOM manipulacija

Browser okolina

Web API-ji

Web nudi mnoštvo standardiziranih API-ja i sučelja (tipova objekata) koje biste mogli koristiti tijekom razvoja web aplikacije ili web mjesta.

API

File System, Storage, **DOM**, Web Animations, Canvas, Ambient Light Events, WebGL, Touch Events, Fetch ...

Sučelja

Navigator, Node, Animation, Performance, Bluetooth, CSS, Position, Cache, Comment, CustomEvent, SVGElement, History, ...

HTML DOM

Platforma i jezično neutralno sučelje koje omogućuje programima i skriptama dinamički pristup i ažuriranje sadržaja, strukture i stila dokumenta.

Definira:

- HTML elemente kao objekte
- svojstva svih HTML elemenata
- metode pristupa svim HTML elementima
- događaje za sve HTML elemente

JavaScript

HTML DOM-u može se pristupiti s JavaScript-om.

Elementi DOM-a su **objekti**, te imaju svojstva i metode.

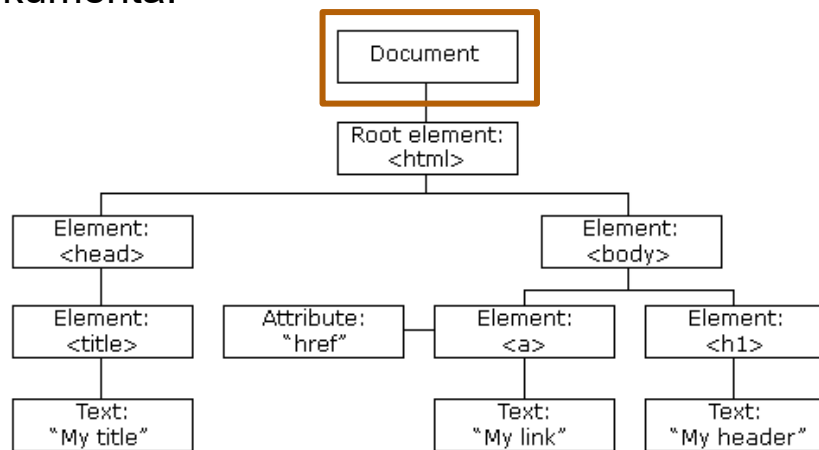
Zapamtite!

JavaScript sloj, zadužen za ponašanje stranice, trebao bi biti nenametljiv, što znači da ne bi trebao učiniti stranicu neupotrebljivom u nepodržanim preglednicima i ne bi trebao biti uvjet da možemo pristupiti sadržaju stranice.

Document objekt

Objekt dokumenta predstavlja našu web stranicu.

Ako želite pristupiti bilo kojem elementu HTML stranice, uvijek započinjemo s pristupom objektu dokumenta.



BOM i Window objekt

Browser **O**bject **M**odel omogućava JavaScriptu "razgovor" s web preglednikom. Objekt preko kojeg to ostvarujemo je globalni **window** objekt.

Window ima sljedeća svojstva, među ostalima:

document	Isti objekt kao u HTML DOM-u
innerHeight & innerWidth	Veličine trenutnog prozora
screen	Objekt koji drži informacije o korisničkom zaslonu
location	Objekt koji sadrži metode za manipulaciju url-a

Browser okolina - vježba

Pratite upute u Upute.txt.

Trajanje vježbe: 20min

DOM manipulacija

Mijenjanje DOM strukture

Primjer DOM-a

Na ovom satu koristit ćemo sljedeću DOM strukturu kao početnu točku:

```
<!DOCTYPE HTML>
<html>
<head>
  <title>Karlovac</title>
</head>
<body>
  <div class="description">
    Grad na <span>4 rijeke</span>
  </div>
  <footer id="footer">
    <ul>
      <li></li>
      <li></li>
    </ul>
  </footer>
</body>
</html>
```

Selektiranje DOM elementa

Za mijenjanje DOMa ili elementa u DOMu prvo ga moramo 'pronaći' u dokument strukturi i selektirati.

