



JavaScript

Šesti dio



Pregled

- Window
 - Uvod
 - Size
 - open()
- BOM
 - Screen
 - Location
 - History
 - Navigator
 - Popup Alert
 - Timing
 - Cookies

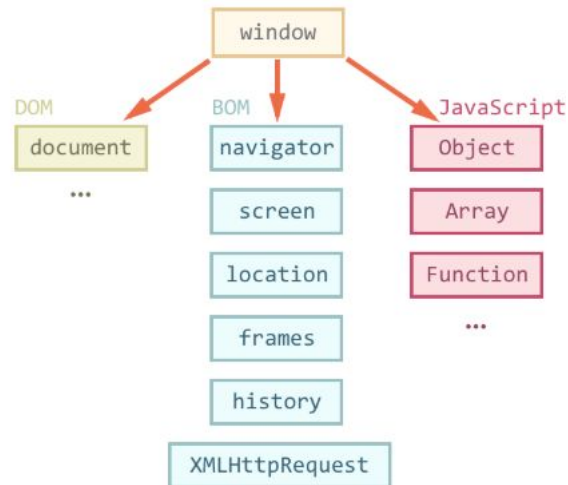


Obnavljanje

- Test 2, komentari
- Domaci 2
- Domaci 3
- Iks/oks i dodatne funkcionalnosti aplikacije koje smo radili na prethodnom času
- Koji su koraci pri dodavanju događaja na HTML element
- Što sve od informacija možemo izvući iz document objekta
- Šta smo rekli za specijalni objekat event
- Šta smo rekli, šta je bubbling, a šta capturing i koji je default
- Čemu služe metodi stopPropagation i preventDefault, specijalnog objekta event
- Opet da ponovimo, HTML DOM

Uvod

- BOM (Browser object model) omogućavaJSu da komunicira sa browser-om





Uvod, nastavak

- Window objekat postoji kod svakog pretraživača. Predstavlja window pretraživača
- Svi globalni JS objekti, funkcije i varijable automatski postaju članovi window objekta
- Globalne varijable su svojstva (properties) window objekta
- Globalne funkcije su metode window objekta
- Čak je document objekat (HTML DOM) property window objekta

```
window.document.getElementById("header");
```

je isto kao

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



Window size

- Da biste provjerili veličinu window-a (u px) koriste se dva metoda
 - **window.innerHeight** - visina browser window-a u px
 - **window.innerWidth** - širina browser window-a u px
- U veličinu prozora se ne uključuju dimenzije toolbara i scrollbara
- Isprobajmo kroz konzolu



Window open()

- Metod open() služi za otvaranje novog prozora
- Isprobajmo window.open("<https://google.com>")
- Postoji i metod close() koji se poziva nad window objektom ali najčešće je sačuvan u nekoj varijabli
 - Isprobaćemo kasnije
- Sintaksa:
 - `window.open(URL, name, specs, replace)`
- Svi argumenti metoda open() su opcioni
- **URL parametar**
 - Definišemo URL stranice koju želimo da prikazemo. Ako URL nije definisan, novi prozor se otvara kao about:blank, tj. empty window
 - Isprobajmo window.open()



Window open(), drugi dio

- **Name parametar**
 - Definiše target atribut ili ime prozora. Dozvoljene su sledeće vrijednosti
 - `_blank` - URL je učitava u novi prozor, default
 - `_self` - URL mijenja trenutno otvorenu stranicu
 - `_parent` - URL se učitava u parent frame
 - `_top` - URL bilo koji frameset koji se učitava
 - `name` - ime prozora

Window open(), treći dio

- Spec parametar (lista opcija razdvojenih zarezima), dozvoljava sledeće vrijednosti

channelmode=yes no 1 0	Whether or not to display the window in theater mode. Default is no. IE only	menubar=yes no 1 0	Whether or not to display the menu bar
directories=yes no 1 0	Obsolete. Whether or not to add directory buttons. Default is yes. IE only	resizable=yes no 1 0	Whether or not the window is resizable. IE only
fullscreen=yes no 1 0	Whether or not to display the browser in full-screen mode. Default is no. A window in full-screen mode must also be in theater mode. IE only	scrollbars=yes no 1 0	Whether or not to display scroll bars. IE, Firefox & Opera only
height=pixels	The height of the window. Min. value is 100	status=yes no 1 0	Whether or not to add a status bar
left=pixels	The left position of the window. Negative values not allowed	titlebar=yes no 1 0	Whether or not to display the title bar. Ignored unless the calling application is an HTML Application or a trusted dialog box
location=yes no 1 0	Whether or not to display the address field. Opera only	toolbar=yes no 1 0	Whether or not to display the browser toolbar. IE and Firefox only
		top=pixels	The top position of the window. Negative values not allowed
		width=pixels	The width of the window. Min. value is 100



