



ftninformatika

Java Web Development

Modul 2

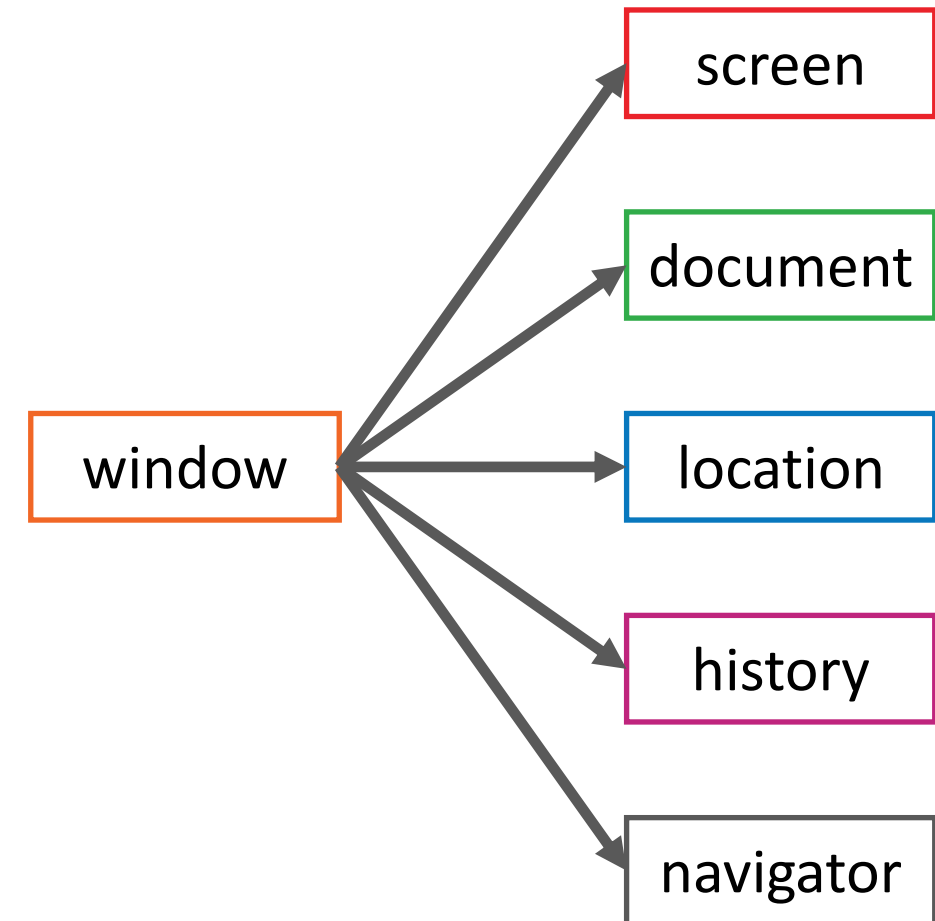
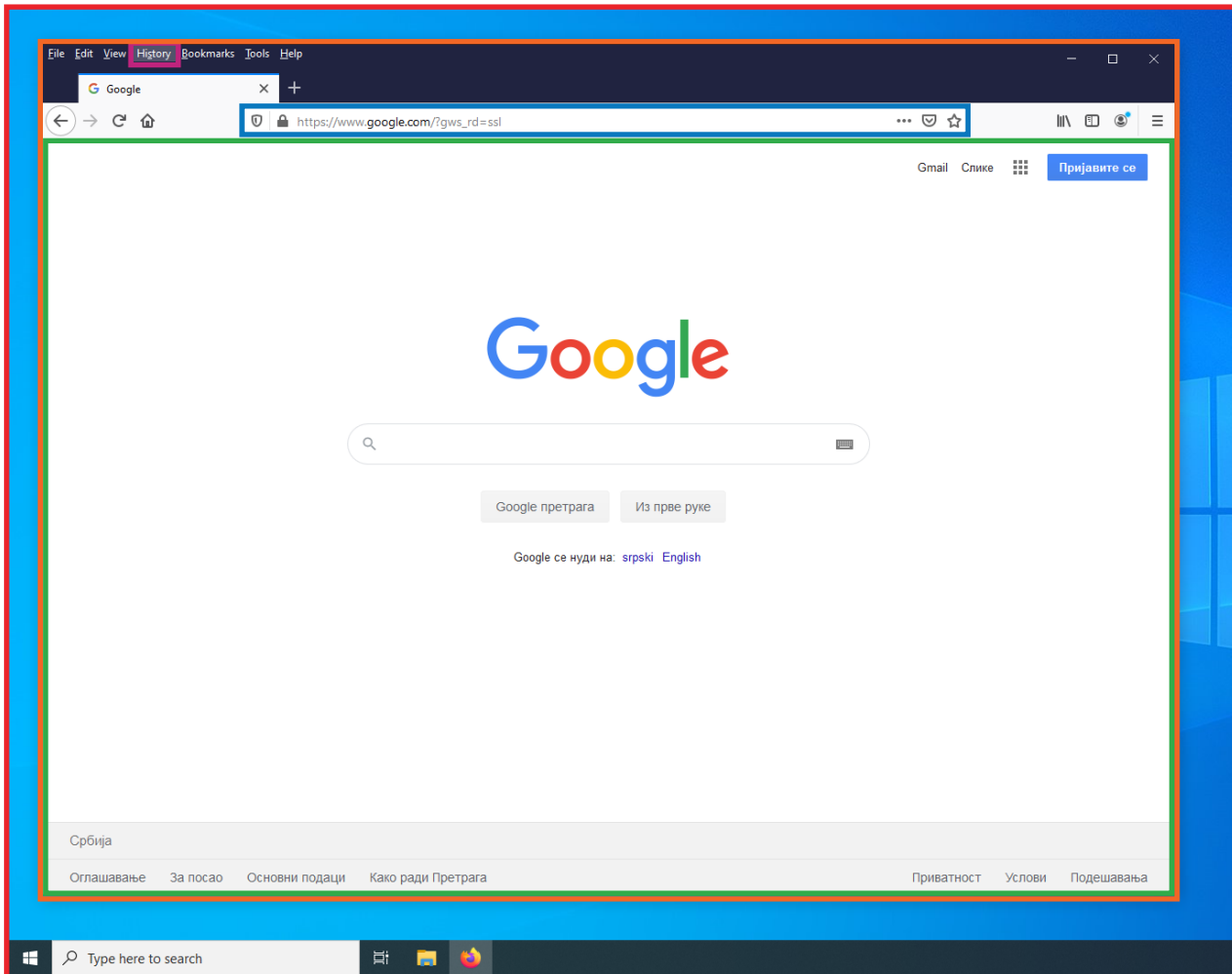
Termin 10

