

ПРОГРАМИРАЊЕ ВЕБ АПЛИКАЦИЈА

Nastavna jedinica: 10

jQuery i AJAXs



JQUERY BIBLIOTEKA

- ▶ jQuery je besplatna i jedna od najmoćnijih JavaScript biblioteka.
- ▶ Upotreba ove biblioteke olakšava ispunjavanje čestih zahteva koji se stavljaju pred programera u svakodnevnom radu (rad sa DOM-om, AJAX, animacija i dr.)
- ▶ jQuery olakšava posao prilagođavanja koda različitim web pretraživačima i dozvoljava programeru da više vremena posveti samom proizvodu nego njegovoj kompatibilnosti
- ▶ Sintaksa jQuery biblioteke olakšava rad, pregledna je i kompaktna

KAKO JQUERY OLAKŠAVA PROGRAMIRANJE

- ▶ Koristi CSS sintaksu za selektovanje elemenata
- ▶ Dizajniran je tako da lako može manipulirati setovima elemenata
- ▶ Omogućava povezivanje koda u lanac izvršavanja (statement chaining)
- ▶ jQuery je veoma proširiv kroz upotrebu plugin-ova

IMPLEMENTACIJA JQUERY BIBLIOTEKE

- ▶ JQuery možemo shvatiti kao gomilu JavaScript koda, koji je grupisan. Da bismo mogli da koristimo ovu biblioteku, potrebno je da na određeni način učitamo njen sadržaj u stranicu. Dodavanje jquery biblioteke na stranicu možemo uraditi na dva načina:
- ▶ Preuzimanjem jquery biblioteke sa zvaničnog sajta jquery.com
- ▶ Uključivanjem jquery biblioteke preko CDN (content delivery network ili content distribution network)

IDEJA JQUERY BIBLIOTEKE

- ▶ JQuery je kompleksna biblioteka koja je objavljena 2006.godine. JQuery je biblioteka koja ima implemetirane JavaScript funkcionalnosti. Ove funkcionalnosti se izvršavaju u pretraživaču na klijentskoj strani. Glavna ideja Johna Resiga, tvorca JQuery biblioteke, bila je da se osposobi interakcija sa elementima JavaScripta pomoću CSS selektora. To je demonstriralo novu ideju koja je formirala početke biblioteke kakvu znamo danas. Od trenutka kada je objavljen JQuery je veoma brzo postao prilično rasprostranjen na internetu. Od tog trenutka JQuery je nastavio da se razvija, da bi prva verzija bila završena par meseci nakon objavljivanja same biblioteke.

IDEJA JQUERY BIBLIOTEKE

- U ranim danima web-a nisu postojali jasni standardi JavaScript implementacije. To je dovelo do toga da su veb pretraživači implementirali JavaScript na različite načine, što je developerima uzrokovalo mnogobrojne probleme. Kao posledica ovakve implementacije, programeri su morali da JavaScript kod pišu za različite web pretraživače. Uvođenjem jQuery biblioteke ovaj problem je umnogome rešen jer je postojala jedinstvena sintaksa koja je obezbeđivala funkcionalanost koda na svim modernim pretraživačima.

IDEJA JQUERY BIBLIOTEKE

- Za rad sa JQuery bibliotekom neophodno je osnovno poznavanje DOM modela. Kada je stranica učitana, web pretraživač generiše reprezentaciju sadržaja te stranice. Za svaki element date stranice, web pretraživač generiše jedan ili više čvorova (nodes), koji predstavljaju elemente te stranice. Prilikom rada sa JavaScriptom, DOM može da prouzrokuje probleme, ali pomoću JQuery-ja ovo lako može da se izbegne

READY I ONLOAD

- Ready nije isto što i onload jer se okida mnogo ranije nego onload koji čeka da se čitav sadržaj dokumenta učitа na stranu uključujući i same slike koje se tu nalaze.

READY

- ▶ Da bi se jQuery uspešno izvršavao potrebno je da dokument bude „spreman“ za rad.
- ▶ Da bi se to osiguralo sve funkcije se smeštaju u

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

```
.....
```

```
});
```

Ili njen skraćeni oblik

```
$(function (){
```

```
.....
```

```
});
```

OSNOVNA SINTAKSA

- ▶ Osnovna sintaksa jQuery instrukcije je:

`$(selector:filter).action()`

- ▶ `$(this).hide()` – sakriva tekući element.
- ▶ `$("p:even").hide()` – sakriva sve parne `<p>` elemente.
- ▶ `$(".test").hide()` – sakriva sve elemente sa `class="test"`.
- ▶ `$("#test").hide()` – sakriva sve elemente sa `id="test"`.

JQUERY SELEKTORI I FILTERI

- ▶ Selektori i filteri su jedni od osnovnih blokova za građenje jQuery koda
- ▶ Selektori i filteri se koriste da bi se izvukle informacije iz Web strane
- ▶ Selektori u jQuery biblioteci se služe istom sintaksom kao i CSS što je jedna od prvih ideja za kreiranje ovakve biblioteke
- ▶ Filteri se koriste za dodatnu pretragu nad rezultatom koji vrate selektori (**p**, **.selector**, **#intro**, **p:first**, **h2:not(.selector)**)

JQUERY SELEKTORI I FILTERI

► Osnovni selektori

Selektor	Rezultat
<code>\$ („naziv_taga“)</code>	Selektuje sve tagove navedenog naziva
<code>\$ („#id_elementa“)</code>	Selektuje element sa navedenim id-jem
<code>\$ („.naziv_klase“)</code>	Selektuje sve elemente navedene klase
<code>\$ (" [attribute] ")</code>	Selektuje sve elemente sa atributom <i>attribute</i>

JQUERY SELEKTORI I FILTERI

► Osnovni jQuery filteri

Filter	Selektuje....
:first, :last	Prvi ili poslednji element nad kojim je pozvan
:even, :odd	Parne ili neparne elemente nad kojim je pozvan
:gt(), :lt(), :eq()	Elemente čiji je indeks: veći od, manji od, jednak
:animated	Elemente koji su u procesu animacije (koristi se kada se želi zaustaviti animacija u toku)
:focus	Element koji je trenutno u fokusu
:not(expr)	Elemente koji ne odgovaraju navedenom izrazu

