

Web dizajn i programiranje

Prof. dr.sc. Dragutin Kermek
Sveučilište u Zagrebu
Fakultet organizacije i informatike
Pavlinska 2, Varaždin 42000
dkermek@foi.hr

12. dio

Programiranje na strani korisnika

Uvod u Javascript jezik i koncept programiranja.

Programiranje događaja.

Razvoj dinamičkih osobina dokumenata.

Objektni model dokumenta (Document Object Model - DOM).

Dozvoljeni izrazi (Regular Expressions - RegExp).

Web 2.0., Ajax

jQuery

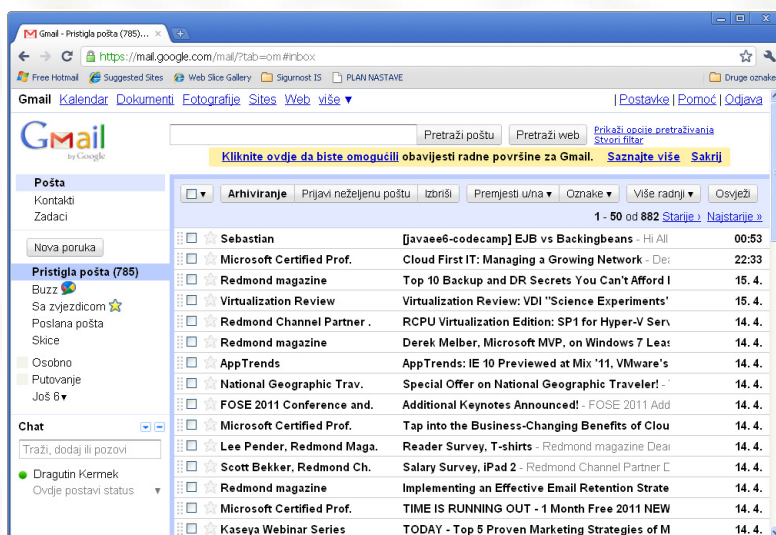


Web 2.0

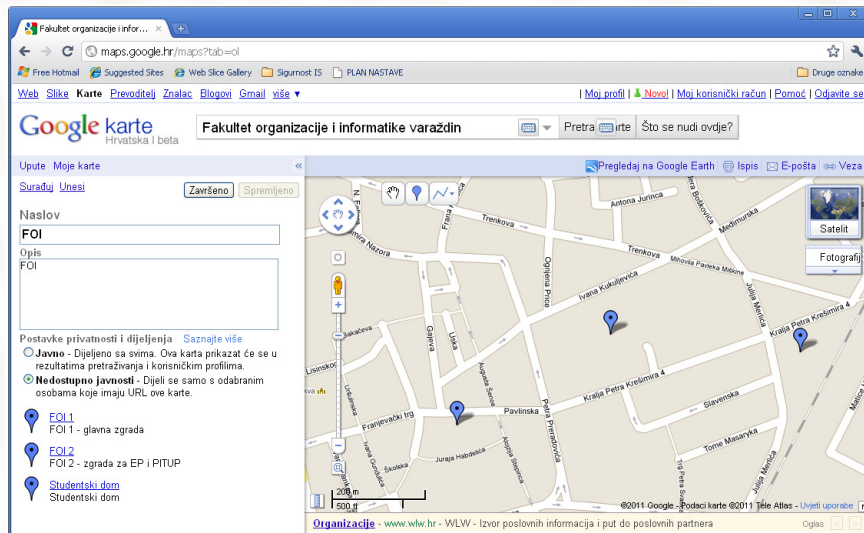
- *Web 2.0 je skup ekonomskih, socijalnih i tehnoloških trendova koji zajednički daju bazu za novu generaciju Interneta – zreliji, prepoznatljiviji medij koji je karakteriziran sudjelovanjem korisnika, otvaranjem i mrežnim efektima.*
- *Najbolja praksa:*
 - ⇒ Što prije i što češće objavi
 - ⇒ Uključi korisnike kao suradnike u razvoju i testere u stvarnom vremenu
 - ⇒ Opremi svoj proizvod
 - ⇒ Ubrzano kreiraj nove proizvode
 - ⇒ Postavi djelovanje kao jezgrenu kompetenciju
 - ⇒ Koristi dinamičke alate i jezike
- **O'REILLY RADAR REPORT Web 2.0 Principles and Best Practices**
http://www.oreilly.com/catalog/web2report/chapter/web20_report_excerpt.pdf
- **Primjeri Web 2.0**
 - ⇒ <http://www.google.com/maphp?hl=en&tab=wl&q>
 - ⇒ <http://www.flickr.com/>
 - ⇒ <http://www.facebook.com/>
 - ⇒ <http://www.linkedin.com/>