Sadržaj

1. BOM (*Browser Object Model*)
2. DOM (*Document Object Model*)

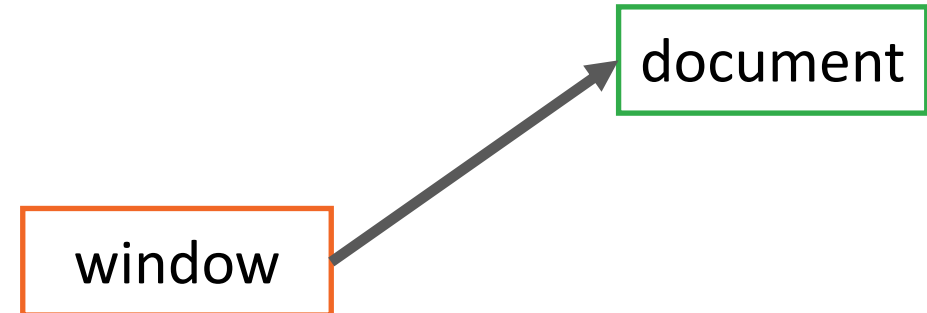
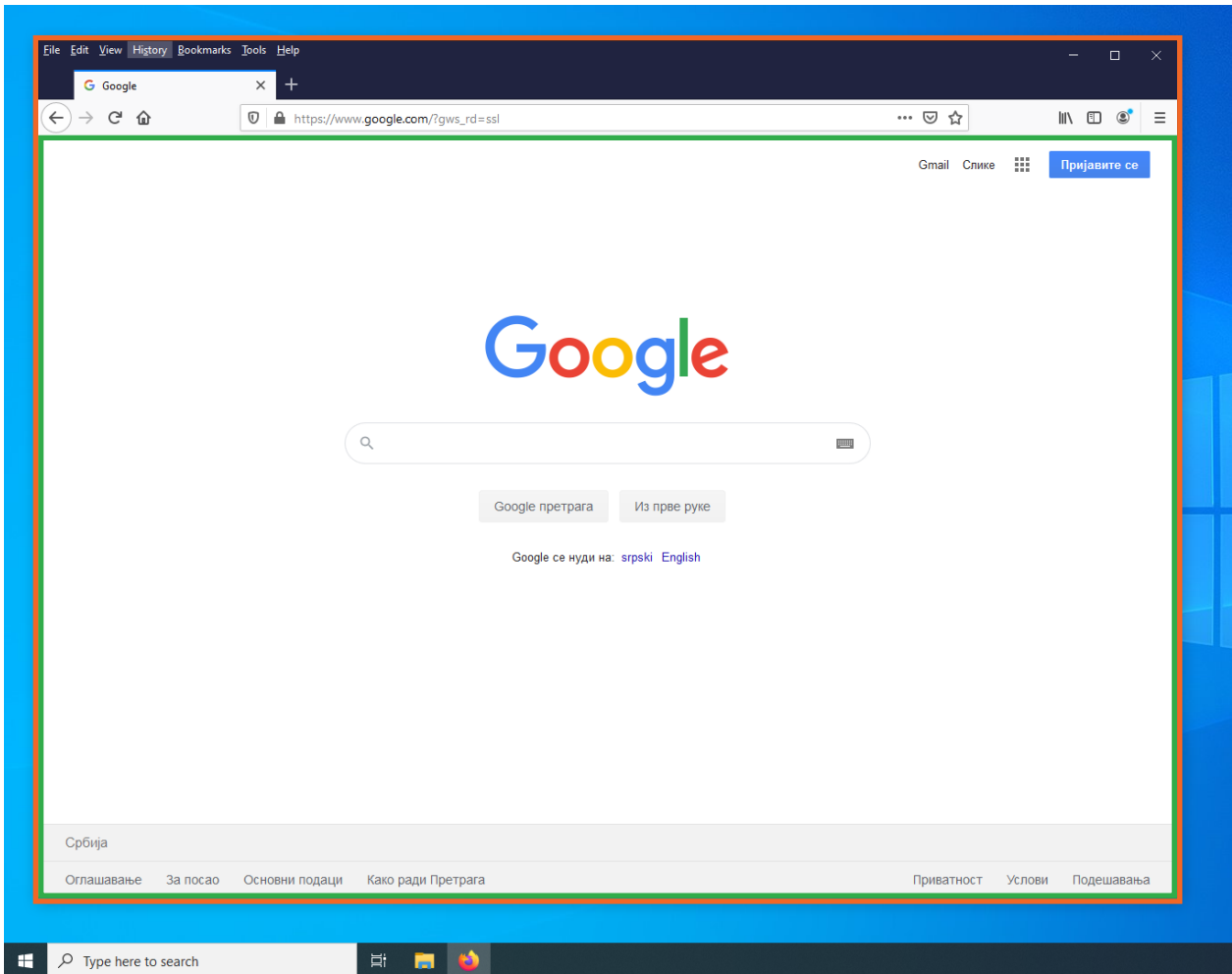
BOM (*Browser Object Model*)

Objektni model prozora *web browser-a* koji omogućuje programsko rukovanje njegovim elementima.



BOM (*Browser Object Model*)

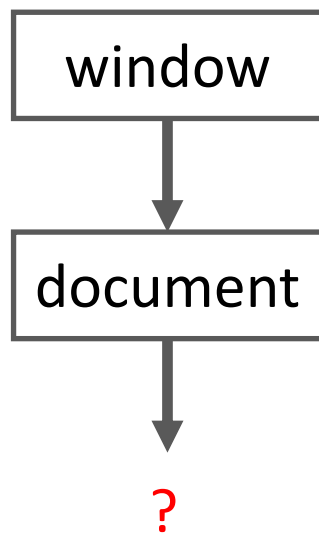
Objektni model prozora *web browser-a* koji omogućuje programsko rukovanje njegovim elementima.



DOM (*Document Object Model*)

Objektni model HTML stranice uz pomoć kog je moguća manipulacija njenim elementima.

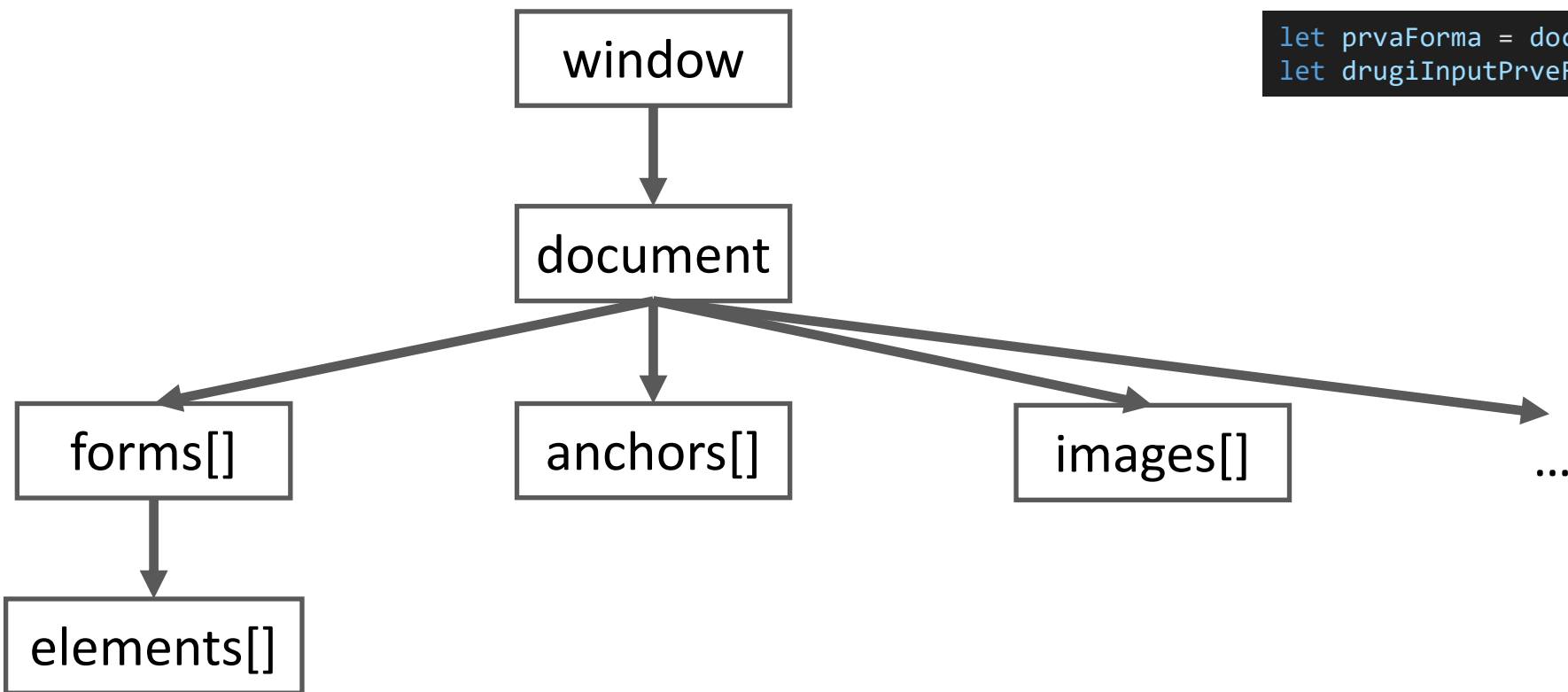
DOM specifikacija je evoluirala kroz više generacija (*Level 0, Level 1, ...*).