DOM definira brojne **načine odabira elemenata**, koristeći:

- **id elementa**
- **tag elementa (tip)**
- **CSS klasu ili drugi selektor**
- **atribut imena**

Selektiranje

```
document.getElementById("footer");
```

```
<!DOCTYPE HTML>
<html>
<head>
  <title>Karlovac</title>
</head>
<body>
  <div class="description">
    Grad na <span>4 rijeke</span>
  </div>
  <footer id="footer">
    <ul>
      <li></li>
      <li></li>
    </ul>
  </footer>
</body>
</html>
```

Selektiranje

```
document.getElementsByTagName("li");
```

U slučaju da postoji samo jedan element tog tipa, vraća se HTML tog elementa.

U slučaju da ih ima više, vraća se niz elemenata pod nazivom **NodeList**.

```
<!DOCTYPE HTML>
<html>
<head>
  <title>Karlovac</title>
</head>
<body>
  <div class="description">
    Grad na <span>4 rijeke</span>
  </div>
  <footer id="footer">
    <ul>
      <li></li>
      <li></li>
    </ul>
  </footer>
</body>
</html>
```


Selektiranje

```
document  
.getElementsByClassName("description");
```

```
document.querySelector(".description");
```

```
document  
.querySelectorAll(".description");
```

```
<!DOCTYPE HTML>  
<html>  
<head>  
  <title>Karlovac</title>  
</head>  
<body>  
  <div class="description">  
    Grad na <span>4 rijeke</span>  
  </div>  
  <footer id="footer">  
    <ul>  
      <li></li>  
      <li></li>  
    </ul>  
  </footer>  
</body>  
</html>
```

Kreiranje i dodavanje elementa

```
var f =  
document.querySelector("#footer");  
var d = document.createElement("div");  
  
f.appendChild(d);
```

```
<!DOCTYPE HTML>  
<html>  
<head>  
  <title>Karlovac</title>  
</head>  
<body>  
  <div class="description">  
    Grad na <span>4 rijeke</span>  
  </div>  
  <footer id="footer">  
    <ul>  
      <li></li>  
      <li></li>  
    </ul>  
    <div></div>  
  </footer>  
</body>  
</html>
```

Micanje elementa

```
var f =  
document.querySelector("#footer");  
  
f.parentNode.removeChild(f);
```

```
<!DOCTYPE HTML>  
<html>  
<head>  
  <title>Karlovac</title>  
</head>  
<body>  
  <div class="description">  
    Grad na <span>4 rijeke</span>  
  </div>  
  <del>footer id="footer"</del>  
    <del>ul</del>  
      <del>li</del></li>  
      <del>li</del></li>  
    </del>  
  <del>div</del>  
</del>  
</body>  
</html>
```

Mijenjanje DOM strukture - vježba

Pratite upute u Upute.txt

Trajanje vježbe: 25min

DOM manipulacija

Mijenjanje sadržaja i atributa

Sadržaj

Koji je sadržaj div.description-a?

- a) Grad na 4 rijeke
- b) Grad na 4 rijeke
- c) Tekstualni i element node-ovi

```
<!DOCTYPE HTML>
<html>
<head>
  <title>Karlovac</title>
</head>
<body>
  <div class="description">
    Grad na <span>4 rijeke</span>
  </div>
  <footer id="footer">
    <ul>
      <li></li>
      <li></li>
    </ul>
  </footer>
</body>
</html>
```

Sadržaj

Koji je sadržaj `div.description-a`?

- a) **Grad na `4 rijeke`**
- b) Grad na 4 rijeke
- c) Tekstualni i element node-ovi

```
var content = description.innerHTML;  
  
description.innerHTML =  
    'Ima <span>4 rijeke</span>';
```

Sadržaj

Koji je sadržaj div.description-a?