Window open(), četvrti dio

- Replace parametar služi da kreira novi unos ili izmijeni postoji unos u history. Podržane su sledeće vrijednosti:
 - true - URL mijenja trenutno dokument i history listi
 - false - URL kreira novi unos u history listi
 - Primjeri (code-1)



Screen

- Sadrži informacije o korisnikovom screen-u
- Posjeduje sledeća svojstva:
 - `screen.width`
 - Vraće širinu screen-a u pikselima
 - `screen.height`
 - Vraće visinu screen-a u pikselima
 - `screen.availWidth`
 - Vraće širinu screen-a u pikselima minus neki delovi interfejsa, npr. taskbar
 - `screen.availHeight`
 - Vraće visinu screen-a u pikselima minus neki delovi interfejsa, npr. taskbar
 - `screen.colorDepth`
 - Vraće broj bita potrebnih za prikaz jedne boje
 - `screen.pixelDepth`
- Isprobajmo ova svojstva



Location

- Koristi se za prikaz stranice sa novim URL-om
 - `window.location.href` vraće href (URL) trenutne stranice
 - `window.location.hostname` vraće domain name web host-a
 - `window.location.pathname` vraće putanju i ime fajla trenutne stranice
 - `window.location.protocol` vraće web protokol koji se koristi
 - Kako biste napravili otvaranje nekog fajla pomoću location?
 - Kako biste omogućili download fajla?
 - Isprobajmo ovo u konzoli



History

- Sadži informaciju o istoriji pretraga
- Da bi se zaštitila privatnost korisnika, postoje određeni limiti kako JS pristupa ovom objektu
- Neke od metoda
 - `history.back()`
 - `history.forward()`
 - Isprobajmo i ovo



Navigator

- Koristi informacije o posjetiocu browsera
- Neka od svojstava navigatora
 - navigator.cookieEnabled
 - Vraće true ili false u zavisnosti od toga da li je korisnik omogućio cookie ili ne
 - navigator.appName
 - Vraće ime aplikacije pretraživača
 - Vraće “Netscape” za IE11, Chrome, Firefox, Safari
 - navigator.appCodeName
 - Vraće ime aplikativnog koda pretraživača
 - Mozilla je application code name za Chrome, Firefox, IE, Safari, Opera
 - navigator.product
 - Vraće ime proizvoda pretraživača
 - Ne oslanjajte se na ovo, većina pretraživača ima Gecko kao product name



Navigator, nastavak (1)

- Još metoda i svojstava
 - navigator.appVersion
 - Vraće informacije o verziji pretraživača
 - navigator.userAgent
 - Vraće user-agent header koji je poslat od strane pretraživača ka serveru
 - navigator.platform
 - Vraće platformu pretraživača (operativni sistem)
 - Navigator.language
 - Vraće jezik pretraživača
 - Navigator.onLine
 - Vraće true ili false u zavisnosti od toga da li je browser online
 - navigator.javaEnabled()
 - Vraće da li je Java enabled



Navigator, nastavak(2)

- Nije baš precizan i treba voditi računa kako i kada se koristi
 - Različiti pretraživači mogu da koriste isto ime
 - Navigator podaci mogu biti izmijenjeni od vlasnika browsera
 - Neki pretraživači se pogrešno “predstave” da bi prošli neke testove sajta
 - Pretraživač ne može da prijavi novi operativni sistem, objavljen nakon nove verzije pretraživača



Popup alert

- JavaScript ima tri vrste popup-ova: alert box, confirm box i prompt box
- Alert box se najčešće koristi ako želite da prikazete neke informacije korisniku
 - Koristi se **alert**("some text") za pozivanje ovog metoda
- Confirm box se koristi kada korisnik treba da prihvati ili odbije nešto
 - Koristi se **confirm**("some text") za pozivanje ovog metoda
 - Ako se klikne na OK ovaj metod vraće true
 - Ako se klikne na Cancel ovaj metod vraće false
- Prompt box se koristi najčešće kada želite da prije neko korisnik pristupi stranici otkuca nešto
 - Koristi se **prompt**("some text", "default text")
 - Ako se klikne na OK ovaj metod vrati tekst iz prompt-a
 - Ako se kline na Cancel ovaj metod vrati null



