

6. pripremno predavanje

- JavaScript
- XMLHttpRequest

6. laboratorijska vježba

- Zadnja vježba 😊
- Završno "dotjerivanje" Web aplikacije
- Naglasak na:
 - klijentskim tehnologijama (**JavaScript**)
 - dohvaćanju informacija s poslužitelja bez osvježavanja stranice (**XMLHttpRequest**)
- Potrebne (manje) promjene ostalih datoteka

Cilj vježbe



- Dopuniti Web aplikaciju promjenjivim elementima na strani klijenta (*DHTML, JavaScript*)
 - iskočni okviri s dodatnim informacijama (*tooltips*)
- Omogućiti dohvat dodatnih podataka komunikacijom klijent-poslužitelj bez osvježavanja stranice (*XMLHttpRequest*)
- U Web aplikaciju uključiti vanjske, javno dostupne biblioteke određene namjene (*JavaScript*)

Proučiti:

- **Osnove osnova jezika JavaScript**
 - uključivanje JavaScripta u HTML
 - rad s varijablama
 - pozivi funkcija
- **Koncept DHTML-a**
 - model DOM
 - rukovanje događajima (*event handling*)
 - promjena stilova
- **Objekt XMLHttpRequest**
 - slanje zahtjeva s parametrima
 - obrada - proslijeđivanje odgovora

Zadatak (1. dio) - *tooltip*

- Prelaskom strelice miša preko pojedinog retka prikazati ***tooltip*** s dodatnim informacijama
 - promjena HTML/JavaScript kôda kojeg generira *pretraga.php*
 - informacije se nalaze u JavaScript kôdu, **nema ponovne komunikacije** s poslužiteljem!
 - registracija *event handlera* za događaj *onmouseover*

```
<tr onmouseover="tooltip( 'osobaA@fer.hr' ) ">
    <td>
        Osoba A
    </td>
</tr>
```

Zadatak (1. dio) - vanjske biblioteke

- Tri predložene biblioteke *tooltipa*
 - možete koristiti i druge (ili ih ostvariti sami 😊)
- Uvijek postoji kratki uvod u korištenje biblioteke
- Osnovna ideja:
 - preuzeti .js datoteke
 - uključiti .js datoteke u svoj kôd
 - **pri aktivaciji događaja** pozvati funkciju za prikaz
 - funkciji predati **tekst** i **argumente prilagodbe**
- tekst se obično predaje kao HTML (dozvoljena sva formatiranja, definiranje klasa za CSS...)

Zadatak (2. dio) - detaljne informacije

- Tablica rezultata pretraživanja ima novi stupac "Akcija", s tekstom ("Više"...), slikom ili gumbom u svakom retku
 - klikom na **pojedini aktivni element** (događaj *onclick*) priprema se **komunikacija** s poslužiteljem
- Objekt **XMLHttpRequest**
 - nema osvježavanja stranice
 - **novoj** skripti na poslužitelju (***detalji.php***) treba poslati **parametar (id)** pojedinog osnovnog elementa
 - skripta *detalji.php* **vrlo slična** skripti *pretraga.php*
 - HTTP zahtjev, metoda GET