KREIRANJE I EDITOVANJE SADRŽAJA STRANE

- ▶ jQuery može da kreira novi sadržaj na strani ali i da menja osobine postojećim elementima
 - ▶ `$ („<p></p>“)`
 - ▶ `$ („<div>Sadržaj div-a</div>“)`
- ▶ Metode za pozicioniranje novog sadržaja selektovanom elementu:
 - ▶ `append()`
 - ▶ `prepend()`
 - ▶ `before()`
 - ▶ `after()`
- ▶ Metode za dodavanje novog HTML ili text sadržaja elementu:
 - ▶ `.html()`
 - ▶ `.text()` – Napomena: Prosleđeni sadržaj neće biti tumačen kao HTML
 - ▶ `.val()` – Samo input elementi

UPRAVLJANJE ATRIBUTIMA

- ▶ `.attr()`
- ▶ `.removeAttr()`

```
$("#mojelemenat").attr("id", "novi_id");
```

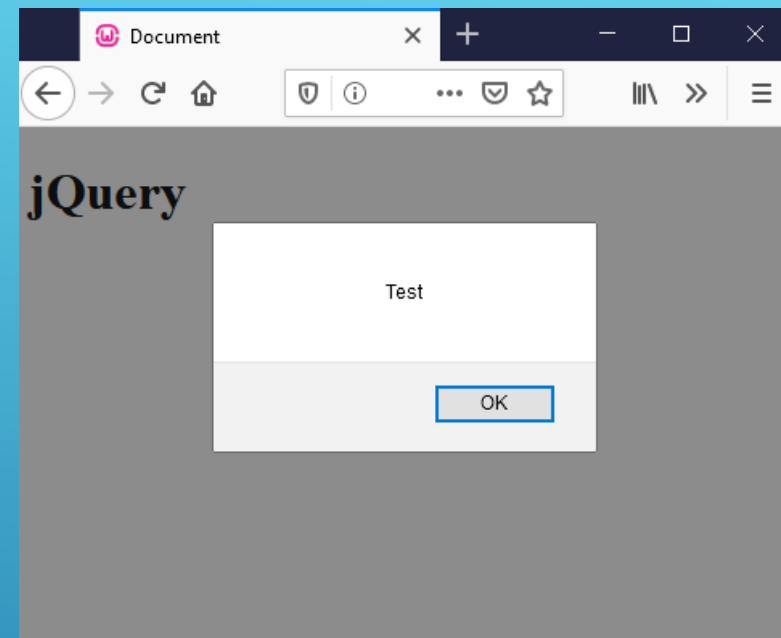
UGRAĐIVANJE PRILAGOĐENIH DATA SVOJSTAVA

- ▶ Prilagođeni data atributi omogućavaju kreiranje i upravljanje posebnim vrstama podataka koji se odnose na element
- ▶ Ovi podaci mogu biti smešteni u vidu ključ-vrednost parova ili u formi objekta
- ▶ Funkcija koja se u ovu svrhu koristi je `.data()`
- ▶ Sintaksa za setovanje prilagođenih podataka:
 - ▶ `.data(key, value)`
 - ▶ `.data(obj)`
- ▶ Sintaksa za dobavljanje prilagođenih podataka:
 - ▶ `.data(key)`
 - ▶ `.data()` //vraća JavaScript objekat
- ▶ Sintaksa za uklanjanje prilagođenih podataka:
 - ▶ `.removeData(key)`
 - ▶ `.removeData()`

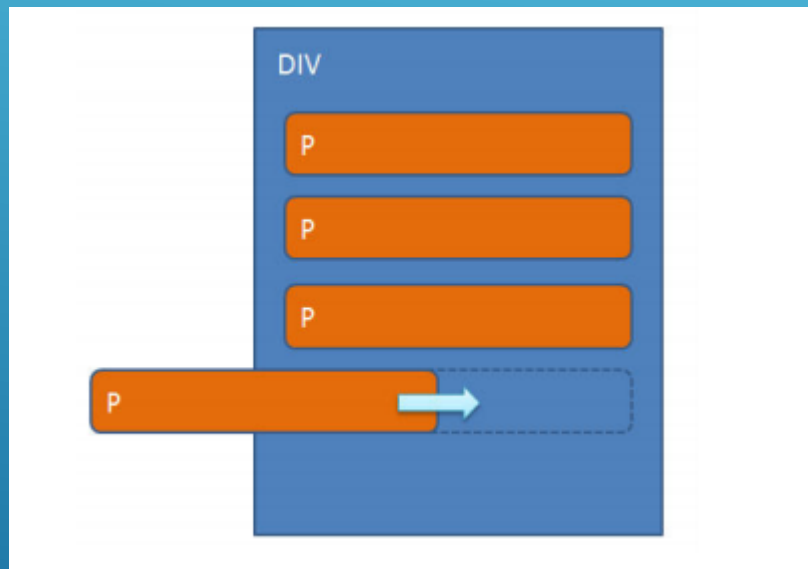
PROVERA ISPRAVNOSTI

```
<script src="jquery-3.4.1.js"></script>
<script>
    $(document).ready(function() {
        alert("Test");
    })
</script>
```

```
<script src="https://code.jquery.com/jquery-3.4.1.js"></script>
<script>
    $(document).ready(function() {
        alert("Test");
    })
</script>
```



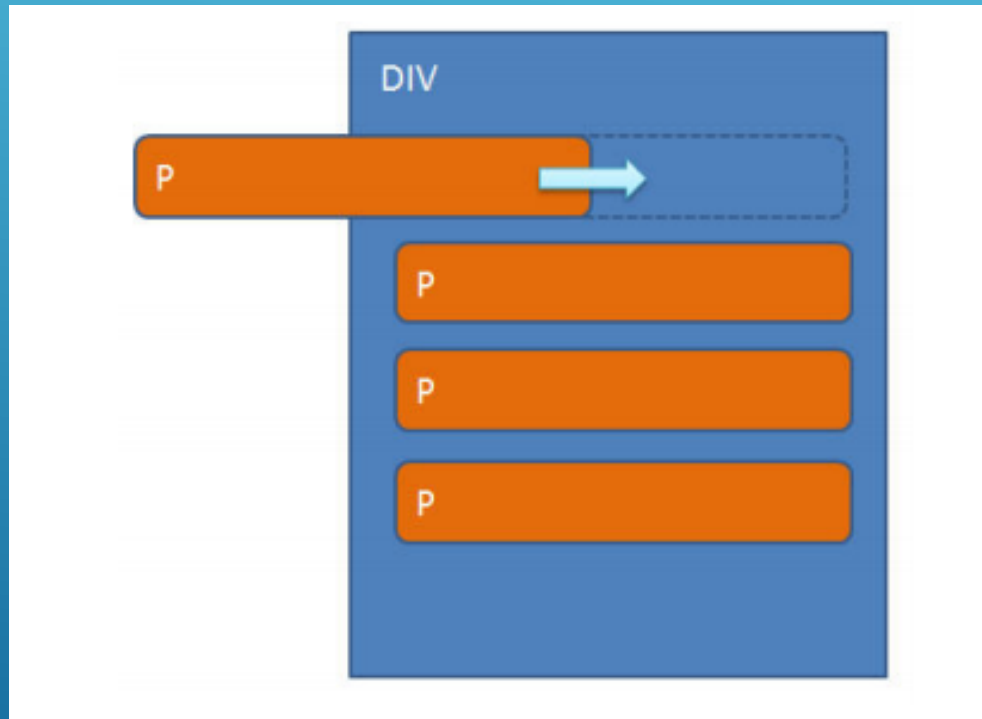
.APPEND()



