

INTERNET TEHNOLOGIJE I VEB DIZAJN

jQuery biblioteka

predavač: dr Miloš Antonijević

Uvod u jQuery

- **jQuery** je JavaScript biblioteka pisana u JavaScript-u.
- Osnovna uloga jQuery-ja jeste da olakša upravljanje elementima u DOM strukturi i pruži mogućnost korišćenja naprednih GUI komponenti i funkcionalnosti na Veb stranici.
- Biblioteka se može preuzeti sa <https://jquery.com/download/>
- Preuzeti js fajl se uključuje kao eksterni fajl u zaglavlje HTML stranice kao bilo koji drugi JavaScript fajl.

```
<script type="text/javascript" src="js/jquery-3.6.0.min.js"></script>
```

- Osim navođenja fajla jQuery biblioteke, najčešće se navodi najmanje još jedan JS fajl sa JavaScript kodom koji koristi jQuery biblioteku.

Pristupanje elementima u DOM-u

- jQuery (slično kao CSS) pristupa HTML elementima pomoću selektora.
- Funkcija `$()` pronalazi element(e) DOM strukture u odnosu na selektor koji je naveden kao argument funkcije. Na primer:
 - `$("h1")` – pretraga po tagu (umesto `getElementsByTagName()`); vraća niz pronađenih h1 elemenata
 - `$("#nekiid")` – pretraga po id-u (umesto `getElementById()`); vraća element sa id-em **nekiid**
 - `$(".nazivklase")` – pretraga po nazivu klase (umesto `getElementsByClassName()`); vraća niz elemenata koji sadrže klasu **nazivklase**
- Korisno je primetiti sličnost između načina korišćenja selektora u CSS-u i jQuery-ju. U skladu sa tim, moguća je i kombinacija selektora kao u CSS-u (npr. `$("p#nekiid")` – selektuje paragraf sa id-em **nekiid**; `$("ul li")` – selektuje sve **li** elemente uređene liste.

ready() metoda

- Veb browser-i interpretiraju linije HTML koda redosledom kako su navedene u Veb stranici.
- U slučaju kada bismo napisali JavaScript (jQuery) kod koji manipuliše nekim objektom u DOM strukturi u trenutku dok još nije formirana DOM struktura, manipulacija ne bi bila uspešna.
- Iz tog razloga se koristi jQuery **ready()** metoda **document** elementa koja omogućava izvršavanje koda odmah nakon učitavanja kompletne DOM strukture. Kao argument metode se navodi funkcija koja se želi izvršiti nakon formiranja DOM-a.

```
$(document).ready(function () {  
  
});
```

Korišćenje događaja u jQuery-ju

- U jQuery-ju svaki JavaScript događaj ima odgovarajuću metodu kojom se obrađuje.

```
$("#paragraf").click(function() {  
    $("#input[type=text]").val("Kliknuto");  
});
```

ili

```
$("#paragraf").on("click", function() { ... });
```

- **on()** metoda u okviru sebe omogućava definisanje odgovora na više događaja

```
$("p").on({  
    mouseenter: function(){ $(this).css("background-color", "lightgray"); },  
    mouseleave: function(){ $(this).css("background-color", "lightblue"); },  
    click: function(){ $(this).css("background-color", "yellow"); }  
});
```

hide(), show() i toggle() metode

JavaScript kod:

```
var x = document.getElementById("mojdiv");  
x.style.display = "none";  
//x.style.display = "block";
```