Zadatak (2. dio) - obrada odgovora

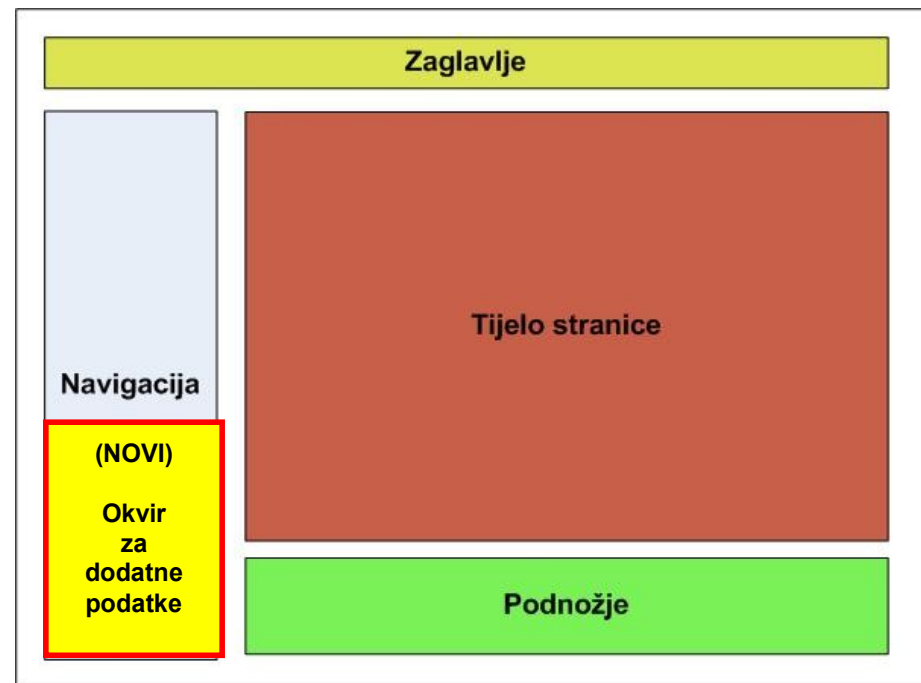


- Skripta *detalji.php* pomoću XPatha ili DOM-a pretražuje XML **po id-u**, te vraća dodatne informacije o osnovnom elementu određenog id-a
- Izlaz skripte je "**mali**" **HTML kôd**
 - spreman za "ubacivanje" na stranicu
 - sadrži sve potrebne oznake, poput <table>, <div>, definicije klasa za CSS i slično
- Na strani klijenta, po primitku odgovora, JavaScript prenosi dobiveni kôd u okvir za dodatne informacije

Promjene u ostalim datotekama



- pretraga.php
 - smanjiti tablicu (treba imati 3 - 5 stupaca)
 - dodati stupac Akcija
 - dodati JavaScript kôd za prikaz *tooltipa* za **svaki redak**
 - **ispod navigacije** dodati **okvir za dodatne podatke** (`<div>`)
- CSS stilovi
 - prilagoditi izgled i strukturu



Promjene u ostalim datotekama

- Za dohvaćanje pojedinog osnovnog elementa potreban je jedinstveni identifikator (naziva "id")
 - npr. `<osoba id="5">`
- podaci.dtd
 - dodati atribut "id" osnovnom elementu (**tip atributa: ID**)
- Java servlet
 - pri generiranju XML-a dodati id
 - preporuka: **sljedni broj**
 - jedan od načina: konstruktor osnovnog elementa **proširiti identifikatorom**, pa ga preuzeti u metodi *ToXML*

Brzi uvod u:

- JavaScript
- XMLHttpRequest

Uključivanje JavaScripta

- U <head> ili <body> dijelu
 - razlike u trenutku učitavanja i "garanciji" učitavanja prije nekog događaja
- *Inline*
- **Vanjska JavaScript datoteka**

```
<head>
  <script type="text/javascript" src="mojJS.js">
  </script>
</head>
```

Sintaksa JavaScripta

- Slična C-u/Javi/PHP-u...
 - if, for, pridruživanja...
 - **new** (instanciranje objekata)
- Spajanje stringova (+)


```
datum = godina + '-' + mjesec + '-' + dan;
```
- Varijable
 - globalne i lokalne
 - tip nije određen pri deklaraciji, ovisan o vrijednosti varijable, promjenjiv

```
var id = 5;
```

- Funkcije

```
function ispisi (mojTekst) {
    alert(mojTekst);
    return true;
}
```

Događaji (*events*)

- Događaji se odnose na elemente HTML-a (odlomke teksta, linkove, gumbe, obrasce...)
- Događaje okidaju:
 - miš (**onclick**, **onmouseover**, **ondblclick**...)
 - tipkovnica (**onkeypress**, **onkeyup**...)
 - HTML objekti i obrasci (**onload**, **onresize**, **onsubmit**...)
- Povezivanje događaja na nekom elementu s naredbama/funkcijama koje treba izvršiti

```
<a href="link.html" onclick="promijeni()">
```

DOM: važna svojstva i metode

- Pristupanje elementima (W3C)
 - Jednom elementu s definiranim atributom *id*
`document.getElementById()`
 - Svim elementima s definiranim atributom *name*
`document.getElementsByName()`
 - Svim elementima iste oznake
`document.getElementsByTagName()`
- `nav = document.getElementById("navigacija");`
- Dobro je provjeriti je li element pronađen
 - inače često radimo s *null* vrijednostima i dobivamo JS pogreške ☹️

