STANDARDNI KORISNIČKI INTERFEJSI

Predavanje broj: 11

Nastavna jedinica: JavaScript

Nastavne teme:

BOM. Objekat window (screen, location, navigator, history, popup boxes, timing events). Cookies (postavljanje, uzimanje, provera). Biblioteka jQuery: selektori, metode, svojstva. Interfejs Page Visibility. Interfejs: Drag And Drop.

Predavač: prof. dr Perica S. Štrbac, dipl. ing.

Literatura:

J. D. Gauchat, "Integrisane tehnologije za izradu WEB strana", Mikroknjiga, Beograd, 2014.

W3C Tutorials, Internet, 2014

BOM

- Browser Object Model (BOM) omogućuje JavaScript-u da komunicira sa browser-om.
  - Ne postoji oficijelni standard za BOM.
  - JavaScript metode i svojstva za interaktivnost se smatraju metodama i svojstvima BOMa.
- **Objekat window**
- Window objekat predstavlja prozor browser-a.
- Svi globalni JavaScript objekti, funkcije i varijable automatski postaju članovi window objekta.
  - Čak je i objekat document (HTML DOM) svojstvo window objekta.

    window.document.getElementById("header");

    ili

    document.getElementById("header");

- Svojstva window objekta koja se odnose na dimenzije su:
  - window.innerWidth - unutrašnja širina browser-ovog prozora
  - window.innerHeight - unutrašnja visina browser-ovog prozora

window

- U datom primeru određuju se dimenzije prozora (ne uključuje toolbarove i scroll barove). Respektivno se uzima prva dimenzija (širina ili visina) različita od 0, za: window.innerWidth, document.documentElement.clientWidth, document.body.clientWidth.

```
<!DOCTYPE html>
<html><body>
<p id="demo"></p>
<script>
var w =    window.innerWidth
        || document.documentElement.clientWidth
        || document.body.clientWidth;
var h =    window.innerHeight
        || document.documentElement.clientHeight
        || document.body.clientHeight;

var x = document.getElementById("demo");
x.innerHTML = "Browser inner window width: " + w + ", height: " +
h + ".";
</script>
</body></html>
```

window

- Metode window objekata su:
  - window.open()
  - window.close()
  - window.moveTo()
  - window.resizeTo()
- Objekat window.screen sadrži informacije o ekranu korisnika. Može se pisati i bez prefiksa window. Sadrži sledeća svojstva:
  - screen.width, širina ekrana u pikselima
  - screen.height, visina ekrana u piskelima
  - screen.availWidth
    - širina ekrana u piselima umanjena za širinu interfejsa kao npr. Windows-ovog Taskbar-a.
  - screen.availHeight
    - visina ekrana u piselima umanjena za visinu interfejsa kao npr. Windows-ovog Taskbar-a.
  - screen.colorDepth, broj bitova kojim se prikazuju boje
  - screen.pixelDepth

window.screen

- Primer koji ispisuje vrednosti svih nabrojanih svojstava:

```
<!DOCTYPE html>
<html>
<body>

<p id="demo"></p>

<script>
document.getElementById("demo").innerHTML =
"Screen width is "          + screen.width      + "<br>" +
"Screen height is "         + screen.height     + "<br>" +
"Available Screen Width: " + screen.availWidth + "<br>" +
"Screen Color Depth: "      + screen.colorDepth + "<br>" +
"Screen Pixel Depth: "      + screen.pixelDepth;;
</script>

</body>
</html>
```

Screen width is 1680
Screen height is 1050
Available Screen Width: 1618
Screen Color Depth: 24
Screen Pixel Depth: 24

window.location

- Objekat window.location može se koristiti da se uzme tekuća adresa stranice (URL) i da se browser preusmeri na na novu stranicu.

Window Location

- Objekat window.location može se pisati i bez prefiksa window.
- Primeri:
  - window.location.href
    - vraća href (URL) tekuće stranice
  - window.location.protocol
    - vraća web protocol koji se koristi
  - window.location.hostname
    - vraća naziv domena web hosta
  - window.location.pathname
    - vraća stazu i ime datoteke tekuće stranice
  - window.location.assign
    - učitava nov dokument

window.location

- Primer ispisa podataka o lokaciji:

```
<!DOCTYPE html>
<html><body>
<p>Display the entire URL
 of the current page.</p>
<p id="demo"></p>
<script>
document.getElementById("demo").innerHTML =
"Page location is: " + window.location.href     + "<br>" +
"Page hostname is: " + window.location.hostname + "<br>" +
"Page path is    : " + window.location.pathname + "<br>" +
"Page protocol is: " + window.location.protocol ;
</script></body></html>
```

Display the entire URL of the current page.

Page location is: file:///C:/__Projects/location.html
Page hostname is:
Page path is : /C:/__Projects/location.html
Page protocol is: file:

- Učitavanje novog dokumanta:

```
<!DOCTYPE html>
<html><head><script>
function newDoc(){ window.location.assign("http://www.b92.com");}
</script></head><body>
<input type="button" value="Load new document" onclick="newDoc()">
</body></html>
```

window.history

- Objekat window.history sadrži istoriju browser-a.
  - Objekat window.history može se pisati bez prefiksa window.
- Metode:
  - window.history.back - prethodna stranica
  - window.history.forward - sledeća stranica

```
<!DOCTYPE html>
<html><head>
<script>
function goBack()    {  window.history.back()    }
function goForward() {  window.history.forward() }
</script>
</head>
<body>
<input type="button" value="Back" onclick="goBack()">
<input type="button" value="Forward" onclick="goForward()">
</body>
</html>
```

window.navigator

- Objekat window.navigator sadrži informacije o browseru (može se pisati i bez prefiksa window).
- Svojstva:
  - window.navigator.appName, naziv browser-a
  - window.navigator.appCodeName, kodni naziv browser-a
    - Obično je odgovor "Netscape" ili "Mozilla".
  - window.navigator.cookieEnabled, da li su cookie-i omogućeni
  - window.navigator.product , naziv browser endžina(npr. Gecko)
  - window.navigator.appVersion, verzija aplikacije
    - ili window.navigator.userAgent
  - window.navigator.platform, operativni sistem browser-a
  - window.navigator.language, jezik browser-a
- Metoda:
  - window.navigator.javaEnabled();
    - vraća da li je Java omogućena

window.navigator

- Ispis podataka o browser-u:

```
<!DOCTYPE html>
<html>
<body>
<p id="demo"></p>
<script>


(function () {
    document.getElementById("demo").innerHTML =
     "<br>1. Name is "             + navigator.appName       +
     "<br>2. Code name is "        + navigator.appCodeName   +
     "<br>3. Cookies enabled is " + navigator.cookieEnabled +
     "<br>4. Browser engine is "   + navigator.product       +
     "<br>5. Application version is "+navigator.appVersion  +
     //ili navigator.userAgent
     "<br>6. Browser platform "     + navigator.platform    +
     "<br>7. Browser language is " + navigator.language    +
     "<br>8. Java enabled is "      + navigator.javaEnabled();
})();
</script></body></html>
```