.APPEND()

```
<!doctype html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>Document</title>
</head>
<body>
<h1>jQuery</h1>
<div>
  <input type="button" value="Create element"/>
  <div id="div1">
    <p class="p1">Ovo je text 1</p>
    <p class="p1">Ovo je text 2</p>
    <p class="p1">Ovo je text 3</p>
  </div>
</div>
</body>
</html>
<script src="jquery-3.4.1.js"></script>
<script>
  $(document).ready(function() {
    $("input").click(function() {
      var a=$("<p></p>").text("Tekst iz jQuery-ja");
      a.attr("id", "novi_id");
      $("#div1").append(a);
    });
  })
</script>
```

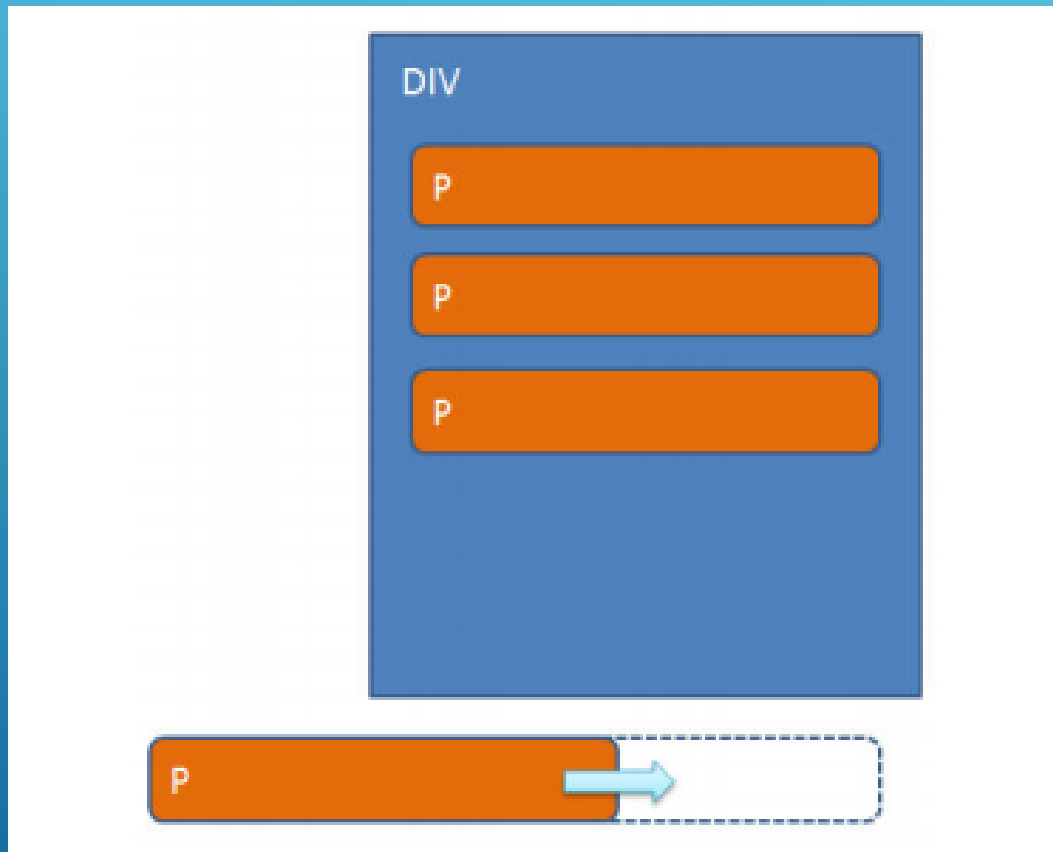
.PREPEND()



.PREPEND()

```
<!doctype html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>Document</title>
</head>
<body>
<h1>jQuery</h1>
<div>
  <input type="button" value="Create element"/>
  <div id="div1">
    <p class="p1">Ovo je text 1</p>
    <p class="p1">Ovo je text 2</p>
    <p class="p1">Ovo je text 3</p>
  </div>
</div>
</body>
</html>
<script src="jquery-3.4.1.js"></script>
<script>
  $(document).ready(function() {
    $("input").click(function() {
      var a=$("<p></p>").text("Tekst iz jQuery-ja");
      a.attr("id", "novi_id");
      $("#div1").prepend(a);
    });
  })
</script>
```