DOM (*Document Object Model*) Level 0

DOM Level 0 specifikacija opisuje predefinisane nizove elemenata koji se mogu ili ne moraju javiti na HTML stranici.

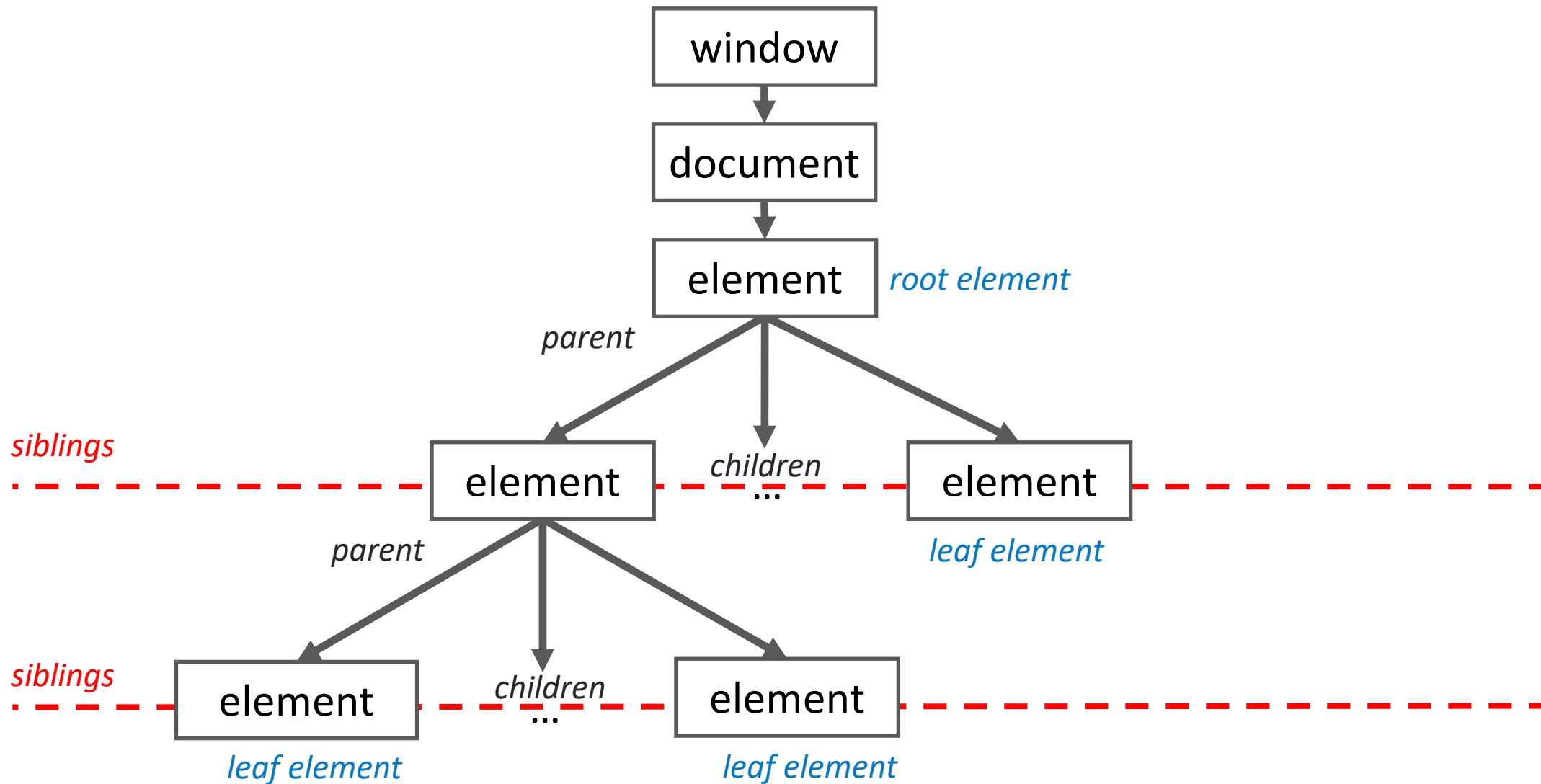
Nizovi uvek postoje, makar bili i prazni.



```
let prvaForma = document.forms[0];  
let drugiInputPrveForme = document.forms[0].elements[1];
```

DOM (*Document Object Model*) Level 1 - 4

DOM Level 1 - 4 specifikacije opisuju HTML dokument kao stablo elemenata.