1. Name is Netscape
2. Code name is Mozilla
3. Cookies enabled is true
4. Browser engine is Gecko
5. Application version is 5.0 (Windows NT 6.1; WOW64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/42.0.2311.135 Safari/537.36
6. Browser platform Win32
7. Browser language is en-US
8. Java enabled is true

window popup boxes

- Ranije smo koristili (malo podsećanje kroz primere):
  - window. alert("Hello\nHow are you?");
    ili
    alert("Hello\nHow are you?");
  - window.confirm("text"); ili confirm("text");

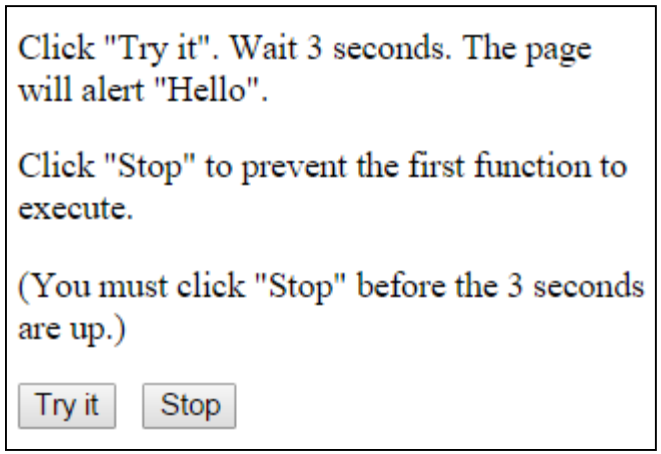
```
<!DOCTYPE html>
<html><body><p id="demo"></p>  <script>
(function () {
 var x;
 if (confirm("Press a button!")==true)x = "You pressed OK!";
 else                                 x = "You pressed Cancel!";
 document.getElementById("demo").innerHTML = x;
})();   </script></body></html>
```

  - window.prompt

```
<!DOCTYPE html>
<html><body> <p id="demo"></p> <script>
(function () { var hm = prompt("Vaše ime ?", "no name");
  if (hm!= null && hm!="")
    document.getElementById("demo").innerHTML="Vi ste "+hm;
})();   </script></body></html>
```

window timing events

- Metode za vremenske događaje, koji aktiviraju kod u specificiranom vremenu su:
  - window.setTimeout("*javascript function*", *milliseconds*); - izvršava datu funkciju jednom nakon isteka specificiranog vremena u milisekundama
  - window.clearTimeout(*timeoutVariable*);
    - zaustavlja izvršavanje date funkcije

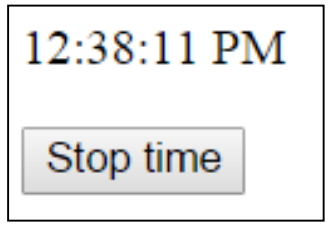



```
<!DOCTYPE html>
<html><body>


<p>Click "Try it". Wait 3 seconds. The page will alert
"Hello".</p>
<p>Click "Stop" to prevent the first function to execute.</(p>
<p>(You must click "Stop" before the 3 seconds are up.)</p>

<button onclick="myVar =
   setTimeout(function(){alert('Hello')},3000)">Try it</button>

<button onclick="clearTimeout(myVar)">Stop</button>
</body></html>
```

window timing events

- window.setInterval("*javascript function*", *milliseconds*);
  - izvršava datu funkciju iznova u specificaranim intervalima vremena u milisekundama

```
<!DOCTYPE html>
<html><body>   <p id="demo"></p>   <script>
var myVar=setInterval(function(){myTimer()},1000);
function myTimer() {
    var d = new Date();
    document.getElementById("demo").innerHTML =
                              d.toLocaleTimeString();
} </script></body></html>
```



- window.clearInterval(*setintervalVariable*); zaustavlja izvršavanje date funkcije

```
<!DOCTYPE html><html><body><p id="demo"></p><button
onclick="clearInterval(myVar)" >Stop time</button> <script>
  var myVar = setInterval(function(){myTimer()},1000);
  function myTimer() {
    var d = new Date();
    document.getElementById("demo").innerHTML =
                            d.toLocaleTimeString();
  }</script></body></html>
```

cookie

- Kolačići (cookies) omogućuju da se zapiše informacija npr. o korisniku na web stranici.
  - Cookies su podaci snimljeni u malom tekstualnom fajlu na računaru.
- Kada web server pošalje stranicu browser-u korisnika, konekcija se gasi i server zaboravlja sve o korisniku.
- Cookies su smišljeni da reše problem pamćenja informacije o korisniku jer http protokol ne pamti stanja (stateless).
- Kada korisnik poseti web stranicu njegovo ime može biti snimljeno u kolačić (cookie).
- Sledeći put kada korisnik poseti stranicu, cookie "pamti" njegovo ime.
- Cookies se snimaju kao parovi ime-vrednost:

```
username=John Doe
```

- Kada browser zahteva web stranicu sa servera, cookie koji pripadaju stranici biće dodati ovom zahtevu.
  - Na ovaj način server dobija potrebne podatke da poseredno "zapamti" informaciju o korisniku.
    - Obično se na strani servera koristi baza podataka

cookie

- U JavaScript-u cookie se kreiraju kao što sledi:

```
document.cookie="username=John Doe";
```

- Može se dodati i datum isteka (UTC vreme). Podrazumevano, cookie se briše kada se browser zatvori:

```
document.cookie="username=John Doe; expires=Thu, 18 Dec 2013
12:00:00 UTC";
```