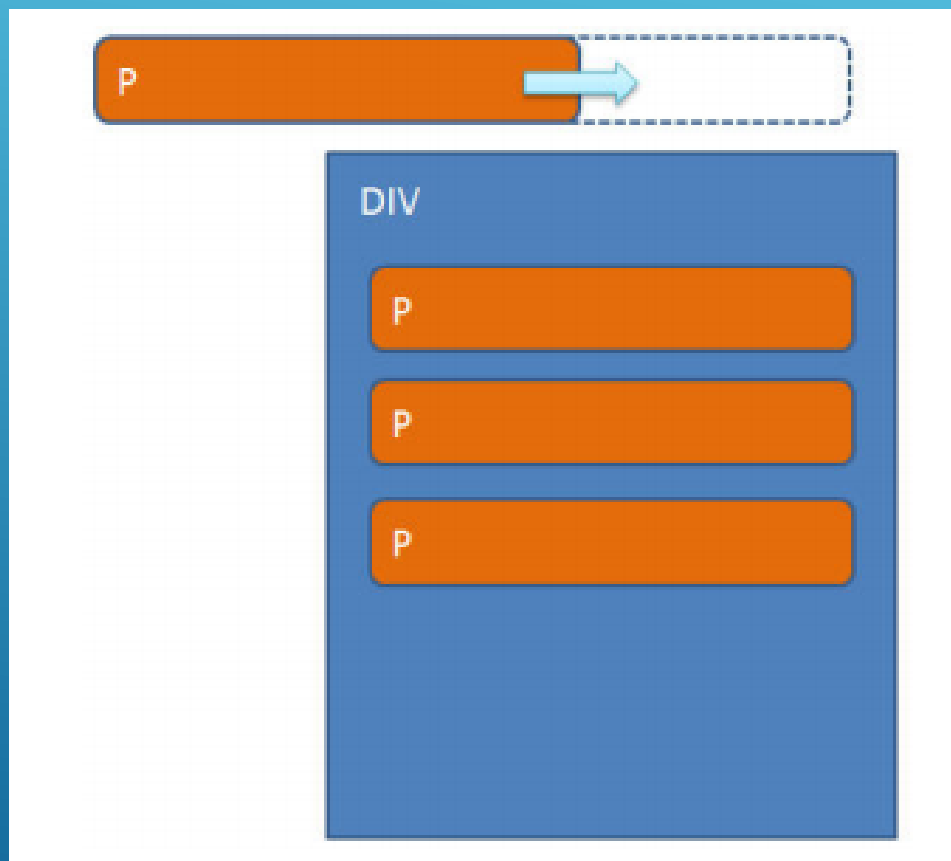
.AFTER()



.AFTER()

```
<!doctype html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>Document</title>
</head>
<body>
<h1>jQuery</h1>
<div>
  <input type="button" value="Create element"/>
  <div id="div1">
    <p class="p1">Ovo je text 1</p>
    <p class="p1">Ovo je text 2</p>
    <p class="p1">Ovo je text 3</p>
  </div>
</div>
</body>
</html>
<script src="jquery-3.4.1.js"></script>
<script>
  $(document).ready(function() {
    $("input").click(function() {
      var a=$("<p></p>").text("Tekst iz jQuery-ja");
      a.attr("id", "novi_id");
      $("#div1").after(a);
    });
  })
</script>
```

.BEFORE()

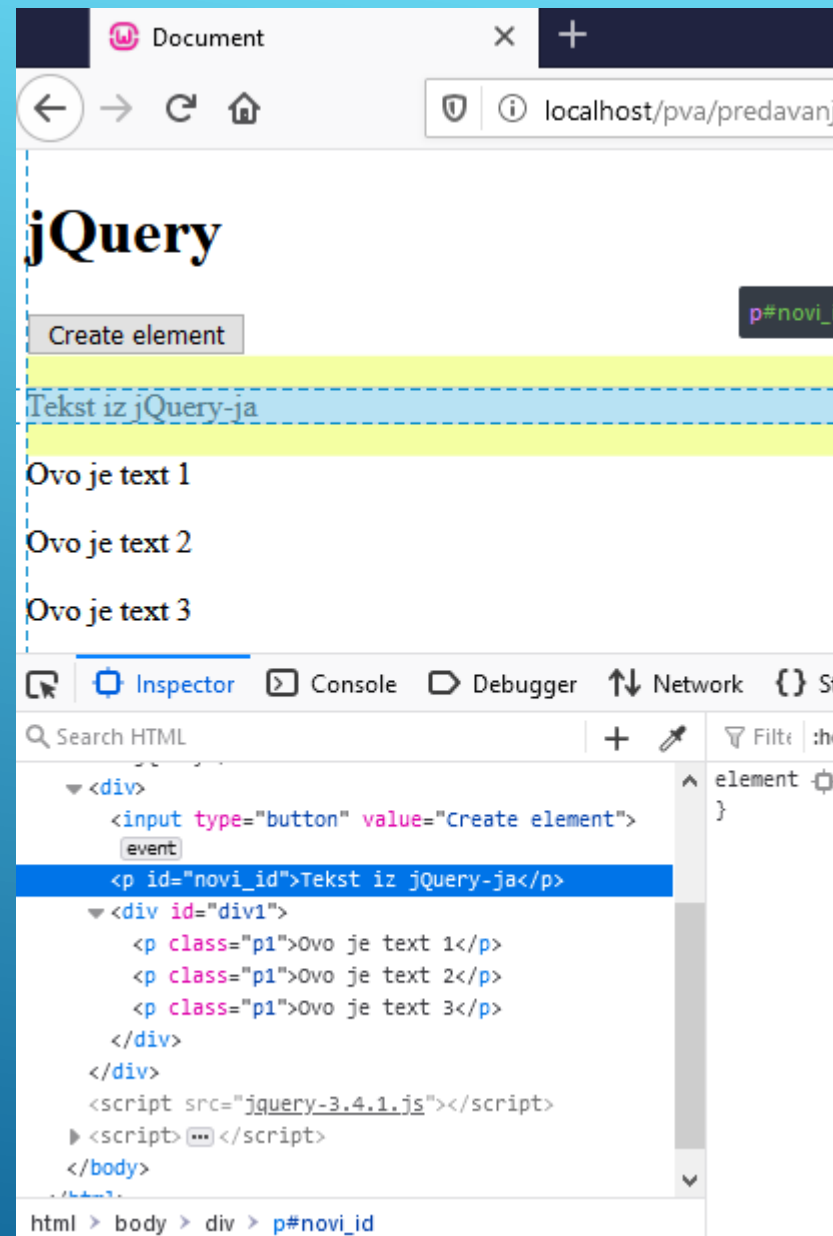


.BEFORE()

```

<!doctype html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>Document</title>
</head>
<body>
<h1>jQuery</h1>
<div>
  <input type="button" value="Create element"/>
  <div id="div1">
    <p class="p1">Ovo je text 1</p>
    <p class="p1">Ovo je text 2</p>
    <p class="p1">Ovo je text 3</p>
  </div>
</div>
</body>
</html>
<script src="jquery-3.4.1.js"></script>
<script>
  $(document).ready(function() {
    $("input").click(function() {
      var a=$("<p></p>").text("Tekst iz jQuery-ja");
      a.attr("id", "novi_id");
      $("#div1").before(a);
    });
  });
</script>