DOM (*Document Object Model*) Level 1 - 4

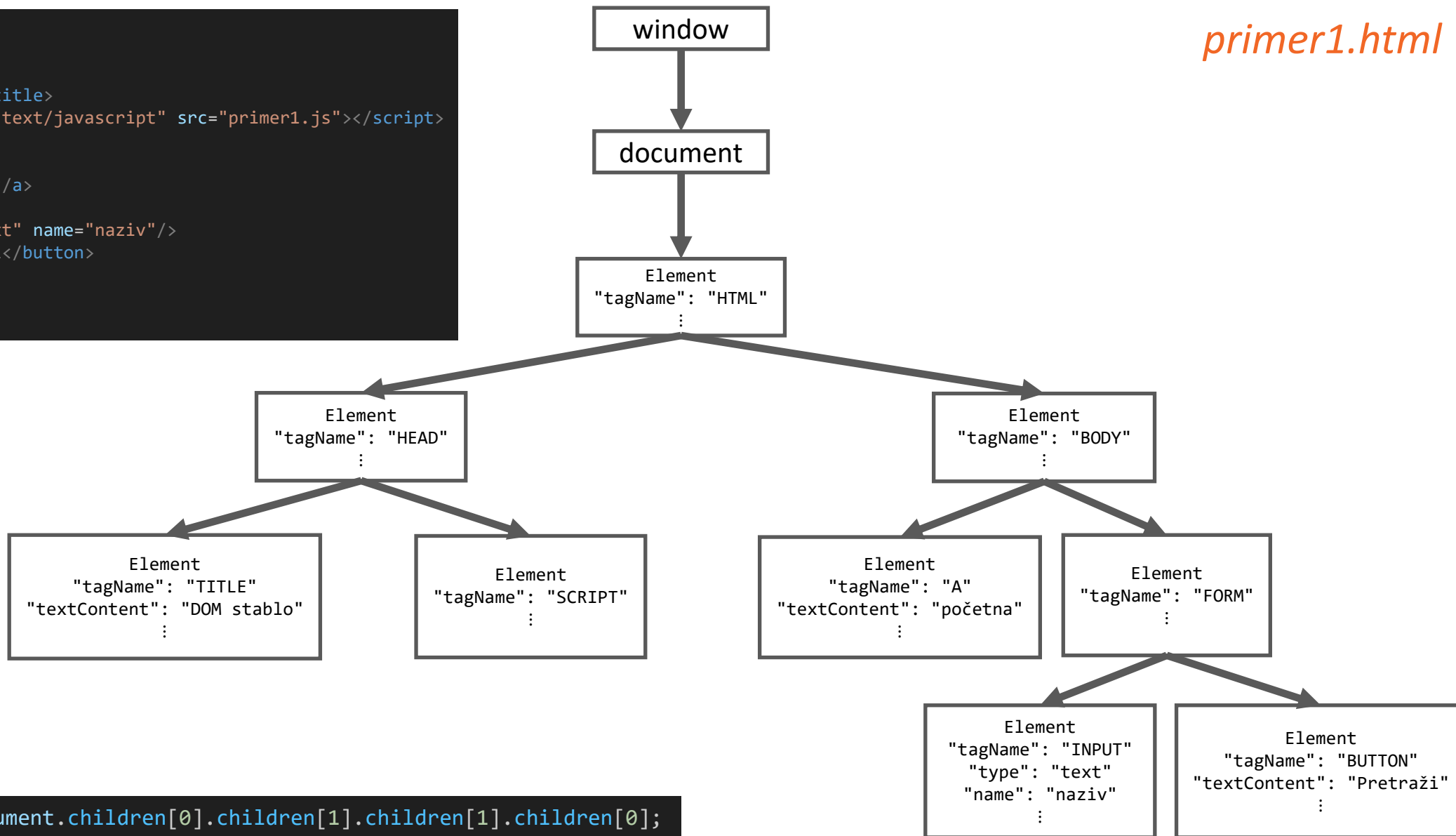
DOM Level 1 - 4 specifikacije opisuju HTML dokument kao stablo elemenata:

- svaki element stabla je opisan objektom tipa element
- element na vrhu stabla se zove korenski element (*root element*)
- svaki element, osim korenskog, ima referencu na jedan roditeljski element (*parent*)
- svaki element ima niz referenci na elemente potomke (*children*); niz može biti prazan i tada se element naziva leaf element
- elementi potomci istog roditeljskog čvora se međusobno nazivaju *siblings*

DOM (*Document Object Model*) Level 1 - 4

```
<!DOCTYPE html>
<html>
<head>
  <title>DOM stablo</title>
  <script defer type="text/javascript" src="primer1.js"></script>
</head>
<body>
  <a href="/">početna</a>
  <form method="get">
    <input type="text" name="naziv"/>
    <button>Pretraži</button>
  </form>
</body>
</html>
```

primer1.html



```
let inputNaziv = document.children[0].children[1].children[1].children[0];
```

DOM (*Document Object Model*) Level 1 - 4

Document

Modeluje HTML dokument.

	Naziv	Upotreba
atributi	<i>head, body</i>	• referenca na čvor <i>head</i> , odnosno <i>body</i> elementa
	<i>forms, anchors, images</i>	• reference na nizove definisane u DOM Level 0
	<i>title</i>	• naslov dokumenta
	<i>URL</i>	• URL dokumenta
	i mnogi drugi...	
metode	<i>write(...)</i>	• ako se pozove za vreme učitavanja DOM stabla, parsira prosleđeni HTML kod, od njega kreira elemente i dodaje ih kraj DOM stabla
	<i>getElementById(id)</i> <i>getElementsByName(name)</i> <i>getElementsByClassName(name)</i> <i>getElementsByTagName(name)</i>	• pronalazi element po vrednosti <i>id</i> atributa • vraća niz elementa pronađenih po vrednosti <i>name</i> atributa • vraća niz elementa pronađenih po vrednosti <i>class</i> atributa • vraća niz elementa pronađenih po nazivu <i>tag</i> -a
	<i>createElement(...)</i>	• kreira novi prazan element i vraća referencu na njega
	i mnoge druge...	

DOM (*Document Object Model*) Level 1 - 4

Element

primer2a.html

Modeluje HTML element.

	Naziv	Upotreba
atributi	<i>innerHTML</i> <i>textContent</i>	<ul style="list-style-type: none">• kompletan HTML kod podstabla elementa• tekstualni sadržaj elementa
	<i>tagName</i> <i>id</i> <i>className</i>	<ul style="list-style-type: none">• naziv elementa• vrednost <i>id</i> atributa (ako se navede)• vrednost <i>class</i> atributa (ako se navede)
	<i>style</i>	<ul style="list-style-type: none">• referenca na <i>style</i> objekat
	<i>parentElement</i> <i>children</i> <i>firstElementChild</i> <i>lastElementChild</i> <i>previousElementSibling</i> <i>nextElementSibling</i>	<ul style="list-style-type: none">• roditeljski element• niz elemenata potomaka• prvi element potomak• poslednji element potomak• prethodni element na istom nivou stabla• sledeći element na istom nivou stabla
	<i>disabled</i>	<ul style="list-style-type: none">• vrednost <i>true</i> blokira element
	i mnogi drugi...	

DOM (*Document Object Model*) Level 1 - 4

Element

Modeluje HTML element.

	Naziv	Upotreba
metode	<i>getAttribute(attributeName)</i> <i>setAttribute(attributeName, attributeValue)</i> <i>removeAttribute(attributeName)</i> <i>hasAttributes()</i>	<ul style="list-style-type: none">• čita vrednost atributa pozivajućeg elementa• postavlja novu vrednost atributa pozivajućeg elementa• uklanja atribut pozivajućeg elementa• vraća <i>true</i> ako pozivajući element u svom podstablu ima attribute
	<i>querySelector(...)</i> <i>querySelectorAll(...)</i> <i>closest(...)</i>	<ul style="list-style-type: none">• vraća element iz podstabla pozivajućeg elementa koji odgovara prosleđenom CSS selektoru (funkcioniše <u>samo za čvorove koji su nastali pri učitavanju stranice</u>, ali ne i za one koji su naknadno dodati)• vraća niz elemenata iz podstabla pozivajućeg elementa koji odgovaraju prosleđenom CSS selektoru (funkcioniše <u>samo za čvorove koji su nastali pri učitavanju stranice</u>, ali ne i za one koji su naknadno dodati)• vraća najbliži element u DOM stablu koji odgovara prosleđenom CSS selektoru
	<i>prepend(...)</i> <i>append(...)</i> <i>before(...)</i> <i>after(...)</i> <i>remove()</i> <i>insertAdjacentHTML(pozicija, html)</i>	<ul style="list-style-type: none">• dodaje čvorove na početak liste čvorova potomaka pozivajućeg elementa• dodaje čvorove na kraj liste čvorova potomaka pozivajućeg elementa• umeće čvorove ispred pozivajućeg elementa u podstablo zajedničkog roditeljskog elementa• umeće čvorove iza pozivajućeg elementa u podstablo zajedničkog roditeljskog elementa• uklanja pozivajući element• parsira prosleđeni HTML kod, od njega kreira elemente i umeće ih na prosleđenu poziciju u DOM stablo u odnosu na pozivajući element
	i mnoge druge...	