- Sa parametrom koji označava stazu (path parameter) može se browser-u proslediti i koja staza pripada cookie-u.
  - Podrazumevano, cookie pripada tekućoj stranici.

```
document.cookie="username=John Doe; expires=Thu, 18 Dec 2013
12:00:00 UTC; path=/";
```

- Čitanje cookie-a pomoću JavaScript-a je kao što sledi:

```
var x = document.cookie;
```

  - document.cookie vraća sve cookie u jednom stringu, npr.

```
cookie1=value; cookie2=value; cookie3=value;
```

cookie

- Pomoću JavaScript-a može se menjati cookie na isti način kao i što se kreira:

```
document.cookie="username=John Smith; expires=Thu, 18 Dec 2013 12:00:00 UTC; path=/";
```

  - ovim će cookie biti promenjen

- Brisanje cookie-a pomoću JavaScript-a se radi tako da se postavi da je datum isteka cookie-a neki u prošlosti, kao što sledi:

```
document.cookie = "username=; expires=Thu, 01 Jan 1970 00:00:00 UTC";
```

Napomena: ne mora se navoditi vrednost cookie-a koga želimo da obrišemo

Baratanje cookie-ima, postavljanje cookie-a

- U primeru koji sledi kreira se cookie koji se odnosi na ime posetioca.
- Kada posetilac prvi put poseti na web stranicu, biće pitan za svoje ime.
  - Uneseno ime će onda biti pohranjeno u cookie.
- Kada sledeći put korisnik poseti navedenu stranicu dobiće pozdravnu poruku u kojoj se navodi i njegovo ime.
- Za primer koji sledi kreirane su 3 JavaScript funkcije:
  - Funkcija koja postavlja vrednost cookie-a

```
function setCookie(cname, cvalue, exdays) {
    var d = new Date();
    d.setTime( d.getTime() + (exdays*86400*1000) );
    var expires = "expires="+d.toUTCString();
    document.cookie = cname + "=" + cvalue
                                  + "; "
                                  + expires;
}
```

  - Funkcija koja uzima vrednost cookie-a
  - Funkcija koja proverava vrednost cookie-a

Baratanje cookie-ima, uzimanje cookie-a

- Daje se funkcija koja može poslužiti za uzimanje vrednosti traženog cookie-a čiji je naziv prosleđen kao parametar:

```
function getCookie(cname) {
    var name = cname + "=";
    var ca = document.cookie.split(';');
    for(var i=0; i<ca.length; i++) {
        var c = ca[i];
        while (c.charAt(0)==' ') c = c.substring(1);
        if (c.indexOf(name)== 0)
            return c.substring(name.length,c.length);
    }
    return "";
}
```

  - Funkciji getCookie prosleđuje se naziv cookie-a kome se dodaje znak =.
  - Formira se niz stringova koji su dobijeni deljenjem stringa document.cookie delimiterom ;
  - Prolazi se kroz elemente niza. Ignorišu se razmaci (' ') koji se mogu naći na početku ekstrahovanog niza, a onda se proverava da li je odabrani element (cookie) onaj koji se traži.

Baratanje cookie-ima, provera cookie-a

- Funkcija koja proverava cookie:
  - ideja je da se proveri da li je cookie *username* postavljen.
- Ako je cookie *username* postavljen ispisaće se pozdravna poruka.
- Ako cookie *username* nije postavljen, prikazaće se prompt box pri čemu se traži da se unese ime korisnika, a onda po unosu imena snima se cookie username uz vreme isteka od 365 dana pozivom funkcije setCookie:

```
function checkCookie() {
    var username=getCookie("username");
    if (username!="") {
        alert("Welcome again " + username);
    }else{
        username = prompt("Please enter your name:", "");
        if (username != "" && username != null) {
            setCookie("username", username, 365);
        }
    }
}
```

Primer 1/2

```
<!DOCTYPE html>
<html><head>
<script>
function setCookie(cname,cvalue,exdays) {
    var d = new Date();
    d.setTime(d.getTime() + (exdays*86400*1000));
    var expires = "expires=" + d.toGMTString();
    document.cookie = cname+"="+cvalue+"; "+expires;
}
function getCookie(cname) {
    var name = cname + "=";
    var ca = document.cookie.split(';');
    for(var i=0; i<ca.length; i++) {
        var c = ca[i];
        while (c.charAt(0)==' ') c = c.substring(1);
        if (c.indexOf(name) == 0) {
            return c.substring(name.length, c.length);
        }
    }
    return "";
}
```

Primer 2/2

```
function checkCookie() {
    var user=getCookie("username");
    if (user != "") {
        alert("Welcome again " + user);
    } else {
       user = prompt("Please enter your name:","");
       if (user != "" && user != null) {
           setCookie("username", user, 30);
       }
    }
}
</script>
</head>
<body onload="checkCookie()">
</body>
</html>
```

- Kada se učita stranica, proverava se da li postoji cookie username,

–ako ne postoji kreira se cookie čiji je naziv username a vrednost odgovara unesenom imenu i čije je trajanje 30 dana.

–ako postoji ispisuje se poruka o dobrodošlici

CDN, jQuery

- CDN (Content Delivery Network) je mreža servera koji sadrže biblioteke deljenog koda.
- jQuery popularan framework-a JavaScript-a i može se uključiti u stranicu kao što sledi:

```
<script src="https://ajax.googleapis.com/ajax/libs/jquery/1.11.1/jquery.min.js"></script>
```

ili ako downloadujete datoteku: `<script src="jquery.min.js"></script>`

- Mala analogija sa onim što je pokazano ranije:

```
function myFunction() {
    document.getElementById("h01").innerHTML = "Hello.";
}
addEventListener('load', myFunction);
```

radi isto što i:

```
function myFunction() {
    $("#h01").html("Hello.");
}
$(document).ready(myFunction);
```

paziti: jQuery vraća jQuery objekat

jQuery

- Ulančavanje u jQuery-ju:

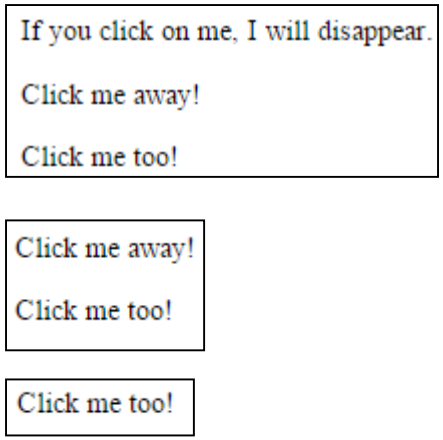
```
<!DOCTYPE html>
<html>
<head>
<script
src="http://ajax.googleapis.com/ajax/libs/jquery/1.11.1/jquery.min
.js">
</script>
<script>
function myFunction() {
    $("#h01").attr("style","color:red").html("Hello jQuery")
}
$(document).ready(myFunction);
</script>
</head>
<body>
<h1 id="h01"></h1>
</body>
</html>
```