DHTML: važna svojstva i metode



- Dinamička promjena svojstva elemenata

- promjena sadržaja elementa

```
document.getElementById('naslov').innerHTML = 'Novi  
naslov';
```

- promjena izgleda

```
document.getElementById('naslov').style.color = 'red';
```

- promjena vidljivosti

```
document.getElementById('naslov').style.visibility =  
'hidden';
```

```
document.getElementById('naslov').style.display =  
'none';
```

- Promjena naziva gumba

```
document.getElementById('gumb').value = 'Novi';
```


XMLHttpRequest (AJAX)

1. Instanciranje objekta
2. Povezivanje s URL-om
 - odabir metode HTTP zahtjeva
3. Slanje zahtjeva
4. Čekanje odgovora poslužitelja (asinkrono!)
 - "život" stranice za vrijeme čekanja normalno nastavljen
5. Ako je odgovor OK, poziv funkcije za parsiranje odgovora
6. (najčešće) Prikaz rezultata unutar Web stranice

Demistificirani XMLHttpRequest



```
if (window.XMLHttpRequest) {  
    var xmlHttp = new XMLHttpRequest();  
}  
else  
{  
    var xmlHttp = new ActiveXObject("Microsoft.XMLHTTP");  
}  
xmlHttp.open("GET", "http://www.primjer.hr");  
xmlHttp.onreadystatechange = promijeniStranicu;  
xmlHttp.send(null);
```

provjeri tip
Web
preglednika,
instanciraj objekt

poveži objekt s
URL-om i
odaberi metodu
zahtjeva

pošalji zahtjev
(u metodi POST,
ovdje se dodaju
parametri)

imenuj funkciju
koja će se izvršiti
po promjeni stanja
zahtjeva

Demistificirani XMLHttpRequest



```
function promijeniStranicu() {  
  if (xmlHttp.readyState == 4) {  
    if (xmlHttp.status == 200) {  
      var odgovor = xmlHttp.responseText;  
      alert ("Dobio sam " + odgovor);  
    }  
    else {  
      alert ("Nešto loše se dogodilo");  
    }  
  }  
}
```

XML zbog DOM
parsiranja treba primiti
s .responseXML

funkcija koja
se izvršava pri
promjeni
stanja zahtjeva

provjeri je li
dovršeno
primanje
odgovora

provjeri je li
odgovor OK

Slanje parametara, primanje podataka

- Za metodu **GET**, parametri se šalju **unutar URL-a**
 - npr. `http://www.fer.hr/detalji.php?id=523&show=simple`)
- Za metodu **POST**, parametri se pišu jednako kao i za GET, no šalju se prilikom poziva **funkcije *send***
 - npr. `req.send("id=523&show=simple") ;`
- Primanje podataka
 - `req.responseText` -> obično ga se "preseli" u neki `<div>`
 - `req.responseXML` -> obično ga se transformira DOM-om

Upucavanje nevolja (*troubleshooting*)

- Čudne sintaksne pogreške vezane za *stringove*
 - pomiješali ste jednostruke (') i dvostruke navodnike (")
 - pazite na navodnike unutar navodnika
 - `<tr onmouseover="alert('Čiča "Gliša"');">`
 - escape znakovi (\", \')
 - stavljanje teksta u varijablu (jedna razina navodnika manje)
 - promjena HTML navodnika u jednostruke
- Prelazak u "novi redak" ne radi. Čak ni \n ne radi.
 - HTML kôd -> `
` ili `</div>`
- XMLHttpRequest mi ne vraća ništa ili nešto krivo
 - probajte u preglednik upisati izravni URL
(.....detalji.php?id=24), zbog lakšeg *debugiranja*

Pomoćni alati

- Vrlo, vrlo korisni
- JavaScript *debuggeri*
- Programi za praćenje prometa protokola HTTP
- Firefox:
 - DOM Inspector
 - Error Console
 - **Firebug**



Pitanja?

- Probleme i pitanja možete i naknadno postaviti u forumu na stranicama predmeta