DOM (*Document Object Model*) Level 1 - 4

CSS selektori (podsetnik)

CSS selektor	Primer	Značenje
*	"*"	svi elementi na stranici
#id	"#neki_id"	element sa atributom id, čija je vrednost navedena iza znaka #
.class	".neka_klasa"	svi elementi koji pripadaju klasi čiji je naziv naveden iza znaka .
.element	"p"	svi elementi čiji <i>tag</i> ima navedeni naziv
s1, s2, ...	"p, th, .neka_klasa, #neki_id"	višestruka selekcija; unija rezultata svih selektora, pojedinačni selektori se odvajaju znakom ,
parent child	"li a"	traženi potomci navedenog roditelja
:first-child	"tr:first-child"	element pogođen selekcijom koji je prvi potomak <u>svog</u> roditelja
:last-child	"tr:last-child"	element pogođen selekcijom koji je poslednji potomak <u>svog</u> roditelja

DOM (*Document Object Model*) Level 1 - 4

CSS selektori (podsetnik)

primer2b.html

CSS selektor	Primer	Značenje
<code>:first-of-type</code>	<code>button:first-of-type</code>	prva pojava elementa u skupu pretrage
<code>:last-of-type</code>	<code>button:last-of-type</code>	poslednja pojava elementa u skupu pretrage
<code>:not(s)</code>	<code>"input:not(:checked)"</code>	svi elementi pogođeni selekcijom, na koje se ne odnosi selekcija u zagradi
<code>:odd</code>	<code>"tr:odd"</code>	svi neparni elementi pogođeni selekcijom
<code>:even</code>	<code>"tr:even"</code>	svi parni elementi pogođeni selekcijom
<code>[attribute]</code>	<code>"input[checked]"</code>	svi elementi koji poseduju navedeni atribut
<code>[attribute=value]</code>	<code>"input[type=text]"</code>	svi elementi koji poseduju navedenu vrednost atributa
i mnogi drugi...		

DOM (*Document Object Model*) Level 1 - 4

Element

Modeluje HTML element.

atribut (<i>handler</i> događaja)	Poziva se kada...	Najčešća upotreba
onload	... se stranica učitava, ako se definiše za <i>window</i> objekat ... se element učitava, ako se definiše za bilo koji element	kada se stranica ili neki element dugo učitava, a neka procedura zahteva da element bude učitav
onclick	... korisnik klikne na element	validacija unosa forme
onsubmit	... korisnik <i>submit</i> -uje formu	
onmouseover	... korisnik pređe mišem preko elementa	promena izgleda elementa, ispis pojašnjenja, prikaz dodatnih opcija
onfocus	... element poprimi fokus	
onblur	... element izgubi fokus	validacija unosa za celo polje
onchange	... se promeni vrednost u <i>input</i> ili <i>select</i> elementu	
onkeydown	... korisnik pritisne taster dok je element u fokusu	validacija unosa za svaki karakter
onkeyup	... korisnik otpusti taster dok je element u fokusu	
i mnogi drugi...		

DOM (*Document Object Model*) Level 1 - 4

Element

primer3.html, primer4.html

Funkcija koja se upiše u atribut elementa *on<događaj>* postaje obrađivač tog događaja (*event handler*) za taj element.

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="UTF-8">
  <title>Događaji</title>
  <script defer src="primer3.js" type="text/javascript"></script>
</head>
<body>
  <p>pritisnite bilo koji taster</p>
  <p style="background: gray; width: 25%">
    pređite mišem preko ovog pasusa
  </p>
  <p><button>kliknite na ovo dugme</button></p>
  <form>
    <input type="text" value="kliknite na ovo polje"/>
    <input type="submit" value="kliknite na ovo dugme"/>
  </form>
</body>
</html>
```

```
// registracija handler-a
window.onload = function() { globalni događaj
  alert("onload");
};

document.onkeyup = function(event) { // event parametar je opisnik događaja globalni događaj
  console.log(event);
  alert("onkeyup");
};

document.querySelector("p:nth-child(2)").onmouseover = function(event) {
  console.log(event);
  alert("onmouseover");
};

document.querySelector("button").onclick = function(event) {
  console.log(event);
  alert("onclick");
};

document.querySelector("form").onsubmit = function(event) {
  console.log(event);
  alert("onsubmit");
  //event.preventDefault(); // sprečavanje slanja zahteva iz forme
};