```



UPRAVLJANJE DOGAĐAJIMA

- ▶ Metoda **.on()** se koristi za postavljanje osluškivača za događaj nad elementom. Prvi parametar ove metode je naziv događaja, a drugi je callback funkcija koja će biti pozvana onog trenutka kada se događaj nad elementom detektuje
- ▶ Uklanjanje osluškivača za događaj se vrši uz pomoću **.off()** metode

PRIKAZ DOGAĐAJA

Mouse Events	Keyboard Events	Form Events	Document/Window Events
click	keypress	submit	load
dblclick	keydown	change	resize
mouseenter	keyup	focus	scroll
mouseleave		blur	unload

UPRAVLJANJE DOGAĐAJIMA

- ▶ Metoda **.on()** se koristi za postavljanje osluškivača za događaj nad elementom. Prvi parametar ove metode je naziv događaja, a drugi je callback funkcija koja će biti pozvana onog trenutka kada se događaj nad elementom detektuje
- ▶ Uklanjanje osluškivača za događaj se vrši uz pomoću **.off()** metode

UPRAVLJANJE DOGAĐAJIMA

```
<script>
    $(document).ready(function() {
        $("p").on("click", function() {
            alert("Neki paragraf je kliknut")
        })
    })
</script>
```

```
<script>
    $(document).ready(function() {
        $("p").click(function() {
            alert("Neki paragraf je kliknut")
        })
    })
</script>
```

.EACH()

- ▶ Vrlo korisna.
- ▶ Koristi se za jednostavnu izmenu svojstava više elemenata

```
$ ( document ) .ready ( function () {  
    var margina=3;  
    var border=2;  
    $ ( "p" ) .each ( function (index, element) {  
        $ (element) .css ( "border", border + "px solid red" )  
        .css ( "margin-left", margina );  
        border+=2;  
        margina+=10;  
    } ) ;  
} ) ;
```

.HTML(), .TEXT(), .VAL()

- ▶ Vrednosti elemenata se mogu pročitati korišćenjem ove tri metode
 - ▶ .html() – vraća html kod
 - ▶ .text() – vraća samo tekst
 - ▶ .val() – vraća vrednost INPUT elementa

.HTML(), .TEXT(), .VAL()

```
<!doctype html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>Document</title>
</head>
<body>
  <h1>jQuery</h1>
  <p id="test">This is some <b>bold</b> text in a paragraph.</p>
  <button id="btn1">Show Text</button>
  <button id="btn2">Show HTML</button>
</body>
</html>
<script src="jquery-3.4.1.js"></script>
<script>
  $(document).ready(function() {
    $("#btn1").click(function() {
      $(this).text("Kliknut");
      alert("Text: " + $("#test").text());
    });
    $("#btn2").click(function() {
      $(this).text("Kliknut");
      alert("HTML: " + $("#test").html());
    });
  });
</script>
```

.CSS()

- ▶ Ova metoda omogućava izmenu css svojstava selektovanog objekta
- ▶ Korišćenjem .css() metode mogu se dohvatiti ili postaviti sva svojstva definisana cascade style sheet-om

.CSS()

- ▶ Čitanje svojstva

`$(„selektor“).css("propertyname");`

- ▶ Postavljanje svojstva

`$(„selektor“).css("propertyname","value");`

.CSS()

```
<!doctype html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>Document</title>
</head>
<body>
  <h1>jQuery</h1>
  <button>Pomeri div</button>
  <div style="background:#98bf21;height:100px;width:100px;position:absolute;"></div>
</body>
</html>
<script src="jquery-3.4.1.js"></script>
<script>
  $(document).ready(function() {
    $("button").click(function() {
      $("div").css({left: '250px'});
    });
  })
</script>
```

UPRAVLJANJE KLASAMA

- ▶ `.addClass()`
- ▶ `.removeClass()`
- ▶ `.toggleClass()`

UPRAVLJANJE KLASAMA

```
<!doctype html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>Document</title>
  <link rel="stylesheet" href="klase.css"/>
</head>
<body>
  <h1>jQuery</h1>
  <h2>Naslov 2</h2>
  <p>Ovo je paragraf.</p>
  <p>Ovo je drugi paragraf.</p>
  <div>Vaaaaaaaaaazan tekst!</div>
  <br>
  <button>Dodaj klase</button>
</body>
</html>
<script src="jquery-3.4.1.js"></script>
<script>
  $(document).ready(function() {
    $("button").click(function() {
      $("h1, h2, p").addClass("blue");
      $("div").addClass("important");
    });
  });
</script>
```

```
.important {
  font-weight: bold;
  font-size: xx-large;
}
.blue {
  color: blue;
}
```

UVOD U JQUERY ANIMACIJU

- ▶ Moderne web aplikacije su bogate različitim atraktivnim animacijama i efektima. Ove animacije veoma često su posledica delovanja upravo jQuery biblioteke.
- ▶ Danas postoji mnoštvo plugin-ova koji olakšavaju kreiranje animacija.
- ▶ Iako sve ono što možemo uraditi uz pomoć jQuery biblioteke možemo uraditi i bez nje, upotreba jQuery-a pisanje koda čini lakšim i brži.
- ▶ U samo nekoliko linija koda moguće je postići prilično atraktivne efekte.

OSNOVNI EFEKTI ANIMACIJE

- ▶.show()
- ▶.hide()
- ▶.toggle()
- ▶.animate()

OSNOVNI EFEKTI ANIMACIJE

```
<!doctype html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>Document</title>
  <link rel="stylesheet" href="klase.css"/>
</head>
<body>
  <h1>jQuery</h1>
  <p>This is a paragraph.</p>
  <button class="btn1">Hide</button>
  <button class="btn2">Show</button>
</body>
</html>
<script src="jquery-3.4.1.js"></script>
<script>
  $(document).ready(function() {
    $(".btn1").click(function() {
      $("p").hide(1000);
    });
    $(".btn2").click(function() {
      $("p").show(1000);
    });
  })
</script>
```

OSNOVNI EFEKTI ANIMACIJE

```
<!doctype html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>Document</title>
  <link rel="stylesheet" href="klase.css"/>
</head>
<body>
  <h1>jQuery</h1>
  <button class="btn1">Animate</button>
  <div style="background:#98bf21;height:100px;width:100px;position:absolute;"></div>
</body>
</html>
<script src="jquery-3.4.1.js"></script>
<script>
  $(document).ready(function() {
    $("button").click(function() {
      $("div").animate({left: '250px', opacity: '0.5', height: '150px', width: '150px'}, "slow");
    });
  });
</script>
```

