

Osnove izrade PHP aplikacija jQuery

Izradio: Marijan Šuflaj
Uredio: Davor Cihlar

Plan

- ▶ Što ćemo naučiti
 - ▶ Upravljanje DOM elementima
 - ▶ Hvatanje događaja
 - ▶ Animacije
 - ▶ AJAX zahtjeve

jQuery

- ▶ jQuery je biblioteka za JavaScript
- ▶ Vrlo jednostavna za korištenje
 - ▶ Ne ovisi o pregledniku
 - ▶ Ubrzava rad s JavaScriptom
 - ▶ Pristup DOM elementima
 - ▶ Stvaranje animacija
 - ▶ Upravljanje događajima
 - ▶ Izrada AJAX aplikacija
 - ▶ Sve se radi preko jQuery objekta koji reprezentira jedan ili više DOM elementa (engl. *wrapper*)
- ▶ Više na:
 - ▶ http://docs.jquery.com/Main_Page

Uključivanje jQueryja

- ▶ Potrebno uključiti JS datoteku
- ▶ Preporuka je datoteku uključiti s Googlea
 - ▶ U tom slučaju nije potrebna instalacija
 - ▶ `<script type="text/javascript" src="https://ajax.googleapis.com/ajax/libs/jquery/1.5.2/jquery.min.js"></script>`

Okruživanje DOM elemenata

- ▶ JavaScriptov DOM objekt (element) moguće je pretvoriti u jQuery objekt
 - ▶ Na taj način dobivamo jQueryjevu jednostavnost upravljanja tim DOM objektom
 - ▶ Dobivamo i neovisnost o pregledniku
- ▶ Primjer:
 - ▶

```
var jsDOME1, jqE1;  
jsDOME1 = document.getElementById('ajDi');  
jqE1 = $(jsDOME1);  
alert(jqE1.val());  
alert($(jsDOME1).val());
```
- ▶ Kraće:
 - ▶

```
alert($('#ajDi').val()); // pomocu selectora
```

Selektori

- ▶ Omogućuju odabir DOM elementa (ili više njih)
 - ▶ Pomoću funkcija `jQuery(selector)` i `$(selector)`
 - ▶ Funkcije vraćaju `jQuery` objekt koji reprezentira jedan ili više DOM element
- ▶ Tri bitna tipa selektora:
 - ▶ `#naziv` - element s identifikatorom `id=naziv`
 - ▶ `.naziv` - elementi s `class=naziv`
 - ▶ `naziv` - elementi tog naziva (`a`, `div`, `table`, ...)
- ▶ Selektore je moguće kombinirati zajedno
 - ▶ Npr. `nazivTaga.nazivClassa` prezentira sve tagove (DOM elemente) s nazivom `nazivTaga` i atributom `class=nazivClassa`

Selektori – primjeri

- ▶ `$("#naziv")`
 - ▶ Element s id=naziv
- ▶ `$(".test")`
 - ▶ Svi elementi s class=test
- ▶ `$("a")`
 - ▶ Svi linkovi
- ▶ `$("a.test")`
 - ▶ Svi linkovi s class=test
- ▶ `$("#naziv a.test")`
 - ▶ Svi linkovi s class=test koji su unutar elementa s id=naziv
- ▶ `$(".test, a.klasa")`
 - ▶ Svi linkovi s class=test **ili** class=klasa

jQuery objekt

- ▶ Funkcije `$` i `jquery` vraćaju `jquery` objekt
- ▶ `jquery` objekt može reprezentirati jedan ili više DOM elementa
- ▶ Postoji velik broj metoda
 - ▶ Proširenjem (engl. *plugin*) je moguće dodati vlastite metode

Važne metode jQuery objekta

- ▶ `.text()` - vraća tekst unutar elementa
- ▶ `.text(vrijednost)` - postavlja tekst
- ▶ `.val()` - vraća vrijednost atributa `value`
- ▶ `.val(vrijednost)` - postavlja vrijednost
- ▶ `.hide(vrijemeAnimacije)` - skriva element
 - ▶ `vrijemeAnimacije` je neobavezan parametar u [ms]
- ▶ `.show(vrijemeAnimacije)` - prikazuje element
 - ▶ `vrijemeAnimacije` je neobavezan parametar u [ms]