Hello jQuery

jQuery

- Reagovanje na događaj (u primeru klik na bilo koji <p> učiniće da isti nestane):

```
<!DOCTYPE html>
<html>
<head>
<script
src="http://ajax.googleapis.com/ajax/libs/jquery/1.11.2/jquery.min
.js"></script>
<script>
$(document).ready(function(){
    $("p").click(function(){
        $(this).hide();
    });
});
</script>
</head>
<body>
<p>If you click on me, I will disappear.</p>
<p>Click me away!</p>
<p>Click me too!</p>
</body></html>
```



jQuery selektori

- Selektovanje svih elemenata stranice i postavljanje žute pozadine:

```
$(document).ready(function(){
    $("*").css("background-color", "yellow");
});
```

- Selektovanje svih elemenata u body elementu i postavljanje za njih žute pozadine:

```
$(document).ready(function(){
    $("body *").css("background-color", "yellow");
});
```

- Selektovanje elementa za dati id="intro" i postavljanje žute pozadine:

```
$(document).ready(function(){
    $("#intro").css("background-color", "yellow");
});
```

- Selektovanje elemenata za datu klasu (class="intro", u navođenju .intro) i postavljanje žute pozadine:

```
$(document).ready(function(){
    $(".intro").css("background-color","yellow");});
```

jQuery selektori

- Selektovanje elemenata koji imaju neku od navedenih klasa i postavljanje žute pozadine :

```
$(document).ready(function(){
    $(".intro, .demo, .end").css("background-color", "yellow");
});
```

- Selektovanje više različitih elemenata (tagova) i postavljanje žute pozadine:

```
$(document).ready(function(){
    $("h2, div, span").css("background-color", "yellow");
});
```

- Selektovanje prvog/poslednjeg datog elementa i postavljanje žute pozadine:

```
$(document).ready(function(){
    $("p:first").css("background-color", "yellow");
    $("p:last").css("background-color", "yellow");
});
```

jQuery selektori

- Selekcija parnih/neparnih elemenata (indeks ide od nule), npr. parnih/neparnih redova:

```
$(document).ready(function(){
 $("tr:odd").css("background-color", "yellow");
 $("tr:even").css("background-color", "green");
});
```

- Selekcija elemenata (npr. <p>) koji su prvo dete:

```
$(document).ready(function(){
    $("p:first-child").css("background-color", "yellow");
});
```

- Selekcija elemenata (npr. <p>) koji su prvi takav tip deteta:

```
$(document).ready(function(){
    $("p:first-of-type").css("background-color", "yellow");
});
```

jQuery selektori

- Selekcija elemenata (npr. <p>) koji su poslednje dete:

```
$(document).ready(function(){
    $("p:last-child").css("background-color", "yellow");
});
```

- Selekcija elemenata (npr. <p>) koji su poslednji takav tip deteta:

```
$(document).ready(function(){
    $("p:last-of-type").css("background-color", "yellow");
});
```

- Selekcija elemenata (npr. <p>) koji su 3. dete:

```
$(document).ready(function(){
    $("p:nth-child(3)").css("background-color", "yellow");
});
```

- Selekcija elemenata (npr. <p>) koji su 3. dete brojeći od nazad:

```
$(document).ready(function(){
    $("p:nth-last-child(3)").css("background-color", "yellow");
});
```

jQuery selektori

- Selekcija elemenata (npr. <p>) koji su 3. dete takvog tipa:

```
$(document).ready(function(){
    $("p:nth-of-type(3)").css("background-color", "yellow");
});
```

- Selekcija elemenata (npr. <p>) koji su 3. dete takvog tipa brojeći od nazad:

```
$(document).ready(function(){
  $("p:nth-last-of-type(3)").css("background-color", "yellow");
});
```

- Selekcija elemenata (npr. <p>) koji su jedino dete:

```
$(document).ready(function(){
  $("p:only-child").css("background-color", "yellow");
});
```

- Selekcija elemenata (npr. <p>) koji su jedino takvo dete:

```
$(document).ready(function(){
  $("p:only-of-type").css("background-color", "yellow");
});
```

jQuery selektori

- Kod svih navedenih jQuery selektora ako se navede kao što sledi:

```
$(":dati_pseudo_selektor")
```

biće selektovani svaki element (svaki tag) koji zadovoljava dati_pseudo_selektor npr. selekcija svih elemenata koji su jedino takvo dete:

```
$(":only-of-type")
```

- Tabela ostalih jQuery selektora:

Selektor	jQuery sintaksa	Opis selekcije
parent>child	$("div > p")	Svi <p> su deca taga
parent descendant	$("div p")	Svi <p> su potomci elementa
element+next	$("div + p")	Tag <p> je prvi brat iza -a
element~siblings	$("div ~ p")	Svi <p> su braća iza -a
:eq(*index*)	$("ul li:eq(3)")	"Četvrti" član liste, indeks=3 (indeks ide od 0)
:gt(*no*)	$("ul li:gt(3)")	Članovi liste liste sa indeksom>3
:lt(*no*)	$("ul li:lt(3)")	Članovi liste sa indeksom<3
:not(*selector*)	$("input:not(:empty)")	Input elementi NISU prazni

jQuery selektori

Selektor	jQuery sintaksa	Opis selekcije
`:header`	`$(":header")`	Svi header-i
`:animated`	`$(":animated")`	Svi animirani elementi
`:focus`	`$(":focus")`	Element koji trenutno ima fokus
`:contains(`*`text`*`)`	`$(":contains('Hello')")`	Elementi koji sadrže tekst "Hello"
`:has(`*`selector`*`)`	`$("div:has(p)")`	Svi elementi koji imaju <p> element
`:empty`	`$(":empty")`	Svi prazni elementi
`:parent`	`$(":parent")`	Svaki roditelj
`:hidden`	`$("p:hidden")`	Svi skriveni <p> tagovi
`:visible`	`$("table:visible")`	Sve vidljive tabele
`:root`	`$(":root")`	Korenski element dokumenta
`:lang(`*`language`*`)`	`$("p:lang(de)")`	Svi <p> kod kojih lang atribut počinje sa "de"

