



# JavaScript

Četvrti dio



# Pregled

- DOM
  - Uvod
  - HTML DOM
    - Čvorovi
    - Odnos među čvorovima
    - Document
    - Selekcija i modifikacija čvorova
    - Izvlačenje sadržaja elemenata
    - Obilazak čvorova
    - Kreiranje čvorova
    - Brisanje čvorova i klase
    - Ostala svojstva čvorova
    - Razlika između HTMLCollection i NodeList

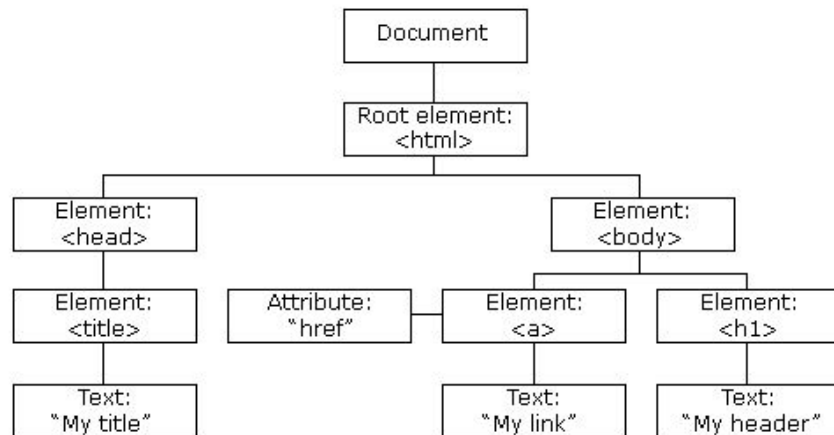


# Obnavljanje

- Šta smo rekli za Scope, šta važi za **let**, **const** i **var**
- Koji prednosti imaju imenovane funkcije u odnosu na anonimne
- Zašto se koriste IIFE
- Hoisting
- Koja je razlika između mutable i immutable tipova podata
- Koja 4 metoda za dodavanje i brisanje elemenata niza se najčešće koriste

# DOM, uvod

- Pri učitavanju stranice, pretraživač na stranici kreira **DOM** (document object model)
- HTML DOM model se kreira kao drvo objekata





## DOM, uvod - 2

- DOM definiše standard za pristupanje dokumentima
- Definicija:
  - *"The W3C Document Object Model (DOM) is a platform and language-neutral interface that allows programs and scripts to dynamically access and update the content, structure, and style of a document."*
- DOM standard se dijeli na 3 dijela i to:
  - Core DOM - standard modela za sve tipove dokumenata
  - XML DOM - standard modela za XML dokumente
  - HTML DOM - standard modela za HTML dokumente (mi ćemo za sad da se fokusiramo samo na ovu cjelinu)
- **HTML DOM** je standard za objektni model i programski interfejs za HTML, a definiše sledeće:
  - HTML elemente kao objekte
  - Svojstva svih HTML elementa
  - Metode za pristup svim HTML elementima
  - Evente/Događaje za sve HTML elemente



## DOM, uvod - 3

- HTML DOM je API za HTML, jer sadrži operacije: get, update, create, delete HTML element
- Šta sve možemo sa ovim modelom
  - JSom je moguće izmijeniti sve HTML elemente na stranici
  - JSom je moguće mijenjati sve HTML atribute na stranici
  - JSom je moguće mijenjati sve CSS stilove na stranici
  - JSom je moguće ukloniti postojeće HTML elemente i atribute
  - JSom je moguće dodati nove HTML elemente i atribute
  - JSom je moguće dodavanje događaja na HTML elemente
  - JSom je moguće definisanje reakcije na događaje