Metoda text

- ▶ **Pažnja:** metode `text()` i `text(vrijednost)` rade nad dječom odabranog elementa
 - ▶ Poput `innerHTML`
 - ▶ `text()` vraća tekst unutar elementa, ali zajedno sa sadržajem djece elementa
- ▶ Primjer:
 - ▶ `<p id="tekst">Lijep dan</p>`
 - ▶ `alert($("#tekst").text());`
 - ▶ Ispisuje "Lijep dan"
 - ▶ `$("#tekst").text("Lijep dan");`
 - ▶ Rezultira bez B tagova:
 - `<p id="tekst">Lijep dan</p>`

Primjer (1)



```
▶ <input type="text" id="tekst"  
      value="Zdravo" />
```

```
<script type="text/javascript">
```

```
var elTxt = $("#tekst");
```

```
elTxt.val(elTxt.val() + " svijete!");
```

```
</script>
```

Događaji

- ▶ Pomoću sljedećih metoda moguće je definirati obradu događaja:
 - ▶ `.click(fn)` – funkcija `fn` poziva se zbog pritiska mišem na element
 - ▶ `.change(fn)` – funkcija `fn` poziva se zbog promjene
 - ▶ `.hover(fn)` – funkcija `fn` poziva se zbog prelaska mišem preko elementa
 - ▶ `.submit(fn)` – funkcija `fn` poziva se prije objave obrasca
- ▶ Ako funkcija za obradu događaja vrati `false`, zaustavlja se daljnja obrada događaja
- ▶ U funkciji događaja `this` je DOM objekt
 - ▶ Kako bi dobili pristup metodama `jQuery` objekta potrebno je koristiti `$(this)` ili `jQuery(this)`

Čekanje na spremnost stranice

- ▶ Kao i prije, potrebno je paziti da su svi DOM elementi definirani prije skripte koja se odmah pokreće
 - ▶ Stranica se učitava slijedno i skripte se pokreću nakon što se učitaju, ali vrlo vjerojatno ono iza skripte još nije učitano
- ▶ Događajem `ready(fn)` DOM elementa `document` možemo pričekati kraj učitavanja stranice
 - ▶ Funkcija `fn` poziva se nakon što je stranica učitana

Primjer (2)



- ▶ **function** stranicaUcitana() {
 alert("Stranica je ucitana");
}
\$(document).ready(stranicaUcitana);

- ▶ Ili kraće pomoću anonimne funkcije:
 - ▶ \$(document).ready(**function**() {
 alert("Stranica je ucitana");
});

Primjer (3)



- ▶

```
$(document).ready(  
    function() {  
        $("a").click(  
            function(e) {  
                var obj = $(this);  
                alert(obj.text());  
                e.preventDefault();  
            }  
        );  
    }  
);
```
- ▶ Na svaki link na koji se pritisne ispisuje se njegov tekst i zabranjuje napuštanje stranice
- ▶ U HTML postavite par običnih linkova

Zadatak (1)



- ▶ Unutar span elementa s identifikatorom `id=brojac` ispisati koliko puta smo pritisnuli na neki link
 - ▶ Metoda `text()`
- ▶ Zabraniti daljnju obradu događaja
- ▶ Kada brojač pređe 10, sakriti sve linkove
 - ▶ Animacija skrivanja neka traje 1500ms

AJAX

- ▶ AJAX omogućuje dohvaćanje podataka s poslužitelja bez osvježavanja stranice
 - ▶ Podatke je moguće dohvaćati samo s poslužitelja s kojeg je dohvaćena stranica!
 - ▶ Npr. s php.fer.hr ne možemo dohvaćati podatke s www.fer.hr ili www.google.hr
- ▶ Vrlo složeno bez jQueryja i ovisi o pregledniku
- ▶ jQuery daje tri načina dohvaćanja podataka:
 - ▶ `$.get()` – metoda koja dohvaća zahtjevom GET
 - ▶ `$.post()` – dohvaća podatke zahtjevom POST
 - ▶ `$.ajax()` – moćnija inačica get/post metode

`$.get()`

- ▶ `$.get(url, data, callback)`
- ▶ Prima tri parametra:
 - ▶ `url` – putanja do lokacije s koje se dohvaća sadržaj
 - ▶ Najčešće se radi o PHP skripti
 - ▶ `data` – asocijativno polje (objekt) s parametrima
 - ▶ `{a:1, b:"sto"}`
 - ▶ `callback` – funkcija koja će biti pozvana nakon što se cijeli podatak dohvati
 - ▶ Parametar toj funkciji je dekodirani podatak (`string`, `number`, `array`, `object` – ovisno o dohvaćenom podatku)
- ▶ Tip podatka kojeg dohvaća određen je u zaglavlju podatka (PHP-ova funkcija `header`)
 - ▶ Koristit ćemo samo JSON (`application/json`)