document.querySelector("input[type=text]").onfocus = function(event) {
  console.log(event);
  alert("onfocus");
};
```


DOM (*Document Object Model*) Level 1 - 4

Style

Style objekat sadrži CSS attribute elementa.

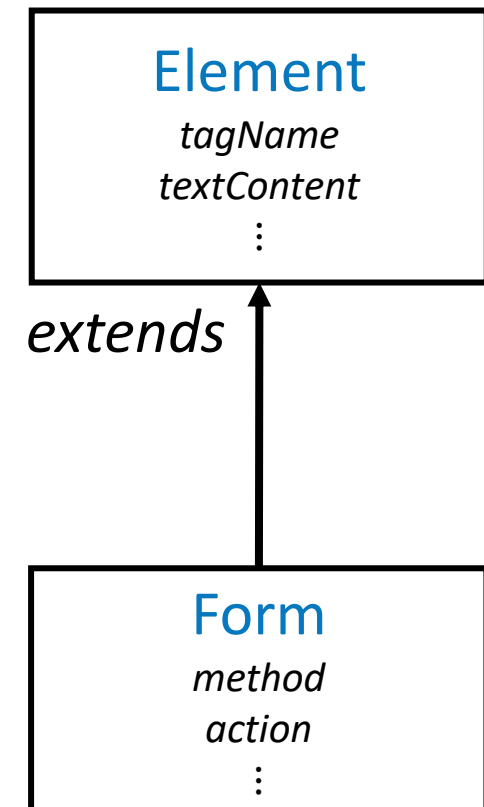
	Naziv	CSS atribut
atributi	<i>display</i>	<i>display</i>
	<i>color</i>	<i>color</i>
	<i>backgroundColor</i>	<i>background-color</i>
	<i>width</i>	<i>width</i>
	<i>borderStyle</i>	<i>border-style</i>
	<i>borderColor</i>	<i>border-color</i>
	<i>borderWidth</i>	<i>border-width</i>
	i mnogi drugi...	

DOM (*Document Object Model*) Level 1 - 4

FormElement

Modeluje HTML element koji je forma.

	Naziv	Upotreba
atributi	<i>method</i>	• vrednost <i>method</i> atributa
	<i>action</i>	• vrednost <i>action</i> atributa
	<i>name</i>	• vrednost <i>name</i> atributa
	<i>length</i>	• ukupan broj <i>input</i> elemenata u formi
	<i>elements</i>	• niz <i>input</i> elemenata u formi
metode	<i>reset()</i>	• <i>reset</i> -uje sva polja forme
	<i>submit()</i>	• programski klik na <i>submit</i> dugme
	i mnogi drugi...	

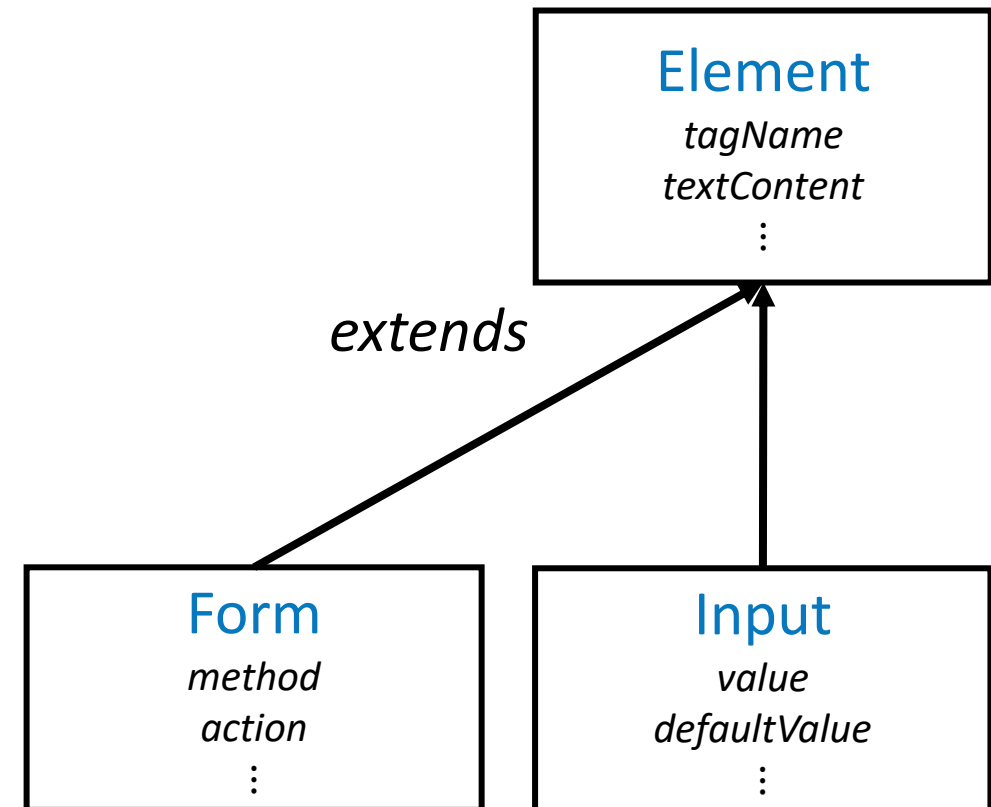


DOM (*Document Object Model*) Level 1 - 4

InputElement

Modeluje HTML element koji je *input*.

	Naziv	Upotreba
atributi	<i>value</i>	• vrednost unosa
	<i>defaultValue</i>	• početna vrednost
	<i>type</i>	• vrednost <i>type</i> atributa
	<i>name</i>	• vrednost <i>name</i> atributa
	<i>form</i>	• referenca na formu kojoj <i>input</i> pripada
	<i>checked</i>	• da li je <i>checkbox</i> ili <i>radio</i> check-iran?
	i mnogi drugi...	

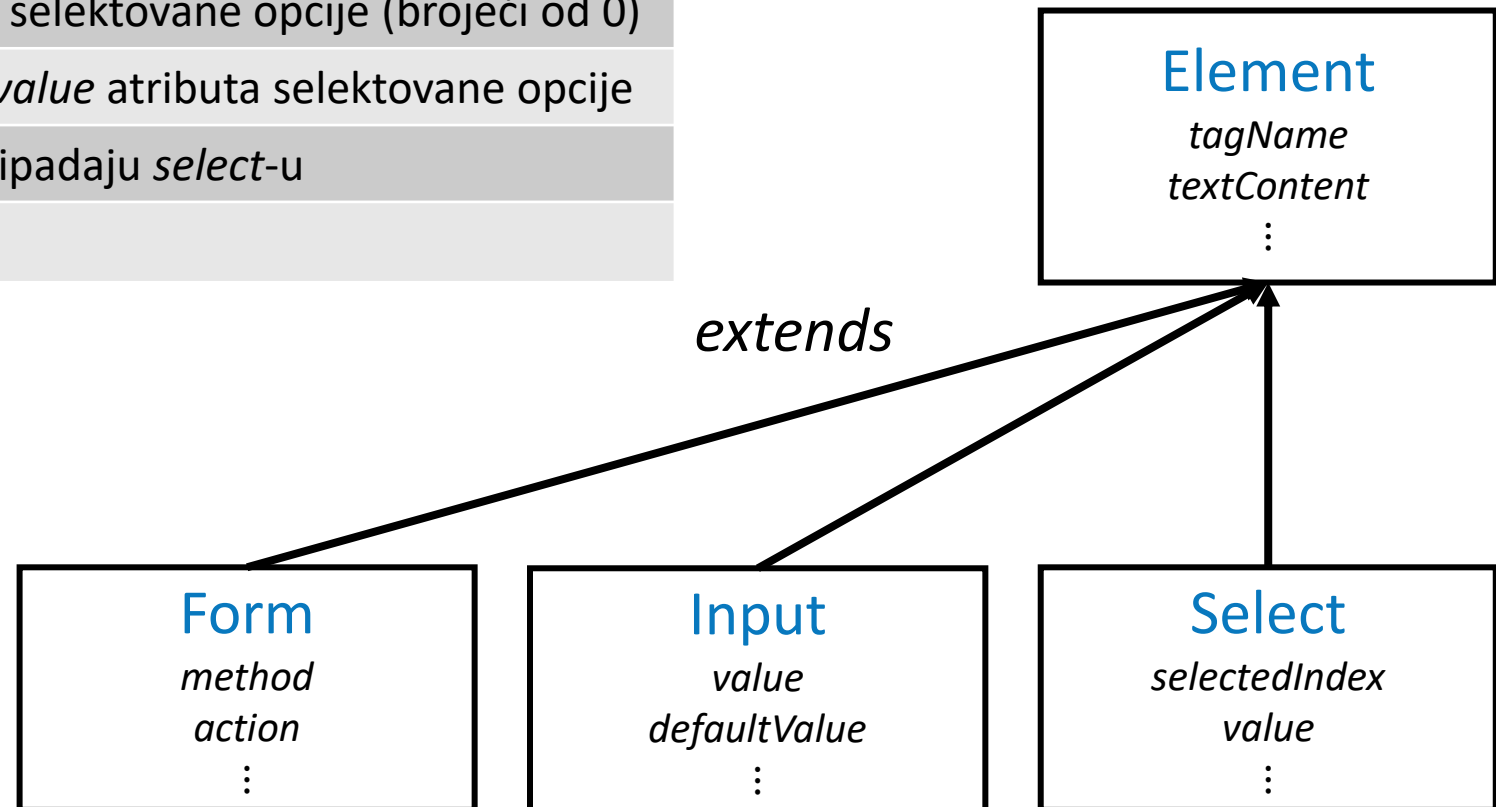


DOM (*Document Object Model*) Level 1 - 4

SelectElement

Modeluje HTML element koji je *select*.

	Naziv	Upotreba
atributi	<i>selectedIndex</i>	• sadrži redni broj selektovane opcije (brojeći od 0)
	<i>value</i>	• sadrži vrednost <i>value</i> atributa selektovane opcije
	<i>options</i>	• niz opcija koji pripadaju <i>select</i> -u
	<i>i mnogi drugi...</i>	



DOM (*Document Object Model*) Level 1 - 4

Metoda *document.write(...)*

primer5.html

Ako se pozove za vreme učitavanja stranice dopisuje sadržaj na kraj HTML dokumenta (korisno za upisivanje dinamičkog sadržaja).

Ako se pozove nakon učitavanja stranice, zamenjuje kompletan sadržaj HTML dokumenta, ali se URL ne menja.

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="UTF-8">
  <title>Funkcija write</title>
</head>
<body>
  <p>običan pasus</p>
  <script type="text/javascript">
    let datum = new Date();
    document.write(`<p>trenutno vreme je: <b>${datum.getFullYear()}.${datum.getMonth() + 1}.${datum.getDate()}. ${datum.getHours()}:${datum.getMinutes()}:${datum.getSeconds()}</b></p>`);
  </script>
  <button onclick="klikNaDugme()">kliknite na ovo dugme</button>
</body>
</html>
```