GMail – Google Mail



GoogleMaps

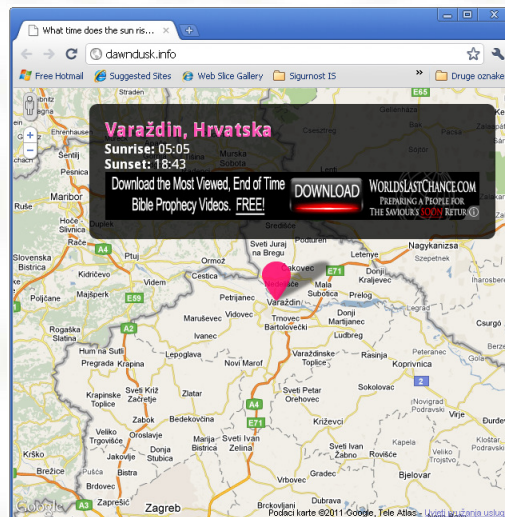


5

Web dizajn i programiranje - FOI
Prof.dr.sc. D.Kermek, Fakultet organizacije i informatike, Varaždin



Korištenje GoogleMaps

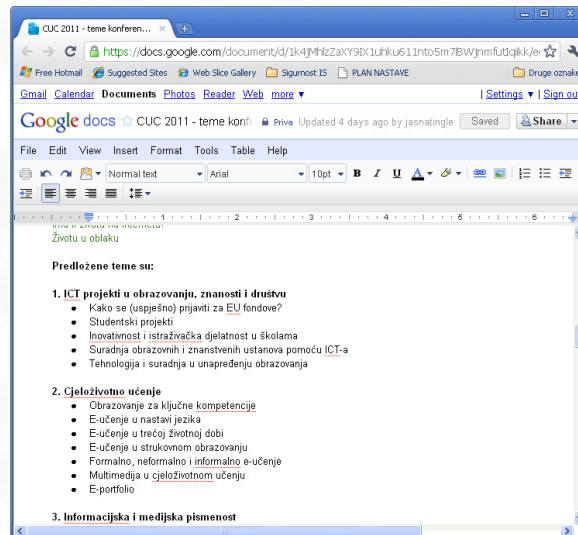


6

Web dizajn i programiranje - FOI
Prof.dr.sc. D.Kermek, Fakultet organizacije i informatike, Varaždin



Google docs



7



Google proizvodi



8

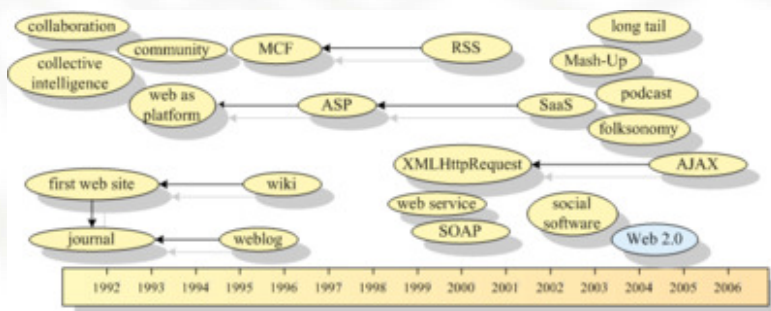


Razine Web 2.0

- Razina 3: Aplikacija može samo postojati na mreži, njena snaga proizlazi iz mreže i veza koje se uspostavljaju između ljudi ili aplikacija (eBay, craigslist, Wikipedia, del.icio.us, Skype)
- Razina 2: Aplikacija može postojati offline no svoju svoju jedinstvenu prednost postiže online (Flickr).
- Razina 1: Aplikacija može uspješno postojati offline no postiže dodatne osobine kada je online (Writely, Google Docs & Spreadsheets)
- Razina 0: Aplikacija se primarno koristi online no ona će raditi jednako i offline ukoliko su podaci u lokalnom cacheu (MapQuest, Yahoo! Local, and Google Maps).



Vremenska crta Web 2.0 termina



http://en.wikipedia.org/wiki/Web_2.0



Web 2.0 termini

- **Mashup** (web application hybrid) – webmjesto ili web aplikacija koja kombinira sadržaj iz više od jednog izvora u integrirano iskustvo
http://en.wikipedia.org/wiki/Mashup_%28web_application_hybrid%29
- **Podcast** – je multimedijaska datoteka koja se distribuira (uz plaćanje ili besplatno) preko Interneta korištenjem mreže distributera kako bi se prikazala na mobilnim uređajima ili PC-u <http://en.wikipedia.org/wiki/Podcast>