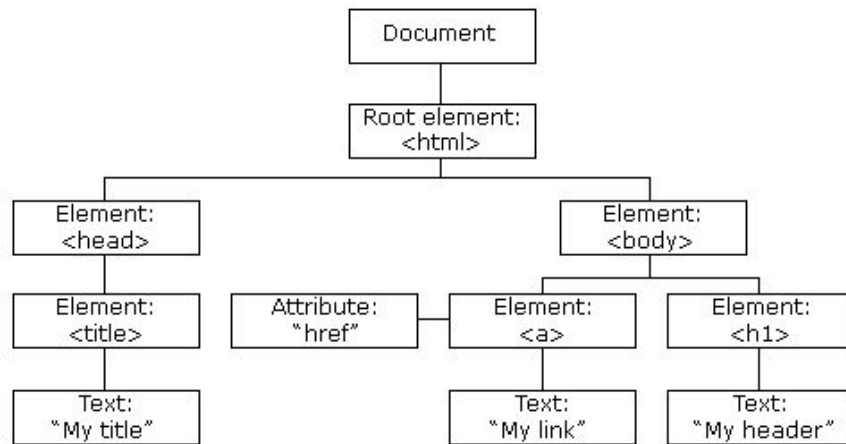


# HTML DOM

- HTML DOM metode su akcije koje mogu biti obavljene nad HTML elementima
- HTML DOM svojstva su vrijednosti HTML elemenata koje mogu da se setuju ili mijenjaju (mijenjanje sadržaja HTML elementa)

# HTML DOM, čvorovi

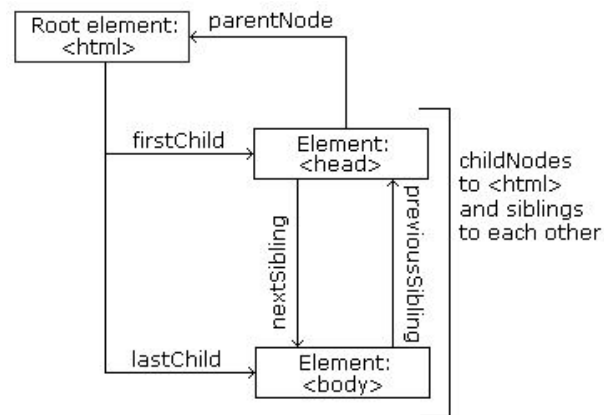
- Kod HTML DOMa moguć je obilazak čvorova jer su čvorovi (HTML elementi) povezani kao elementi drveta
- Sve u HTML dokumentu je čvor
  - Cijeli dokument je **document node**
  - Svaki HTML element je **element node**
  - Tekst unutar HTML elemenata je **text node**
  - Svi komentar je **comment node**
  - Svaki HTML atribut je atribut node (zastarjelo)





# HTML DOM, odnos među čvorovima

- Čvorovi u drvetu imaju hijerarhijski odnos međusobno
- Koriste se termini **parent**, **child** i **sibling** da bi se opisao odnos među čvorovima
  - U DOM drvetu, top čvor se zove **root**, tj. korijen drveta
  - Svaki čvor u drvetu ima bar jedan **parent** node, osim root čvora (nema parenta)
  - Čvor može da sadrži više **child** čvorova, children nodes
  - **Siblings** su čvorovi unutar istog parent čvora





# HTML DOM, odnos među čvorovima -2

- Primjer:

- <html> je root node
- <html> nema parent čvor
- <html> je parent <head> i <body> čvorova
- <head> je prvi child <html> čvora
- <body> je zadnji child <html> elementa
- <head> ima jedan child node <title>
- <title> ima one child node "DOM Tutorial"
- <body> ima dva child noda <h1> i <p>
- <h1> ima jedan child node "DOM Lesson one"
- <p> ima jedan child node "Hello world!"
- <h1> i <p> su siblings

## HTML

```
<html>
  <head>
    <title>DOM Tutorial</title>
  </head>
  <body>
    <h1>DOM Lesson one</h1>
    <p>Hello world!</p>
  </body>
</html>
```