jQuery selektori

Selektor	jQuery sintaksa	Opis selekcije
[*attribute*]	$("[href]")	Svi elementi sa href atributom
[*attribute=value*]	$("[href='hm.html']")	Svi elementi sa href="hm.html"
[*attribute!=value*]	$("[href!='hm.html']")	Svi elementi gde nije href= "hm.html"
[*attribute|=value*]	$("[title|='HM']")	Svi elementi gde vrednost atribut title ima prvu **reč** 'HM'
[*attribute~=value*]	$("[title~='hm']")	Svi elementi gde vrednost atributa title sadrži **reč** "hm"
[*attribute^=value*]	$("[title^='hm']")	Svi elementi gde vrednost title atributa počinje **podstring**om 'hm'
[*attribute*=value*]	$("[title*='hm']")	Svi elemetni gde vrednost atributa title ima **podstring** "hm"
[*attribute$=value*]	$("[href$='.jpg']")	Svi elementi čija vrednost href atributa završava **podstring**om ".jpg"

jQuery selektori

Selektor	jQuery sintaksa	Opis selekcije
:input	$(":input")	Svi ulazni elementi
:text	$(":text")	Svi elementi sa type="text"
:password	$(":password")	Svi elementi sa type="password"
:radio	$(":radio")	Svi elementi sa type="radio"
:checkbox	$(":checkbox")	Svi elementi sa type="checkbox"
:submit	$(":submit")	Svi elementi sa type="submit"
:reset	$(":reset")	Svi elementi sa type="reset"
:button	$(":button")	Svi elementi sa type="button"
:image	$(":image")	Svi elementi sa type="image"
:file	$(":file")	Svi elementi sa type="file"
:enabled	$(":enabled")	Svi omogućeni elementi
:disabled	$(":disabled")	Svi onemogućeni elementi
:selected	$(":selected")	Svi selektovani elementi
:checked	$(":checked")	Svi čekirani elementi

jQuery event metode i svojstva

- Metoda bind dodeljuje obrađivač/e događaja elementu:

```
$("p").bind("click", function(){
    alert("The paragraph was clicked.");
});
```

```
$(document).ready(function(){
    $("p").bind("mouseover mouseout", function(){
        $("p").toggleClass("intro");//on-off klase
    });
});
```

```
$(document).ready(function(){
  $("button").bind(                    //event map
    {  click     :function(){ $("p").slideToggle(); } ,
       mouseover:function(){
                    $("body").css("background-color","yellow");
                 } ,
       mouseout :function(){
                    $("body").css("background-color", "white");
                 }
    } );
});
```

jQuery event metode i svojstva

- Slanje parametara metodom bind:

```
<!DOCTYPE html>
<html><head>
<script
src="http://ajax.googleapis.com/ajax/libs/jquery/1.11.2/jquery.min
.js"></script>
<script>
    function handlerName(e){
        alert(e.data.msg); alert(e.data.ojha);
    }
    $(document).ready(function(){
        $("p").bind("click",
                    {msg: "You just clicked me!", ojha:"ojha"},
                    handlerName)
    });
</script>
</head>
<body>
<p>Click me!</p>
</body></html>
```

jQuery event metode i svojstva

`blur()`	Povezuje i reaguje na blur događaj
`change()`	Povezuje i reaguje na change događaj
`click()`	Povezuje i reaguje na click događaj
`dblclick()`	Povezuje i reaguje na double click događaj
`delegate()`	Dodaje handler detetu `$("div").delegate("p", "click",function(){ $("p").css("background-color", "pink"); });`
`die()`	Uklanja sve obrađivače događaja dodatih metodom live()
`error()`	Povezuje i reaguje na error događaj `$("img").error(function(){ $("img").replaceWith("<p>Error!</p>");});`
`event.currentTarget`	Tekući DOM element u bubbling fazi
`event.data`	Sadrži opcionalne podatke koji su prosleđeni metodi koja obrađuje događaj
`event.delegateTarget`	`$("div").on("click", "button", function(event){ $(event.delegateTarget).css("background-color", "pink"); }); //u <div> je <p> i <button>`
`event.isDefaultPrevented()`	Vraća da li je event.preventDefault() bio pozvan za objekat koji generiše događaj.

jQuery event metode i svojstva

`event.isImmediatePropagationStopped()`	Vraća da li je event.stopImmediatePropagation() bio pozvan za objekat koji generiše događaj.
`event.isPropagationStopped()`	Vraća da li je event.stopPropagation() bio pozvan za objekat event.
`event.namespace`	Vraća namespace nastalog događaja.
`event.pageX`	Vraća relativnu poziciju miša prema levoj ivici dokumenta.
`event.pageY`	Vraća relativnu poziciju miša prema gornjoj ivici dokume.
`event.preventDefault()`	Sprečava podrazumevanu akciju za dati događaj.
`event.relatedTarget`	Vraća u koji element se ulazi ili iz koga se izlazi pri pomeranju miša.
`event.result`	Sadrži poslednju/prethodnu vrednost koju vraća event handler aktiviran za dati događaj.
`event.stopImmediatePropagation()`	Sprečava poziv ostalih event handler-a (npr. tri su dodeljena za click).
`event.stopPropagation()`	Sprečava propagaciju događaja za bubbling ka vrhu.
`event.target`	Vraća koji je element izazvao događaj (e.target.nodeName)
`event.timeStamp`	Vraća broj milisekundi od 1.1.1970. do nastanka događaja
`event.type`	Vraća tip nastalog događaja.
`event.which`	Vraća **kod** tastera koji je izazvao događaj.

jQuery event metode i svojstva

`focus()`	Povezuje i reaguje na focus događaj
`focusin()`	Povezuje event handler za focusin događaj
`focusout()`	Povezuje event handler za focusout događaj

`hover()`	Povezuje event handler za hover događaj.
	`$("#p1").hover(`
	`function(){ alert("Ulaz u p1!" );},`
	`function(){ alert("Izlaz iz p1");} );`