FADDING EFEKTI

- ▶ `.fadeIn()`
- ▶ `.fadeOut()`
- ▶ `.fadeTo(1000, 0.3)`

FADDING EFEKTI

```
<!doctype html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>Document</title>
  <link rel="stylesheet" href="klase.css"/>
</head>
<body>
  <h1>jQuery</h1>
  <p>This is a paragraph.</p>
  <button class="btn1">Fade out</button>
  <button class="btn2">Fade in</button>
</body>
</html>
<script src="jquery-3.4.1.js"></script>
<script>
  $(document).ready(function() {
    $(".btn1").click(function() {
      $("p").fadeOut(2000)
    });
    $(".btn2").click(function() {
      $("p").fadeIn();
    });
  })
</script>
```

SLIDING EFEKTI

- ▶ `.slideUp()`
- ▶ `.slideDown()`
- ▶ `.slideToggle()`

SLIDING EFEKTI

```
<!doctype html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>Document</title>
  <link rel="stylesheet" href="klase.css"/>
</head>
<body>
  <h1>jQuery</h1>
  <p>This is a paragraph.</p>
  <button class="btn1">Slide Up</button>
  <button class="btn2">Slide Down</button>
</body>
</html>
<script src="jquery-3.4.1.js"></script>
<script>
  $(document).ready(function() {
    $(".btn1").click(function() {
      $("p").slideUp(2000)
    });
    $(".btn2").click(function() {
      $("p").slideDown();
    });
  })
</script>
```

JEDNOSTAVNA ANIMIRANA GALERIJA

```
<!doctype html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>Document</title>
  <link rel="stylesheet" href="klase.css"/>
</head>
<body>
  <h1>jQuery</h1>
  <div style="align: center; width: 100%">
    <br>
    
    
    
    
  </div>
</body>
</html>
<script src="jquery-3.4.1.js"></script>
<script>
  $(document).ready(function() {
    $("img").click(function() {
      $("#glavna")
        .hide()
        .stop()
        .attr("src", $(this).attr("src"))
        .css({"opacity": "0"})
        .show()
        .animate({"opacity": "1"}, 2000);
    });
  });
</script>
```

ASINHRONO UPRAVLJANJE SERVERSKIM RESURSIMA POMOĆU JAVASCRIPT-A

- ▶ Може се остварити на minimum три načina
 - ▶ **A**synchronous JavaScript **A**nd **X**ML
 - ▶ Long Pooling
 - ▶ WebSocket

ŠTA JE AJAX?

- ▶ AJAX je skraćenica od: Asynchronous JavaScript and XML
- ▶ Ovo je nešto novija tehnologija kreiranja boljih, bržih i interaktivnijih web aplikacija.
- ▶ AJAX koristi HTML za sadržaj, CSS za stilizaciju kao i DOM i JavaScript za dinamičko upravljanje sadržajem.
- ▶ Konvencionalne web aplikacije razmenjuju informacije sa serverom koristeći se sinhronim zahtevima.
- ▶ Uz pomoć AJAX-a moguće je izvršiti komunikaciju sa severom tako da klijent ne primeti da je do bilo kakve komunikacije došlo (submit).

ŠTA JE AJAX?

- ▶ XML je često korišćen tip podatka koji se dobavlja od servera, ali ne i jedini. Pored XML-a koristi se JSON, HTML pa čak i regularan tekst.
- ▶ AJAX je tehnologija koja je nezavisna od softwera na serveru.
- ▶ Korisnik može da nastavi sa korišćenjem aplikacije dok AJAX vrši komunikaciju sa serverom.
- ▶ Aktivacija AJAX-a može biti pokrenuta nakon registracije događaja koji je prirodan i intuitivan za korisnika.

AJAX

- ▶ Standard u JavaScriptu danas
- ▶ Koristi se za preuzimanje podataka sa servera na zahtev klijenta
- ▶ Najčešće se dovodi u blisku vezu sa klasom **XMLHttpRequest**

AJAX I OPEN STANDARDS

- ▶ AJAX je baziran na otvorenim standardima (Open Standards):
 - ▶ Prezentacija podataka je bazirana na samom pretraživaču i upotrebi HTML-a i CSS-a.
 - ▶ Podaci su smešteni u XML (JSON, HTML, text) formatu i prikupljeni sa servera.
 - ▶ Podaci se pribavljaju sa servera u pozadini uz pomoć **XMLHttpRequest** objekta.
 - ▶ Čitav proces pokreće JavaScript

PODRŠKA ZA AJAX

- ▶ Napomena: AJAX ne podržavaju svi pretraživači.
- ▶ Lista najpoznatijih WEB pretraživača koji podržavaju AJAX:
 - ▶ Mozilla Firefox 1.0 i noviji.
 - ▶ Netscape version 7.1 i noviji.
 - ▶ Apple Safari 1.2 i noviji.
 - ▶ Microsoft Internet Explorer 5 i noviji.
 - ▶ Konqueror.
 - ▶ Opera 7.6 i noviji.
- ▶ Prilikom kreiranja aplikacija, imati u vidu pretraživače koji ne podržavaju AJAX.
- ▶ Kada se kaže da pretraživač ne podržava AJAX to znači da pretraživač jednostavno nije u stanju da kreira **XMLHttpRequest** objekat.

KORACI

- ▶ Klijent registruje događaj.
- ▶ jQuery kreira XMLHttpRequest objekat.
- ▶ jQuery konfiguriše objekat.
- ▶ jQuery upućuje asinhroni zahtev ka Web serveru.
- ▶ Web server procesira zahtev i vraća rezultat.
- ▶ jQuery poziva callback() funkciju i procesira rezultat.
- ▶ HTML DOM se ažurira novim podacima.