# HTML DOM, document

- HTML DOM document objekat je “vlasnik” svih ostalih objekata na stranici
- Document objekat predstavlja web stranicu
- Ako želite da pristupite bilo kom elementu na stranici, uvijek krećete od pristupa document objektu
- U nastavku ćemo vidjeti kako možemo da iskoristimo document objekat za pristup i manipulaciju HTML elemenata
- Prvi dio primjera (code - 5)



# HTML DOM, document properties

<code>document.anchors</code>	Returns all <code>&lt;a&gt;</code> elements that have a name attribute
<code>document.applets</code>	Returns all <code>&lt;applet&gt;</code> elements (Deprecated in HTML5)
<code>document.baseURI</code>	Returns the absolute base URI of the document
<code>document.body</code>	Returns the <code>&lt;body&gt;</code> element
<code>document.cookie</code>	Returns the document's cookie
<code>document.doctype</code>	Returns the document's doctype
<code>document.documentElement</code>	Returns the <code>&lt;html&gt;</code> element
<code>document.documentMode</code>	Returns the mode used by the browser
<code>document.documentURI</code>	Returns the URI of the document
<code>document.domain</code>	Returns the domain name of the document server
<code>document.domConfig</code>	Obsolete. Returns the DOM configuration
<code>document.embeds</code>	Returns all <code>&lt;embed&gt;</code> elements
<code>document.forms</code>	Returns all <code>&lt;form&gt;</code> elements



## HTML DOM, document properties - 2

<code>document.head</code>	Returns the <head> element
<code>document.images</code>	Returns all <img> elements
<code>document.implementation</code>	Returns the DOM implementation
<code>document.inputEncoding</code>	Returns the document's encoding (character set)
<code>document.lastModified</code>	Returns the date and time the document was updated
<code>document.links</code>	Returns all <area> and <a> elements that have a href attribute
<code>document.readyState</code>	Returns the (loading) status of the document
<code>document.referrer</code>	Returns the URI of the referrer (the linking document)
<code>document.scripts</code>	Returns all <script> elements
<code>document.strictErrorChecking</code>	Returns if error checking is enforced
<code>document.title</code>	Returns the <title> element
<code>document.URL</code>	Returns the complete URL of the document



# HTML DOM, selekcija i modifikacija čvorova

- HTML elemente moguće je selektovati na više načina
  - `document.getElementById("value")` - vraće HTML element iz DOMa čiji je id = "value"
  - `document.getElementsByClassName("value")` - vraće sve HTML elemente čija je vrijednost atributa class="value"
  - `document.getElementsByTagName("value")` - vraće sve elemente čiji je tag value
    - Sjećate se šta je tag?
  - `document.querySelector("selector")` - vraća prvi HTML element koji zadovoljava upit (selektor)
    - selector - definiše se kao kod CSSa
  - `document.querySelectorAll("selector")` - vraća sve HTML elemente koji zadovoljavaju upit
- HTML elemente moguće je mijenjati (sadržaj ili attribute)
  - `document.querySelector("selector").innerHTML = "HTML code"`
    - Mijenja se sav HTML unutar selektovanog elementa
  - `document.querySelector("selector").attributeName = "attribute value"`
    - Mijenja se vrijednost atributa selektovanog HTML elementa
    - Može i `document.querySelector("selector").setAttribute("attribute_name", "attribute_value")`
  - `document.querySelector("selector").style.property = "property value"`
    - Mijenja stil selektovanog HTML elementa
  - Šta ako imate niz `document.getElementsByClassName("value")` ?