JavaScript

Link-ovi

primer6.html

Atribut *href* link-a može da sadrži *JavaScript* poziv i tada on definiše ponašanje pri kliku na *link*.

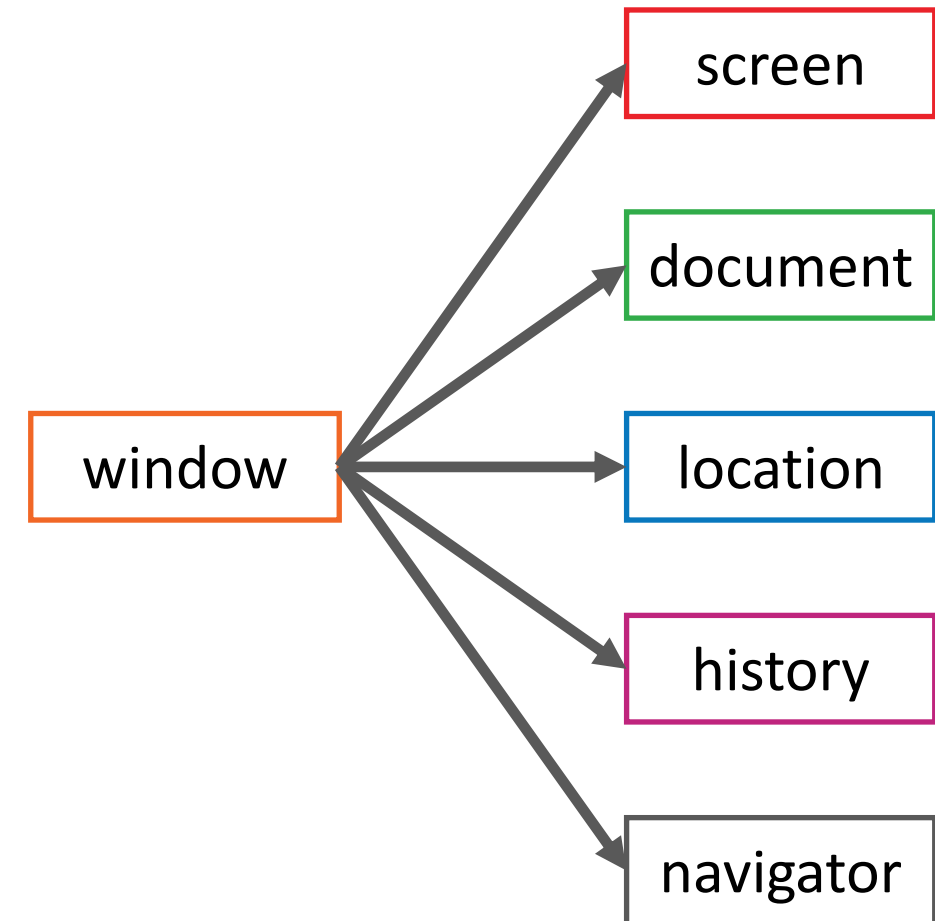
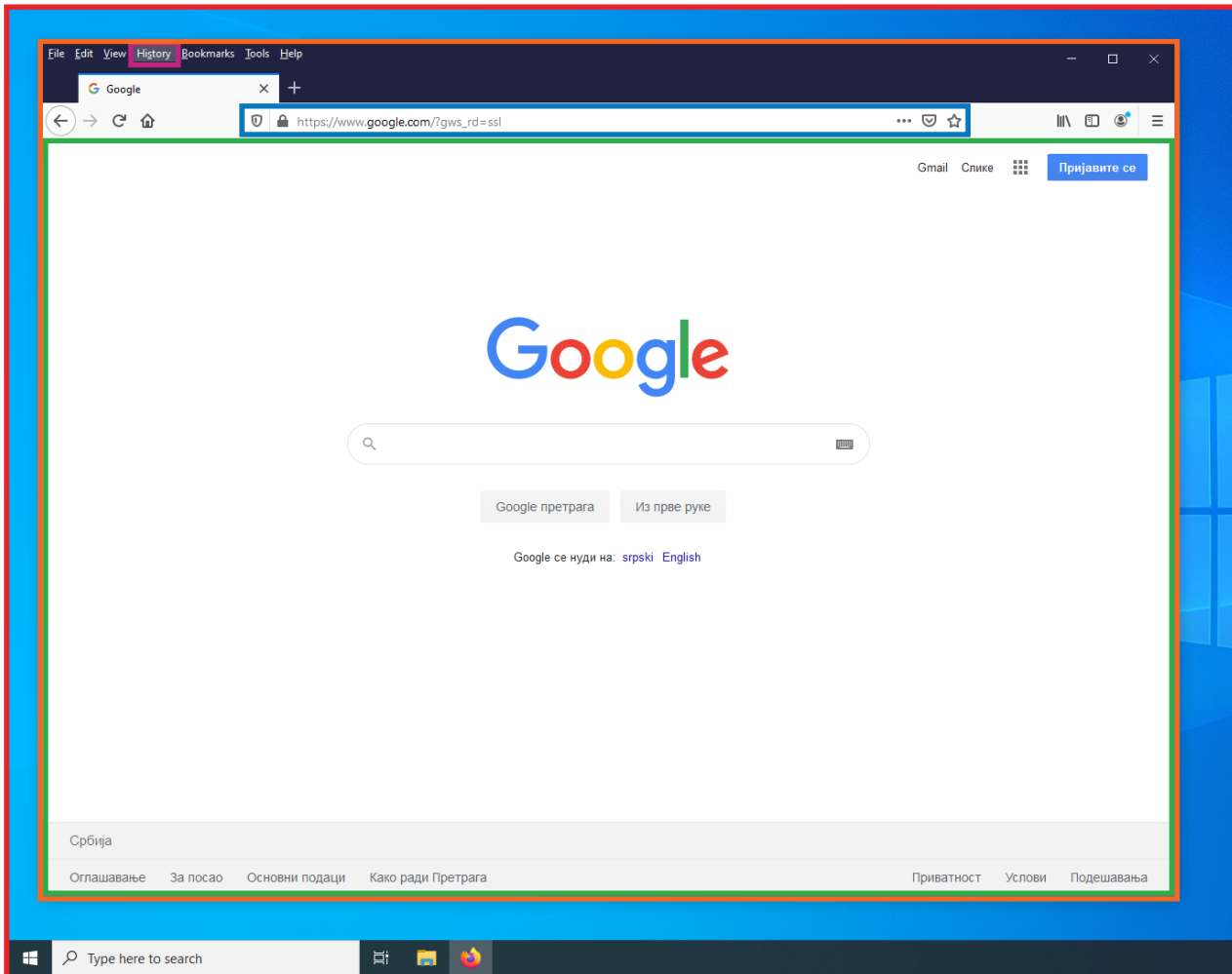
Ako funkcija koja se poziva ne vraća ništa, tada se na klikom na *link* izvršava procedura koja definisana u funkciji.