Timing

- JS može biti izvršen u vremenskim intervalima
 - Ovaj koncept zovemo timing events
- Window objekat omogućava izvršavanje koda u određenim vremenskim intervalima
- Dva ključna metoda za timing events su
 - `setTimeout(function, milliseconds)`
 - Izvrši funkciju na definisanog broja milisekundi
 - `setInterval(function, milliseconds)`
 - Izvršavaj fuknciju svakih unaprijed definisanih milisekundi
 - Da li je moguće prekinuti interval?
 - Koristi se `clearTimeout(timeoutVariable)`, ali prethodno `setInterval` mora biti sačuvan
 - Primjer, code-2



Cookies

- Omogućavaju čuvanje korisničkih informacija na web stranici
 - Ti podaci se čuvaju na vašem računaru
- Kada web server pošalje web stranicu pretraživaču, konekcija se gasi i server zaboravlja sve o korisniku
- Cookies su uvedeni da riješe sledeći problem “Kako da se sačuvaju informacije o korisniku”
 - Kada korisnik sledeći put posjeti stranicu, njegovo ime može biti sačuvano u cookie
 - Sledeći put kada korisnik posjeti web stranicu, cookie pamti njegovo ime
 - Cookies se čuvaju kao name-value par
 - username = “John Doe”
- Kada pretraživač pošalje zahtjev za web stranicu serveru, cookies koji odgovaraju toj stranicu budu dodati u zahtjev koji se šalje serveru. Na taj način server uzima neophodne informacije koje se pamte o korisniku
- Napomena: Cookies moraju biti omogućeni da bi radili



Kreiranje cookie-a

- Sa `document.cookie` možete da kreirate, učitajte ili obršete cookie sa `document.cookie`
 - `document.cookie = "username=John Doe"`
 - `document.cookie = "username=John Doe; expires=Wen, 10 Oct 2018 12:00:00 UTC"`
 - Sa `expires` definišete kad cookie ističe
 - Po defaultu cookie se briše kad se zatvori pretraživač
 - `Document.cookie ="username=John Doe; expires=Wen, 10 Oct 2018 12:00:00 UTC; path=/"`
 - Sa parametrom `path` kažete pretraživaču kom path cookie pripada
 - Default je current page



Učitavanje, mijenjanje i brisanje cookie-a

- Za čitanje cookie-a
 - `document.cookie` vraće sve cookie-e u jednom stringu, npr.
 - `cookie1=value; cookie2=value; cookie3=value;`
- Za ažuriranje
 - Na isti način kao kreiranje
 - Stari cookie biva prebrisan (overwritten)
- Za brisanje
 - Kao i kreiranje ali se za `expires` definiše datum koji je već istekao
 - `document.cookie = "username=John Doe; expires=Wen, 10 Oct 2000 12:00:00 UTC"`



Cookie string

- String koji vraće `document.cookie` izgleda kao normalni tekstualni string ali nije
- Ako setujete novi cookie, stari ostaje sačuvan. Novi cookie se dodaje u `document.cookie`, pa ako ponovo probate da pročitate cookie, vidjećete nešto nalik
 - `cookie1 = value; cookie2 = value;`
- Ako želite da pročitate određeni cookie, morate napisati JS funkciju za to



Cookie primjer

- U našem primjer napravićemo cookie koji čuva ime posjetioca stranice
- Prvi put kada korisnik dođe na stranicu, od njega ćemo tražiti da popuni ime. Ime se čuva u cookie
- Sledeći put kada korisnik dođe na stranicu, dobiće poruku sa imenom koje je unio kada je posjetio stranicu prvi put
- Kreiraćemo tri funkcije
 - Funkciju za setovanje cookie-a
 - Funkciju za getovanje cookie-a
 - Funkciju za provjeru vrijednosti cookie-a
- `encodeURIComponent(uri)` i `decodeURIComponent(uri_enc)`
 - Najčešće se koristi za enkodovanje i dekodovanje linkova
 - Mi ćemo koristiti za dekodovanje cookie-a
 - Za encode [link](#)
 - Za decode [link](#)
- Napomena: Instalirati live-server globalno kroz terminal VSCode-a



Pitanja