- a) Grad na 4 rijeke
- b) Grad na 4 rijeke**
- c) Tekstualni i element node-ovi

```
var content = description.textContent;
```

```
description.textContent = 'Ima 4 rijeke';  
description.innerText = 'Ima 4 rijeke';
```


Atributi

Vrijednosti atributa HTML elemenata dostupne su kao **svojstva HTMLElement** objekata koji predstavljaju te elemente.

```
var image =  
document.querySelector("image");
```

```
image.src = "ipsum.png";
```

```
<!DOCTYPE HTML>  
<html>  
<head>  
  <title>Karlovac</title>  
</head>  
<body>  
  <image src="lorem.png" width="30"/>  
  <div class="description">  
    Grad na <span>4 rijeke</span>  
  </div>  
  <footer id="footer">  
    <ul>  
      <li></li>  
      <li></li>  
    </ul>  
  </footer>  
</body>  
</html>
```

Ne-HTML i data Atributi

```
var image =  
document.querySelector("image");  
  
image.getAttribute("width");  
image.setAttribute("height") = 40;  
image.removeAttribute("height");  
  
var ul = document.querySelector("ul");  
  
ul.dataset.index = 2;
```

```
<!DOCTYPE HTML>  
<html>  
<head>  
  <title>Karlovac</title>  
</head>  
<body>  
  <image src="lorem.png" width="30"/>  
  <div class="description">  
    Grad na <span>4 rijeke</span>  
  </div>  
  <footer id="footer">  
    <ul data-index="1">  
      <li></li>  
      <li></li>  
    </ul>  
  </footer>  
</body>  
</html>
```

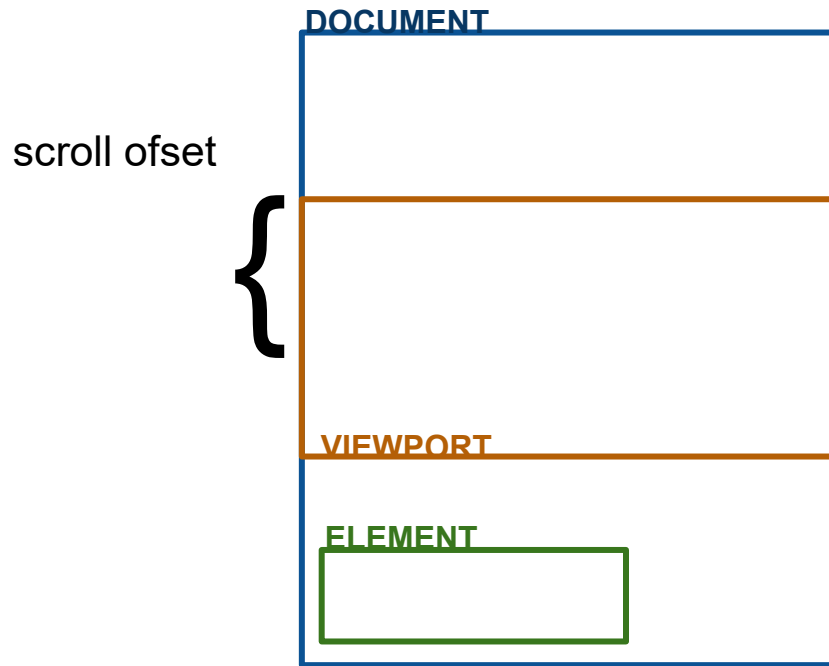
Geometrija elemenata

Svaki HTML element ima širinu i visinu te poziciju izraženu u pikselima.

Koordinate x i y mogu biti u odnosu na:

- gornji lijevi kut **dokumenta**
- gornji lijevi kut **viewporta**

Atributi elementa kojima možemo pristupiti: **pageOffsetX**, **pageOffsetY**, **scrollLeft**, **scrollTop**



Mijenjanje sadržaja i atributa - vježba

Pratite upute u Upute.txt

Trajanje vježbe: 30min

DOM manipulacija

Mijenjanje stilizacije

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

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