- Za upravljanje **display** stilom elementa jQuery koristi metode elementa hide(), show() i toggle() koje pružaju dodatne mogućnosti usporenog prikazivanja (sakrivanja):

```
<p>Ovo je prvi paragraf</p>  
<p>Ovo je neki drugi paragraf</p>
```

```
$("button").click(function(){  
    $("p").hide(1000);  
});
```

fadeIn() i fadeout() metode

- **fadeIn()** metoda se koristi za postepeno prikazivanje elemenata.
- **fadeOut()** metoda se koristi za postepeno izbleđivanje elemenata (do potpunog sakrivanja)
- Argument obe metode predstavlja broj milisekundi do konačnog pojavljivanja, odnosno nestajanja elementa.

```
$("#div1").fadeIn();  
$("#div2").fadeIn(1500);  
$("#div3").fadeIn(3000);
```

```
<div id="div1" style="width:80px;height:80px;display:none;background-color:red;"></div><br>  
<div id="div2" style="width:80px;height:80px;display:none;background-color:green;"></div><br>  
<div id="div3" style="width:80px;height:80px;display:none;background-color:blue;"></div>
```

slideUp() i slideDown() metode

- Metoda **slideUp()** se koristi da prikaže sakriveni element sa efektom klizanja. Kao argument prihvata broj milisekundi koji određuje trajanje efekta.
- Metoda **slideDown()** se koristi da sakrije vidljivi element sa efektom klizanja. Kao argument prihvata broj milisekundi koji određuje trajanje efekta.

```
$("#panel").slideDown("slow");
```

```
<div id="panel" style="display: none; padding: 50px;">Sadržaj div  
elementa</div>
```

```
$("#panel").slideUp(3000);
```

```
<div id="panel" style="padding: 50px;">Sadržaj div elementa</div>
```


Animiranje elemenata iz jQuery-ja

- Za postizanje efekta animacije elemenata koristi se metoda `animate()`.
- Metoda `animate()` prihvata obavezno jedan, a opciono dva argumenta. Prvi argument predstavlja objekat sa opcijama animacije. Drugi argument predstavlja trajanje animacije, a treći argument predstavlja *callback* funkciju koja će se pozvati po završetku efekta.
- Da bi element mogao da se pomera, vrednost CSS svojstva `position` mora biti promenjena sa `static` (na `relative` ili `absolute`).

```
<div  
style="background:#98bf21;height:100px;width:100px;position:absolute;"></div>  
$("button").click(function(){  
    $("div").animate({left: '250px'});  
});
```

Animiranje elemenata iz jQuery-ja (2)

```
$("#div").animate({  
    left: '250px',  
    opacity: '0.5',  
    height: '150px',  
    width: '150px'});
```

Sukcesivno izvršavanje animacija:

```
$("#div").animate({height: '300px', opacity: '0.4'}, "slow");  
$("#div").animate({width: '300px', opacity: '0.8'}, "slow");  
$("#div").animate({height: '100px', opacity: '0.4'}, "slow");  
$("#div").animate({width: '100px', opacity: '0.8'}, "slow");
```

Pribavljanje i postavljanje vrednosti (sadržaja elementa)

- U slučaju da su u pitanju elementi HTML forme koji imaju atribut **value**, pribavljanje i postavljanje vrednosti elementa se obavlja korišćenjem metode **val()**.
- Ukoliko su u pitanju elementi koji nemaju atribut value, njihov sadržaj se pribavlja i postavlja korišćenjem metoda **text()** i **html()**.
- Ukoliko navedene metode sadrže argument, sadržaj elementa će biti postavljen na vrednost tog argumenta. Ukoliko se ne prosledi argument, metode vraćaju vrednost atributa value, odnosno teksta elementa.
- Razlika između text() i html() metode je u tome što metoda text() vraća samo tekstualni deo sadržaja elementa, a metoda html() vraća sadržaj zajedno sa postojećim HTML tagovima.
- Metoda **attr()** se koristi za pribavljanje i postavljanje vrednosti atributa elementa.

```
<p id="test">Ovo <i>je</i> neki <b>formatiran</b> tekst.</p>
```

```
alert("Tekst: " + $("#test").text());
```

```
alert("HTML: " + $("#test").html());
```