Ako funkcija koja se poziva vraća *string* tada se klikom na *link* prikazuje i interpretira taj *string* kao HTML sadržaj nove stranice na koju se prelazi, ali se URL ne menja.

```
<a href="javascript:klikNaLink()">kliknite na ovaj link</a>
```

BOM (*Browser Object Model*)

Objektni model prozora *web browser-a* koji omogućuje programsko rukovanje njegovim elementima.



BOM (*Browser Object Model*)

Window

[primer7.html](#)

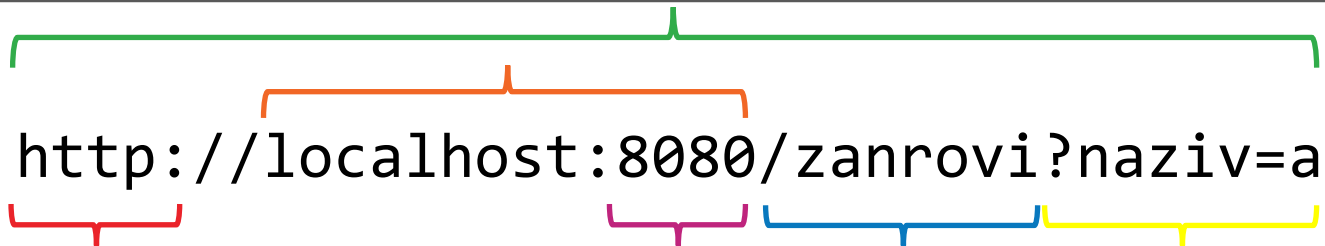
	Naziv	Upotreba
atributi	<i>screen, document, location, history, navigator</i>	<ul style="list-style-type: none">• pristup podelementima prozora
	<i>screenX, screenY</i>	<ul style="list-style-type: none">• pozicija prozora na ekranu
	<i>outerWidth, outerHeight</i>	<ul style="list-style-type: none">• dimenzije prozora
	i mnogi drugi...	
metode	<i>alert(...), confirm(...), prompt(...)</i>	<ul style="list-style-type: none">• <i>Popup Boxes</i>
	<i>back()</i> <i>forward()</i>	<ul style="list-style-type: none">• povratak na prethodnu stranicu• odlazak na sledeću stranicu
	<i>moveBy(...)</i> <i>moveTo(...)</i>	<ul style="list-style-type: none">• pomeranje prozora
	<i>open(...)</i>	<ul style="list-style-type: none">• otvara novi tab sa zadatom adresom
	<i>setTimeout("izraz", timeout)</i>	<ul style="list-style-type: none">• zadavanje <i>JavaScript</i> izraza koji će se izvršiti nakon određenog vremena
	<i>setInterval("izraz", interval)</i>	<ul style="list-style-type: none">• zadavanje <i>JavaScript</i> izraza koji će se izvršavati periodično
	i mnoge druge...	

BOM (*Browser Object Model*)

Location

primer8.html

Modeluje adresnu liniju *web browser-a*.



http://localhost:8080/zanrovi?naziv=a

	Naziv	Upotreba
atributi	<i>href</i>	kompletan URL
	<i>protocol</i>	protokol
	<i>host</i>	adresa servera
	<i>port</i>	port
	<i>pathname</i>	putanja do resursa
	<i>search</i>	parametri
metode	<i>reload()</i> <i>replace(...)</i>	<ul style="list-style-type: none">ponovo učitava tekuću adresuučitava novu adresu

BOM (*Browser Object Model*)

Navigator

primer9.html

Opisuje *web browser* u kome je učitana HTML stranica.

	Naziv	Upotreba
atributi	<i>userAgent</i>	naziv <i>web browser</i> -a
	<i>cookieEnabled</i>	da li browser podržava <i>cookies</i> ?
	<i>language</i>	jezik <i>web browser</i> -a
	i mnogi drugi...	

BOM (*Browser Object Model*)

History

Modeluje istoriju posećenih stranica.

	Naziv	Upotreba
atributi	<i>length</i>	<ul style="list-style-type: none">• ukupan broj zapamćenih posećenih stranica
	<i>current</i>	<ul style="list-style-type: none">• URL tekuće stranice
	<i>next</i>	<ul style="list-style-type: none">• URL sledeće stranice (postoji samo ako se prethodno kretalo unazad)
	<i>previous</i>	<ul style="list-style-type: none">• URL prethodne stranice (postoji samo ako je posećeno više stranica)
metode	<i>back()</i> <i>forward()</i> <i>go(...)</i>	<ul style="list-style-type: none">• povratak na prethodnu stranicu• odlazak na sledeću stranicu• odlazak na zadatu stranicu iz liste

Primeri

- *primer1.html, primer2.html, ..., primer3.html*

Primeri

- *com.ftninformatika.jwd.modul2.termin10.bioskop*

Zadatak 1

Po ugledu na primer *com.ftninformatika.jwd.modul2.termin10.bioskop*, implementirati validacije unosa i sortiranja u zadatku *com.ftninformatika.jwd.modul2.termin10.dostava*:

1. Dopuniti HTML stranice za dodavanje i prikaz jednog entiteta small elementima za ispis poruka o greški.
2. Dopuniti HTML stranice za prikaz svih entiteta dugmadima za sortiranje.
3. U direktorijumu *resources/static* kreirati poddirektorijum *js*.
4. Kreirati potrebne *.js* datoteke i uvezati ih na odgovarajuće stranice upotrebnom *script* elementa.
5. Implementirati i testirati validaciju i sortiranje.

Zadatak 2

Na stranici za prijavu u *com.ftninformatika.jwd.modul2.termin10.dostava* napraviti u okviru forme za prijavu *checkbox* za sakrivanje i otkrivanje lozinke.

Dodatni materijali

- <https://www.w3schools.com/js/>
- [https://www.w3schools.com/js/js object definition.asp](https://www.w3schools.com/js/js_object_definition.asp)
- <https://www.w3schools.com/jsref/>
- <https://developer.mozilla.org/en-US/docs/Web>