JQUERY METODE

- ▶.ajax()
- ▶.load()
- ▶.get()
- ▶.post()

.AJAX()

- ▶ Osnovna metoda za pozivanje asinhronе komunikacije
 - ▶ Najsloženija
 - ▶ Sve ostale metode su pojednostavljena metoda `.ajax()`
 - ▶ Sintaksa `.ajax()` metode
- `$.ajax({name:value, name:value, ... })`**

.AJAX() – NEKI OD PARAMETARA

Name	Value/Description
<code>beforeSend(xhr)</code>	A function to run before the request is sent
<code>cache</code>	A Boolean value indicating whether the browser should cache the requested pages. Default is true
<code>complete(xhr,status)</code>	A function to run when the request is finished (after success and error functions)
<code>contentType</code>	The content type used when sending data to the server. Default is: "application/x-www-form-urlencoded"
<code>context</code>	Specifies the "this" value for all AJAX related callback functions
<code>data</code>	Specifies data to be sent to the server
<code>dataType</code>	The data type expected of the server response.
<code>error(xhr,status,error)</code>	A function to run if the request fails.
<code>processData</code>	A Boolean value specifying whether or not data sent with the request should be transformed into a query string. Default is true
<code>success(result,status,xhr)</code>	A function to be run when the request succeeds
<code>timeout</code>	The local timeout (in milliseconds) for the request
<code>type</code>	Specifies the type of request. (GET or POST)
<code>url</code>	Specifies the URL to send the request to. Default is the current page

.AJAX() – PRIMENA

```
<html>
<head>
  <meta charset="utf-8">
  <title>Index 1</title>
  <script src="jquery-3.4.1.js"></script>
</head>
<body>
  <h2>index 1</h2>
  <button id="dugme">Klikni me</button><hr>
  <div id="odgovor">Ovde ide odgovor</div>
</body>
</html>
<script>
$(document).ready(function() {
  $("#dugme").click(function() {
    $.ajax({
      url: "nepostoji.txt",
      success: function(response) {
        $("#odgovor").html(response);
      },
      error: function(xhr) {
        $("#odgovor").html("<div style='color:red'>Došlo je do greške<br>" + xhr.status + " " + xhr.statusText + "</div>");
      },
      beforeSend: function() {
        alert("Ovo se događa pre slanja samog zahteva");
      },
      complete: function() {
        alert("Asinhrona komunikacija završena");
      }
    });
  });
});
</script>
```



.AJAX() – PRIMENA SA POST METODOM

```
<!--index.html-->
<html>
  <head>
    <meta charset="utf-8">
    <title>index 2</title>
    <script src="jquery-3.4.1.js"></script>
    <script src="skript.js"></script>
  </head>
  <body>
    <h2>Index 2</h2>
    <input type="text" id="ime" name="ime" placeholder="Unesite ime"/><br><br>
    <input type="text" id="prezime" name="prezime" placeholder="Unesite prezime"/><br><br>
    <button id="dugme">Klikni me</button><hr>
    <div id="odgovor">Ovde ide odgovor</div>
  </body>
</html>
```

```
<?php
//obrada.php
$ime=$_POST['ime'];
$prezime=$_POST['prezime'];
echo "Vaši podaci su: $ime $prezime";
?>
```

```
//script.js
$(document).ready(function() {
  $("#dugme").click(function() {
    $.ajax({
      url: "obrada.php",
      type: "POST",
      data: {ime: $("#ime").val(), prezime: $("#prezime").val()},
      success: uspeh,
      error: greska,
      beforeSend: preSlanja
    });
  });
});

function uspeh(response) {
  $("#odgovor").html(response);
}

function greska(xhr) {
  $("#odgovor").html(xhr.status+" "+xhr.statusText);
}

function preSlanja() {
  if($("#ime").val()==" " || $("#prezime").val()==" ")
  {
    $("#odgovor").html("<span style='color:red'>Niste uneli sve podatke</span>");
    return false;
  }
}
```

```
<html>
  <head>
    <meta charset="utf-8">
    <title>index 3</title>
    <script src="jquery-3.4.1.js"></script>
    <script src="script.js"></script>
    <style>
      .spoljasnji{
        border: 1px solid red;
        margin: 2px;
        padding: 2px;
        display: inline-block;
        /*width: 200px;*/
        height: 200px;
        text-align:center;
      }
      .klasaBrisanje{
        width:100px;
        border: 1px solid black;
        cursor: pointer;
        margin:10px auto;
        text-align: center;
      }

      .klasaBrisanje:hover{
        background-color: lightgray;
      }

    </style>
  </head>
  <body>
    <h2>Index 3</h2>
    <form id="forma" name="forma" action="" method="POST" enctype="multipart/form-data" onsubmit="return false">
      <input type="file" id="fajl" name="fajl" /><br><br>
      <button id="dugme">Klikni me</button>
    </form>
    <br>
    <div id="status"></div>
    <hr>
    <div id="odgovor"></div>
  </body>
</html>
```

```
<?php
$time=$_FILES['fajl']['name'];
$temp=$_FILES['fajl']['tmp_name'];
if(@move_uploaded_file($temp, "upload/".$time))
{
    echo $time;
}
else
    echo "<span style='color:red'>NEUSPEŠAN upload datoteke</span>";
?>
```

```
$(document).ready(function(){
    $("#dugme").click(function(){
        //alert("Kliknut sam");
        $.ajax({
            url: "obrada3.php",
            type: "post",
            data: new FormData(document.getElementById("forma")),
            contentType: false,
            cache: false,
            processData: false,
            success: uspeh,
            error: greska,
            beforeSend: preSlanja
        });
    });
});
function uspeh(response)
{
    //$("#odgovor").html(response);
    var spoljasnji=$("#<div class='spoljasnji'></div>");
    var slika=$("#<img src='upload/'+response+' height='150px'>");
    var brisanje=$("#<div class='klasaBrisanje' onclick='obrisiSliku(this)' data-ime='"+ response +
">Obriši</div>");
    spoljasnji.append(slika);
    spoljasnji.append(brisanje);
    $("#odgovor").append(spoljasnji);
    $("#status").html("<span style='color:green'>Uspešan upload</span>");
}
function greska(xhr){
    $("#odgovor").html(xhr.status+" "+xhr.statusText);
}
function preSlanja(){
    if($("#fajl").val()=="")
    {
        $("#status").html("<span style='color:red'>Niste uneli sve podatke</span>");
        return false;
    }
}
function obrisiSliku(a)
{
    $.post("brisanje.php", {ime: a.getAttribute("data-ime")}, function(response){
        if(response=="1")
        {
            $("#status").html("<span style='color:green'>Uspešno brisanje slike</span>");
            a.parentNode.parentNode.removeChild(a.parentNode);
        }
        else
            $("#status").html("<span style='color:red'>Neuspešno brisanje slike</span>");
    });
}
```

.LOAD()