Primjer (4)



- ▶ PHP skripta koja vraća trenutno vrijeme:

- ▶ `header('Content-type: application/json');`

```
$vrijeme = "Vrijeme je:\n" . date('r');  
$json = json_encode($vrijeme);  
echo $json;
```

- ▶ JavaScript koji dohvaća rezultat gornje skripte:

- ▶ `function prikaziVrijeme(vrijeme) {
 alert(vrijeme);
}`

```
$(document).ready(function() {  
    $('#tipka').click(function(e) {  
        $.get('vrijeme.php', {}, prikaziVrijeme);  
        // ne želimo poslat obrazac:  
        e.preventDefault();  
    });  
});
```

Primjer (5.1)



▶ Internetska stranica koja zbraja na poslužitelju

▶ HTML:

▶ **<body>**

```
a=<input type="text" id="a" /> ,
```

```
b=<input type="text" id="b" /><br />
```

```
<input type="submit" value="a+b="  
id="tipka" />
```

```
<input type="text" id="rezultat" />
```

```
</body>
```



Primjer (5.2)



▶ PHP skripta:

```
▶ header('Content-type: application/json');
```

```
$zbroj = $_GET['a'] + $_GET['b'];  
$json = json_encode($zbroj);  
echo $json;
```

Primjer (5.3)



▶ JavaScript:

```
▶ function prikaziRezultat(rez) {  
    $('#rezultat').val(rez);  
}  
function izracunaj(e) {  
    var varA = $('#a').val();  
    var varB = $('#b').val();  
    var opcije = {a: varA, b: varB};  
    $.get('xzbrajalo.php', opcije,  
        prikaziRezultat);  
    e.preventDefault();  
}  
$(document).ready(function() {  
    $('#tipka').click(izracunaj);  
});
```

Zadatak (2)



- ▶ Izraditi dinamičko biranje grada
 - ▶ Korisnik bira županiju
 - ▶ Odabirom županije ponuđuju se gradovi iz odabrane županije
 - ▶ Koristiti AJAX za dohvaćanje gradova iz županije
 - ▶ Koristiti naše jQuery proširenje za rad sa `select`
 - ▶ `.addOption(vrijednost, tekst)` – dodaje opciju u `select`
 - ▶ `.removeOptions()` – briše sve opcije iz `select`
 - ▶ `.getSelectedValue()` – vraća vrijednost odabrane opcije
 - ▶ Iskoristite i kostur zadatka
 - ▶ U kosturu je implementirana funkcionalnost za koju se pretpostavlja da ju znate, a nepotrebno bi Vam potrošilo vrijeme prilikom rješavanja ovog zadatka i skrenulo pažnju s poante zadatka
 - ▶ **Pažnja!** JavaScript datoteke moraju imati ovlasti 644

Zadatak (3.1)



- ▶ Omogućiti korisniku prikaz svih artikala
- ▶ Listu dohvaćati isključivo preko AJAX-a
 - ▶ Stranicu prikazati praznu s animiranom sličicom koja označava da se tablica učitava
 - ▶ Potražite "spinner filetype:gif" na <http://images.google.com>
 - ▶ Kad se stranica učita (dogadaj `ready`) dohvatiti (pomoću `$.get`) i prikazati (metodom `html` umjesto `val`) listu artikala iz skripte "ajaxArtikli.php"
 - ▶ Skripta artikle vraća kao polje, ali potrebno ih je prikazati po redovima
 - U JavaScriptu generirati HTML s `
` ili s tablicom
 - ▶ Koristite `div` kao korijenski element kojem će se mijenjati unutarnji HTML
- ▶ Korisniku omogućiti osvježavanje tablice
 - ▶ Pritiskom na tipku "Osvježi" ponovno se dohvaća

Zadatak (3.2)



- ▶ Dodajte podršku za dodavanje artikala
 - ▶ Isto koristeći AJAX! Stranica se ne smije ponovno učitavati.
- ▶ Prikazujte sve podatke o artiklu (red. br., naziv i cijenu)
- ▶ Omogućite brisanje artikla koristeći AJAX