`keydown()`	Povezuje i reaguje na keydown događaj
`keypress()`	Povezuje i reaguje na keypress događaj
`keyup()`	Povezuje i reaguje na keyup događaj
`load()`	Povezuje event handler kada je dati element učitan.
`mousedown()`	Povezuje i reaguje na mousedown događaj
`mouseenter()`	Povezuje i reaguje na mouseenter događaj
`mouseleave()`	Povezuje i reaguje na mouseleave događaj

jQuery event metode i svojstva

`mousemove()`	Povezuje i reaguje na mousemove događaj
`mouseout()`	Povezuje i reaguje na mouseout događaj
`mouseover()`	Povezuje i reaguje na mouseover događaj
`mouseup()`	Povezuje i reaguje na mouseup događaj
`off()`	Uklanja event handler-e postavljene sa metodom on() (`$("p").off("click");`)
`on()`	Povezuje event handlers i elemente `$("p").on( "click", function(){ $(this).hide(); } );`
`one()`	Dodaje 1 ili više event handler-a selektovanim elementima. Ovaj hendler može biti trigerovan samo jednom po elementu i događaju.
`$.proxy()`	`$("button").click($.proxy(objPerson, "test"));` //aktivira funkciju test objekta objPerson
`ready()`	Specificira funkciju koja se poziva kada je DOM potpuno učitan
`resize()`	Povezuje i reaguje na resize događaj
`scroll()`	Povezuje i reaguje na scroll događaj
`select()`	Povezuje i reaguje na select događaj

jQuery event metode i svojstva

`submit()`	Povezuje i reaguje na submit događaj
`toggle()`	Smenjuje metode hide i show (uklanjanje i vraćanje elementa)

`trigger()`	Okida događaj vezan za izabrani element.
	`$("input").select(`
	`  function(){ $("input").after("Text!");  });`
	`$("button").click(`
	`  function(){`
	`  $("input").trigger("select");`
	`  });`

`triggerHandler()`	Okida sve funkcije koje su vezane za specifičan događaj za selektovane elemente (ne radi default operaciju npr. selekcije teksta iz prethodnog primera, već smo dodaje "Text!" iza inputa)
`unbind()`	Uklanja sve handlere za selektovane elemente
`undelegate()`	Uklanja event handler-e dodate metodom delegate
`unload()`	Povezuje event handler za unload događaj.

jQuery svojstva

- Slede neka jQuery svojstva:
  - verzija jQuery-a: `var version = $().jquery;`
  - da li se može kreirati XMLHttpRequest objekat preko support svojstva:

```
var moze= jQuery.support.ajax;
```

  - broj elemenata u jQuery objektu, npr. : `var brojli = $("li").length;`
  - promena frame-rate-a (podrazumevano 13ms) :

```
jQuery.fx.interval = 500;
```

  - globalno onemogućavanje/omogućavanje animacije, :

```
$("#true").click(function()  {
                    jQuery.fx.off = true;   } );
$("#false").click(function() {
                    jQuery.fx.off = false;  } );
$("#toggle").click(function(){
                    $("div").toggle("fast");} );
                                  //ili npr. 200
```

Primer

- O(ne)mogućavanje animacije:


```
<!DOCTYPE html>
<html><head><script
src="http://ajax.googleapis.com/ajax/libs/jquery/1.11.2/jquery.min
.js"></script>
<script>
$(document).ready(function(){
    $("#disable").click(function(){ jQuery.fx.off = true;    });
    $("#enable").click( function(){ jQuery.fx.off = false;   });
    $("#toggle").click( function(){ $("div").toggle("fast"); });
});
</script>
</head><body>
<button id="disable">jQuery.fx.off = true ( Disable )</button>
<button id="enable">jQuery.fx.off = false ( Enable )</button>
<br><br>
<button id="toggle">Toggle animation</button>
<div
style="background:yellow;height:100px;width:100px;margin:50px;"></
div></body></html>
```

Razne jQuery metode

- Vezivanje podataka za element i vraćanje vrednosti podatka vezan za element:

```
$("div").data("greeting", "Hello World");
alert($("div").data("greeting"));
```

- Dodeljivanje funkcije za svaki selektovani element:

```
$("button").click(function(){
        $("li").each( function(){alert($(this).text());} );
   });
```

- Uzimanje naziva i vrednosti elementa po datom indeksu traženja:

```
var x = $("p").get(0);
$("div").text(x.nodeName + ": " + x.innerHTML);
// P: tekst prvog (nultog) paragrafa
```

- Uzimanje relativog indeksa datog elementa:

```
$("li").click(  function(){ alert( $(this).index() ); }  );
```

- Prebacivanje jQuery selekcije u niz:

```
var x = $("li").toArray();
for (i = 0; i < x.length; i++) { alert(x[i].innerHTML);    }
```

Razne jQuery methods

- Uklanjanje prethodno dodeljenih podataka:

```
$("div").data("greeting", "Hello World");
$("div").removeData("greeting");
```

- Kreiranje serijalizovane reprezentacije objekta (parovi name-value):

```
$("button").click(function(){
    $("div").text($.param(personObj));
});
```

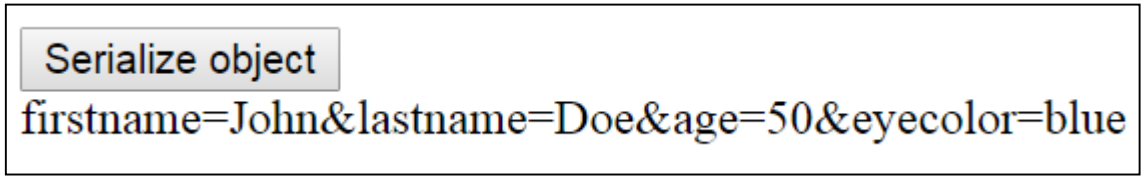
- Specificiranje nove jQuery varijable:

```
var jq = $.noConflict(); //ako ima parameter true, oslobodice
                         //sve jquery varijable
jq(document).ready(function(){
    jq("button").click(function(){   jq("p").hide(1000);    });
});
```

Primer serijalizacije objekta

- Primer serijalizacije objekta:

```
<!DOCTYPE html>
<html><head>
<script
src="http://ajax.googleapis.com/ajax/libs/jquery/1.11.2/jquery.min
.js"></script>
<script>
$(document).ready(function(){
    personObj = new Object();    personObj.firstname = "John";
    personObj.lastname = "Doe";  personObj.age = 50;
    personObj.eyecolor = "blue";
    $("button").click(function(){
        $("div").text($.param(personObj));
    });
});
</script>
</head>
<body>
<button>Serialize object</button>
<div></div>
</body></html>
```