► Osnovna sintaksa

- `$("#div1").load("test.txt");`
- `$("#div1").load("test.html #p1");`

► Proširena sintaksa

- `$("#div1").load("test.html", function(responseTxt, statusTxt, xhr){ ..... })`

.LOAD() - PRIMER

test.txt

Ovo je tekst u tekstualnoj datoteci

index.html

```
<html><head>
<script src="jquery-3.4.1.js "></script>
<script>
$(document).ready(function(){
    $("button").click(function(){
        $("#div1").load("test.txt");
    });
});
</script>
</head>
<body>
<div id="div1"><h2>jQuery i AJAX ce promeniti ovaj tekst</h2></div>
<button>Procitaj sadrzaj spoljne datoteke</button>
</body></html>
```

.GET()

- ▶ GET metoda, pored URL-a dokumenta koji treba da se izvrši, šalje i GET parametre
- ▶ Parametre čita skript jezik na serveru (u našem slučaju PHP), obrađuje ih i vraća rezultat jQuery-ju
- ▶ jQuery smešta rezultat na HTML stranicu

.GET() - SINTAKSA

- ▶ **`$.get(URL,data,function(odgovor,status,xhr),dataType)`**
 - ▶ *URL : putanja do datoteke koju pozivamo (obavezan)*
 - ▶ *data : podaci koje šaljemo (opcion)*
 - ▶ *function(odgovor,status,xhr) : callback funkcija (opcion)*
 - ▶ *dataType : tip odgovora (opcion)*

.GET() - SINTAKSA

► Postoji nekoliko načina

- `$.get("test.php?ime=Bosko&grad=Beograd");`
- `$.get("test.php", { ime:"Bosko", grad:"Beograd" });`
- `$.get("test.php", function(odgovor){
 alert("PHP je vratio: " + odgovor);
});`

.GET() - SINTAKSA

ajax2.php

```
<?php echo "ovo je iz PHP-a"?>
```

index2.html

```
<html>
<head>
  <script src="jquery-3.4.1.js"></script>
  <script>
    $(document).ready(function() {
      $("button").click(function() {
        $.get("ajax2.php", function(data, status) {
          $("div").html("Data: <b>" + data + "</b><br>Status: " + status);
        });
      });
    });
  </script>
</head>
<body>
  <button>Slanje HTTP GET zahteva</button>
  <div></div>
</body>
</html>
```

.GET() - SINTAKSA

ajax3.php

```
<?php
    $ime=$_GET['ime'];
    $grad=$_GET['grad'];
    echo "Dobrodosao $ime, kako je u $grad?";
?>
```

index3.html

```
<html>
<head>
    <script src="jquery-3.4.1.js"></script>
    <script>
        $(document).ready(function() {
            $("button").click(function() {
                $.get("ajax3.php?ime=Bosko&grad=Beograd", function(data) {
                    $("div").html("<h2>" + data + "</h2><br>");
                });
            });
        });
    </script>
</head>
<body>
<button>Slanje HTTP GET zahteva sa parametrima</button>
<div></div>
</body>
</html>
```

.GET() - SINTAKSA

ajax3.php

```
<?php
    $ime=$_GET['ime'];
    $grad=$_GET['grad'];
    echo "Dobrodosao $ime, kako je u $grad?";
?>
```

index5.html

```
<html>
<head>
    <script src="jquery-3.4.1.js"></script>
    <script>
        $(document).ready(function() {
            $("button").click(function() {
                $.get("ajax3.php", {"ime":$("#tekst1").val(), "grad":$("#tekst2").val()}, function(data) {
                    $("div").html("<h2>" + data + "</h2><br>").fadeIn(2000);
                });
            });
        });
    </script>
</head>
<body>
    <input type="text" id="tekst1" placeholder="Unesite ime"/><br>
    <input type="text" id="tekst2" placeholder="Unesite grad"/><br>
    <button>Slanje HTTP GET zahteva sa parametrima iz text box-a</button>
    <div style="display:none"></div>
</body>
</html>
```



.POST()

- ▶ POST metoda, pored URL-a dokumenta koji treba da se izvrši, šalje i POST parametre
- ▶ Parametre čita skript jezik na serveru (u našem slučaju PHP), obrađuje ih i vraća rezultat jQuery-ju
- ▶ jQuery smešta rezultat na HTML stranicu

.POST() - SINTAKSA

▶ **`$.post(URL, data, function(odgovor, status, xhr));`**

- ▶ *URL : putanja do datoteke koju pozivamo (obavezan)*
- ▶ *data : podaci koje šaljemo (opcion)*
- ▶ *function(odgovor, status, xhr) : callback funkcija (opcion)*

.POST() - SINTAKSA

► Najčešći način

► **\$.post**("ajax6.php",{ "ime":"Bosko",
"grad":"Beograd"}, function(data, status){...}):

.POST() - SINTAKSA

ajax6.php

```
<?php
    $ime=$_POST['ime'];
    $grad=$_POST['grad'];
    echo "Dobrodosao $ime, kako je u $grad?";
?>
```

index6.html

```
<html>
<head>
    <script src="jquery-3.4.1.js"></script>
    <script>
        $(document).ready(function() {
            $("button").click(function() {
                $.post("ajax6.php", {"ime": "Bosko", "grad": "Beograd"}, function(data, status) {
                    $("div").html("<h2>" + data + "</h2>Status: " + status);
                });
            });
        });
    </script>
</head>
<body>
<button>Slanje HTTP POST zahteva sa parametrima</button>
<div></div>
</body>
</html>
```



.POST() - SINTAKSA

ajax6.php

```
<?php
    $ime=$_POST['ime'];
    $grad=$_POST['grad'];
    echo "Dobrodosao $ime, kako je u $grad?";
?>
```

index7.html

```
<html>
<head>
    <script src="jquery-3.4.1.js"></script>
    <script>
        $(document).ready(function() {
            $("#button").click(function() {
                $.post("ajax6.php", {"ime":$("#tekst1").val(), "grad":$("#tekst2").val()}, function(data) {
                    $("#div").html("<h2>" + data + "</h2><br>").fadeIn(2000);
                });
            });
        });
    </script>
</head>
<body>
    <input type="text" id="tekst1" placeholder="Unesite ime"/><br>
    <input type="text" id="tekst2" placeholder="Unesite grad"/><br>
    <button>Slanje HTTP POST zahteva sa parametrima iz text box-a</button>
    <div style="display:none"></div>
</body>
</html>
```