Dodavanje elemenata (ili sadržaja) iz jQuery-ja

- Dodavanje elemenata u DOM strukturu iz jQuery-ja se obavlja korišćenjem četiri metode:
 - `append()` – Dodaje sadržaj (ili novi element) na kraju odabranog elementa (kao dete).
 - `prepend()` – Dodaje sadržaj na kraju početak odabranog elementa (kao dete).
 - `after()` – Dodaje sadržaj nakon odabranog elementa (kao srodnika).
 - `before()` – Dodaje sadržaj ispred odabranog elementa (kao srodnika).

```
<ul>
```

```
    <li>Stavka 1</li>  <li>Stavka 2</li>  <li>Stavka 3</li>
```

```
</ul>
```

```
$("ul").append("<li>Dodata stavka</li>");
```

```
// Dodavanje elemenata preko DOM strukture
```

```
var txt1 = document.createElement("p");
```

```
txt1.innerHTML = "Sadržaj paragrafa";
```

```
$("body").append(txt1);
```

Brisanje elemenata iz jQuery-ja

- Brisanje postojećih elemenata se obavlja korišćenjem dve metode `remove()` i `empty()`.
- Metoda **`remove()`** briše odabrani element i njegovu decu iz DOM-a.
- Metoda **`empty()`**, briše svu decu odabranog elementa.

```
<div id="div1" style="height:100px;width:300px;border:1px solid black;background-color:yellow;">
```

Ovo je tekst div-a

```
<p>Prvi paragraf</p>
```

```
<p>Drugi paragraf</p>
```

```
</div>
```

```
$("#div1").remove();
```

```
$("#div1").empty();
```

Dodavanje i brisanje klasa i izmena CSS svojstava

- Dodavanje klasa u jQuery-ju se vrši korišćenjem metode **addClass()**, koja dodaje klase navedene u argumentu. Nazivi klasa u argumentu se odvajaju razmakom (space).

```
$("#nekidiv").addClass("klasa1 klasa2");
```

- Uklanjanje klasa iz elemenata se obavlja korišćenjem **removeClass()** metode, koja briše klase navedene u argumentu.

```
$("#h1, h2, p").removeClass("blue");
```

- Pribavljanje i izmena CSS stilova se vrši upotrebom **css()** metode. Ukoliko se metoda navede sa jednim argumentom, ona vraća vrednost navedenog CSS svojstva. Ukoliko se proslede dva argumenta, metoda postavlja svojstvo navedeno u prvom argumentu na vrednost navedenu u drugom argumentu.

```
$("#p").css("background-color", "yellow");
```

Pronalaženje roditelja, srodnika i dece u DOM strukturi

- Pronalaženje direktnog roditelja elementa se vrši uz pomoć metode **parent()**.
- Za pronalaženje **svih** roditelja elementa koristi se metoda **parents()**.
`$("#nekidiv").parents().css({"color": "red", "border": "2px solid red"});`
- Pronalaženje dece prvog kolena (neposredne dece) obavlja se korišćenjem metode **children()**.
- Pronalaženje sve dece, bez obzira na dubinu, vrši se korišćenjem metode **find()**.
- Metode **children()** i **find()** mogu sadržati argumente u kojima se navode jQuery selektori.
`$("#nekidiv").find("span").css({"color": "red", "border": "2px solid red"});`
- Pronalaženje srodnika se vrši upotrebom metoda **siblings()** – svi srodnici, **next()** – naredni srodnik, **prev()** – prethodni srodnik.

Primer checkbox, select

```
$('#input[type="checkbox"]').click(function(){  
    if($(this).prop("checked") == true) { console.log("Checkbox is checked."); }  
    else { console.log("Checkbox is unchecked."); }  
});
```

```
<p><input type="checkbox"> Probaj me!</p>  
<div id="result" style="background: yellow;"></div>  
// Primer sa SELECT  
<select id="myselect">  
    <option value="1">Mr</option>  
    <option value="2">Mrs</option>  
    <option value="3">Ms</option>  
</select>  
$("#myselect option:selected").text();
```


PITANJA?