Interfejs: Page Visibility

- Interfejs Page Visibility ima svojstvo za izveštaj o tekućem stanju i događaj koji obaveštava aplikaciju da se stanje promenilo:
  - visibility**State**
    - **svojstvo** koje vraća tekuće stanje vidljivosti dokumenta, vrednosti su: hidden, visible, prerender, unloaded
  - visibility**change**
    - **događaj** koji se aktivira kada se promeni vrednost svojstva visibilityState
- U sledećem primeru funkcija showstate() obrađuje događaj visibilitychange.
- Kada se ova funkcija aktivira na ekranu se prikazuje vrednost svojstva visibilityState.
- Kada se:
  - kartica zameni drugom karticom ili se prozor minimizira vrednost svojstva visibilityState menja se u hidden.
  - kada se kartica ili prozor vrate u prvobitno stanje vrednost visibilityState se vraća na visible.

Interfejs: Page Visibility

```
<!DOCTYPE html>
<html lang="en">
<head>  <meta charset="utf-8"> <title>Page Visibility API</title>
  <script>
    function initiate(){
      document.addEventListener('visibilitychange',
                                showstate);
    }
    function showstate(){
      var element = document.getElementById('application');
      element.innerHTML+='<br>' + document.visibilityState;
    }
    addEventListener('load', initiate);
  </script>
</head>
<body>
  <section id="application">
    Move to another tab or minimize this window to change the
visibility state
  </section>
</body></html>
```

Reprodukcija videa samo ako je dokument vidljiv

```
<!DOCTYPE html>
<html lang="en"><head><meta charset="utf-8"><title>Video see</title>
<script>
  var video;
  function initiate(){
    video = document.getElementById('media');
    document.addEventListener('visibilitychange', showstate);
    video.play();
  }
  function showstate(){
      switch(document.visibilityState){
        case 'visible': video.play();     break;
        case 'hidden' : video.pause();    break;
      }
  }
  addEventListener('load', initiate);
</script>
</head><body>
  <video id="media" width="720" height="400">
    <source src="trailer.mp4">
  </video></body></html>
```

Interfejs: Page Visibility, reagovanje na blur i focus

```
<!DOCTYPE html>
<html lang="en"><head> <meta charset="utf-8"> <title>BF</title>
<script>
 var state;
 function initiate(){
  addEventListener('blur', function(){ changestate('hidden'); });
  addEventListener('focus',function(){ changestate('visible');});
  document.addEventListener('visibilitychange',
    function(){ changestate(document.visibilityState); });
    }
  function changestate(newstate){
      if(state != newstate){ state = newstate;  showstate();  }
  } //sada reaguje i kada se npr. otvara neki fajl
  function showstate(){
      var element = document.getElementById('application');
      element.innerHTML += '<br>' + state;
  }
  addEventListener('load', initiate);
</script>
</head><body><section id="application">    Move to another window
to change the visibility state  </section></body></html>
```

Interfejs: Drag And Drop

- Ideja je da se realizuje prevlačenje elemenata.
- Prevlačenjem polaznog elementa aktivitraju se događaji kao što sledi:
  - dragstart,
    - aktivira se kada počne operacija drag&drop.
  - drag,
    - slično kao mousemove ali ga prilikom prevlačenja aktivira polazni element.
  - dragend,
    - kada se prevlačenje završi, polazni element aktivira ovaj događaj.
- Odredišni element aktivira sledeće događaje:
  - dragenter,
    - aktivira se kada se pokazivač miša tokom prevlačenja nađe u oblasti mogućeg odredišnog elementa
  - dragover,
    - slično kao mouseover s tim što ga aktiviraju mogući određeni elementi tokom prevlačenja

Interfejs: Drag And Drop

- drop,
  - aktivira se kada se element prevuče na odredište,
- dragleave,
  - aktivira se kada pokazivač miša izlazi iz elementa tokom prevlačenja.
  - koristi se sa događajem dragenter (za identifikaciju odredišnog elementa).

```
<!DOCTYPE html>
<html lang="en">
<head>  <meta charset="utf-8">  <title>Drag and Drop</title>
  <link rel="stylesheet" href="dragdrop.css">
  <script src="dragdrop.js"></script>
</head>
<body>
  <section id="dropbox">
    Drag and drop the image here
  </section>
  <section id="picturesbox">
    <img id="image"
       src="slika.gif">
  </section>
</body></html>
```

Interfejs: Drag And Drop

- Datoteka dragdrop.css

```
#dropbox{
  float: left;
  width: 500px;
  height: 300px;
  margin: 10px;
  border: 1px solid #999999;
}
#picturesbox{
  float: left;
  width: 320px;
  margin: 10px;
  border: 1px solid #999999;
}
#picturesbox > img{
  float: left;
  padding: 5px;
}
```

Interfejs: Drag And Drop

- Datoteka dragdrop.js

```
var source1, drop;
function initiate(){
  source1 = document.getElementById('image');
  source1.addEventListener('dragstart', dragged);
  drop = document.getElementById('dropbox');
  drop.addEventListener('dragenter', function(e){
                                        e.preventDefault(); });
  drop.addEventListener('dragover', function(e){
                                        e.preventDefault(); });
  drop.addEventListener('drop', dropped);
}
function dragged(e){
  var code = '<img src="' + source1.getAttribute('src') + '">';
  e.dataTransfer.setData('Text', code);
}
function dropped(e){
  e.preventDefault();
  drop.innerHTML = e.dataTransfer.getData('Text');
}
addEventListener('load', initiate);
```

Interfejs: Drag And Drop

- Izmenjen kod datoteke dragdrop.js

```
var source1, drop;
function initiate(){
  source1 = document.getElementById('image');
  source1.addEventListener('dragstart', dragged);
  source1.addEventListener('dragend', ending);
  drop = document.getElementById('dropbox');
  drop.addEventListener('dragenter', entering);
  drop.addEventListener('dragover', function(e){
                                    e.preventDefault(); });
  drop.addEventListener('dragleave', leaving);
  drop.addEventListener('drop', dropped);
}
function dragged(e){
  var code = '<img src="' + source1.getAttribute('src') + '">';
  e.dataTransfer.setData('Text', code);
}
function ending(e){
  elem = e.target;
  elem.style.visibility = 'hidden';
}
```