# HTML DOM, izvlačenje sadržaja elemenata

- Moguće je izvlačenje sadržaja čvora:
  - **`document.getElementById("value").textContent`**
    - Izvlači vidljivi tekst iz elementa
  - **`document.getElementById("value").innerText`**
    - Izvlači sav tekst iz čvora bez obzira na vrijednost display svojstva kod CSSa
  - **`document.getElementById("value").nodeValue`**
    - Vraće null ako je element node ili document node
    - Vraće sadržaj ako je tekst node
    - Vraće sadržaj ako je comment node
- Nastavak primjera (code - 5, drugi dio)



# HTML DOM, obilazak čvorova

- Možemo koristiti sledeća svojstva čvorova za obilazak DOMa
  - `node.parentNode` (vraće parent node trenutno selektovanog elementa)
    - `parentElement` slično kao `parentNode`
    - da vidite šta je razlika pročitati [članak](#)
  - `node.children[i]` (vraće i-ti netekstualni child node među čvorovima)
    - `node.childNodes[i]` - slično kao `children` ali vraće i tekst čvorove
  - `node.firstElementChild` (vraće prvi netekstualni child čvor)
    - `node.firstChild` (vraće prvi child node)
  - `node.lastElementChild` (vraće zadnji netekstualni child čvor)
    - `node.lastChild` (vraće zadnji child node)
  - `node.nextElementSibling` (vraće sledeći netekstualni sibling čvor)
    - `node.nextSibling` (vraće sledeći sibling čvor)
  - `node.previousElementSibling` (vraće prethodni netekstualni sibling čvor)
    - `node.previousSibling` (vraće prethodni sibling čvor)
- Nastavak primjera (code-5, treći dio)





# HTML DOM, kreiranje čvorova

- Kreiranje elemenata
  - **document.createElement("element\_name")**
    - Kreiranje praznog HTML elementa "element\_name"
    - Npr. za kreiranje praznog div-a, **document.createElement("div")**
  - **document.createTextNode("tekst")**
    - Kreiranje tekstualnog cvora
- Za dodavanje HTML elementa u drugi HTML element
  - **parent.appendChild(child)** - dodavanje na kraj
  - **parent.insertBefore(child, beforeWhichNode)**
- Nastavak primjera (code-5, četvrti dio)



# HTML DOM, brisanje čvora i klase

- Brisanje čvorova
  - `parent.removeChild(child)`
- Rad sa klasama HTML elemenata
  - `element.classList.remove("ime klase")` - brisanje klase
  - `element.classList.add("ime klase")` - dodavanje klase
  - `element.classList.contains("ime klase")` - provjera da li element ima "ime klase"
  - `element.classList` - sve klase zakačene na elementu
- Nastavak primjera (code-5, peti dio)



# HTML DOM, ostala svojstva čvorova

- Još svojstva čvorova
  - **node.nodeType** - vraće tip čvora (da li je text, element, comment ili attribute node)
  - **node.nodeName** - naziv taga elementa
  - Zadatak, walkTree
- Tipovi čvorova

Node	Type	Example
ELEMENT_NODE	1	<h1 class="heading">W3Schools</h1>
ATTRIBUTE_NODE	2	class = "heading" (deprecated)
TEXT_NODE	3	W3Schools
COMMENT_NODE	8	<!-- This is a comment -->
DOCUMENT_NODE	9	The HTML document itself (the parent of <html>)
DOCUMENT_TYPE_NODE	10	<!Doctype html>



# HTML DOM, razlika između HTML Collection i NodeList

- HTML Collection je kolekcija HTML elemenata
- NodeList je kolekcija document čvorova
- Mala razlika između HTML Collection i NodeList
  - HTML Collection i NodeList se ponašaju kao niz objekata
  - Imaju property length
  - Njihovim elementima može se pristupiti pomoću indeksa
  - HTML Collection elementima može se pristupiti po nazivu, id-u ili indeksu
  - Elementima NodeListe može se samo pristupiti po indeksu
  - Samo NodeList može da sadrži attribute i text čvorove
- HTML Collection i NodeList nisu standardni nizovi
  - Možete da iterirate kao kroz niz ali ne možete koristiti metode nizova kao što su valueOf(), push(), pop(), join()



Pitanja