Interfejs: Drag And Drop

```
function entering(e){
  e.preventDefault();
  drop.style.background = 'rgba(0, 150, 0, .2)';
}
function leaving(e){
  e.preventDefault();
  drop.style.background = '#FFFFFF';
}
function dropped(e){
  e.preventDefault();
  drop.style.background = '#FFFFFF';
  drop.innerHTML = e.dataTransfer.getData('Text');
}
addEventListener('load', initiate);
```

Prevlačenje više elemenata

- Sledi primer za prevlačenje više elemenata:

```
<!DOCTYPE html>
<html lang="en">
<head>  <meta charset="utf-8">  <title>Drag and Drop</title>
  <link rel="stylesheet" href="dragdrop.css">
  <script src="dragdrop3.js"></script>
</head>
<body>
  <section id="dropbox">
    Drag and drop images here
  </section>
  <section id="picturesbox">
    <img id="image1" src="slika1.gif">
    <img id="image2" sre="slika2.gif">
    <img id="image3" src="slika3.gif">
    <img id="image4" src="slika4.gif">
  </section>
</body>
</html>
```

Prevlačenje više elemenata

```
var drop;

function initiate(){
  var images = document.querySelectorAll('#picturesbox > img');
  for(var i = 0; i < images.length; i++){
    images[i].addEventListener('dragstart', dragged);
  }
  drop = document.getElementById('dropbox');
  drop.addEventListener('dragenter', function(e){
                                        e.preventDefault(); });
  drop.addEventListener('dragover', function(e){
                                        e.preventDefault(); });
  drop.addEventListener('drop', dropped);
}

function dragged(e){
  elem = e.target;
  e.dataTransfer.setData('Text', elem.getAttribute('id'));
}
```

Prevlačenje više elemenata

```
function dropped(e){
  e.preventDefault();
  var id = e.dataTransfer.getData('Text');
  if(id != "image4"){
    var src = document.getElementById(id).src;
    drop.innerHTML = '<img src="' + src + '">';
  }
  else
  {
    drop.innerHTML =
        'Za sliku koju ste odabrali nije dozvoljeno prevlacenje!';
  }
}
addEventListener('load', initiate);
```

- U kodu je dozvoljeno prebacivanje slika čiji id nije "image4".
- Pokušaj prebacivanja slike gde je id = "image4" rezultira ispisivanjem poruke da nije dozvoljeno prebacivanje ove slike.

Prevlačenje elementa sa umanjenim prikazom

- Ideja je da se prilikom prevlačenja elementa prikazuje njegova umanjena verzija.
  - Ovo se izvodi pomoću metode setDragImage koja zadaje sliku i poziciju kursora na slici koja će biti ista za svaku operaciju prevlačenja elementa mišem.

```
<!DOCTYPE html>
<html lang="en">
<head>  <meta charset="utf-8">  <title>Drag and Drop</title>
  <link rel="stylesheet" href="dragdrop.css">
  <script src="dragdrop.js"></script>
</head>
<body>
  <section id="dropbox">
    <canvas id="canvas" width="500" height="300"></canvas>
  </section>
  <section id="picturesbox">
    <img id="image1" src="slika1.gif">
    <img id="image2" src="slika2.gif">
    <img id="image3" src="slika3.gif">
    <img id="image4" src="slika4.gif">
  </section>
</body></html>
```

Prevlačenje elementa sa umanjenim prikazom

```
var drop, canvas;
function initiate(){
  var images = document.querySelectorAll('#picturesbox > img');
  for(var i = 0; i < images.length; i++){
    images[i].addEventListener('dragstart', dragged);
    images[i].addEventListener('dragend', ending);
  }
  drop = document.getElementById('canvas');
  canvas = drop.getContext('2d');
  drop.addEventListener('dragenter', function(e){
                                        e.preventDefault(); });
  drop.addEventListener('dragover', function(e){
                                        e.preventDefault(); });
  drop.addEventListener('drop', dropped);
}
function dragged(e){
  elem = e.target;
  e.dataTransfer.setData('Text', elem.getAttribute('id'));
  e.dataTransfer.setDragImage(elem, 0, 0);
} // za elem stavite neku malu sliku umesto e.target
```

Prevlačenje elementa sa umanjenim prikazom

```
function ending(e){
  elem = e.target;
  elem.style.visibility = 'hidden';
}
function dropped(e){
  e.preventDefault();
  var id = e.dataTransfer.getData('Text');
  var elem = document.getElementById(id);
  var posx = e.pageX - drop.offsetLeft;
  var posy = e.pageY - drop.offsetTop;
  canvas.drawImage(elem, posx, posy);
}
addEventListener('load', initiate);
```

- Dati kod omogućuje prenos svih navednih slika pri čemu se preciznije vidi gde će slika biti postavljena
  - Ovde se pointer odnosi na poziciju 0,0 dakle gornji levi ugao slike.
  - Postavite da je slika koja se prikazuje pri prevlačenju kvadrat stranice 10px.

Prevlačenje datoteka

- Za prevlačenje datoteka koristi se svojstvo files objekta dataTransfer:
  - Ovim svojstvo vraća niz prevučenih datoteka.
  - Informacije koje vraća svojstvo files mogu se smestiti u promenljivu a onda očitati *for* petljom.
  - U primeru koji sledi omogućuje da se u odredišnom elementu prikazuju podaci (naziv, veličina) prevučenih datoteka.

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="utf-8">
  <title>Drag and Drop</title>
  <link rel="stylesheet" href="dragdrop.css">
  <script src="dragdropfile.js"></script>
</head>
<body>
  <section id="dropbox">
    Drag and drop FILES here
  </section>
</body>
</html>
```

Prevlačenje datoteka

- Datoteka dragdropfile.js:

```
var drop;
function initiate(){
  drop = document.getElementById('dropbox');
  drop.addEventListener('dragenter', function(e){
                                        e.preventDefault(); });
  drop.addEventListener('dragover', function(e){
                                        e.preventDefault(); });
  drop.addEventListener('drop', dropped);
}
function dropped(e){
  e.preventDefault();
  var files = e.dataTransfer.files;
  var list = '';
  for(var f = 0; f < files.length; f++){
    list += ':' + files[f].name + ' ' +
                  files[f].size + '<br>';
  }
  drop.innerHTML = list;
}
addEventListener('load', initiate);
```