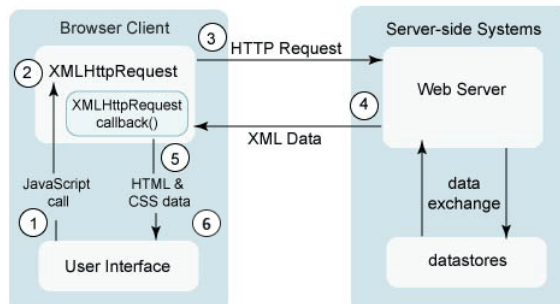


AJAX

- AJAX – Asynchronous JavaScript + XML
- Ponovno definiranje uloge Web aplikacija
- Klijent postaje jak
- 4 principa AJAX-a (D.Crane, E.Pascarelo:Ajax in Action)
 - ⇒ preglednik sadrži aplikaciju, ne sadržaj
 - ⇒ poslužitelj isporučuje podatke (XML, **JSON**), ne sadržaj (HTML)
 - ⇒ korisnička interakcija s aplikacijom može biti tekuća i kontinuirana
 - ⇒ to je stvarno kodiranje i zahtjeva disciplinu (?)



Obrada korisničke akcije



1. Korisnik generira događaj na strani klijenta. To uzrokuje poziv Javascript funkcije
2. Javascript funkcija kreira i konfigurira XMLHttpRequest objekt na klijentskoj strani i definira funkciju povratnog poziva (callback)
3. XMLHttpRequest objekt stvara poziv – asinkroni HTTP zahtjev – za web poslužitelja
4. Web poslužitelj obrađuje zahtjev i vraća XML dokument koji sadrži rezultat
5. XMLHttpRequest objekt poziva funkciju povratnog poziva i vraća odgovor od web poslužitelja koji sadrži rezultat
6. Klijent ažurira HTML DOM predstavljajući stranicu s novim podacima



XMLHttpRequest

```

interface XMLHttpRequest {
    attribute EventListener onreadystatechange;
    readonly attribute unsigned short readyState;
    void open(in DOMString method, in DOMString url);
    void open(in DOMString method, in DOMString url, in boolean async);
    void open(in DOMString method, in DOMString url, in boolean async, in DOMString user);
    void open(in DOMString method, in DOMString url, in boolean async, in DOMString user, in DOMString password);
    void setRequestHeader(in DOMString header, in DOMString value);
    void send();
    void send(in DOMString data);
    void send(in Document data);
    void abort();
    DOMString getAllResponseHeaders();
    DOMString getResponseHeader(in DOMString header);
    readonly attribute DOMString responseText;
    readonly attribute Document responseXML;
    readonly attribute unsigned short status;
    readonly attribute DOMString statusText;
};
  
```



XMLHttpRequest atributi

- **onreadystatechange** – rukovatelj događaja koji se poziva kod svake promjene stanja
- **readyState** – stanje objekta
 - 0 = uninitialized
 - 1 = loading
 - 2 = loaded
 - 3 = interactive
 - 4 = complete
- **responseText** – odgovor kao string
- **responseXML** – odgovor kao XML
- **status** – bročani status (npr. 404 za "Not Found" ili 200 for "OK")
- **statusText** - status kao string (npr. "Not Found" ili "OK")



AJAX razvoj

- Zbog pozadinske komunikacije s poslužiteljem i obrade primljenih podataka korisno je imati pregled nad poslanim zahtjevom i vraćenim podacima odgovora
- Za Firefox preglednika postoji vrlo koristan dodatak Firebug
- Za Chrome preglednika također postoji vrlo koristan dodatak Razvojni programer
- Za Internet Explorer preglednika također postoji vrlo koristan dodatak Developer Tools
- Vlastito iskustvo pokazalo da su gotovo nezamjenjivo u raznim fazama rada s AJAX-om



Ajax primjer 1 – podaci: ajax_1.xml

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<kolegiji>
  <name studij="IS" sifra="01">ALGORITMI</name>
  <name studij="IS" sifra="02">ARHITEKTURA RACUNALNIH SUSTAVA</name>
  <name studij="IS" sifra="03">AUTOMATIZACIJA UREDSKOG POSLOVANJA</name>
  <name studij="IS" sifra="04">BAZE PODATAKA</name>
  <name studij="IS" sifra="71">DIREKTNI MARKETING</name>
  ...
</kolegiji>
```



Ajax primjer 1. - slanje podataka

```
<body>
  <form name="forma">
    <a href="javascript:dajPodatke()">Prikaži</a>
  </form>
  <div id="cilj"></div>
</body>
```

Pokreće postupak slanja zahtjeva i
preuzimanja odgovora

Obično se praznom DIV elementu
dodijeli id koji se kasnije koristi
kod punjenja sadržaja. Može se
CSS-om odrediti način prikaza
(tip, sakrivanje i sl.)



Ajax primjer 1. / 1

```
<script language="JavaScript" type="text/javascript">
<!--
var xmlHttp = createXmlHttpRequestObject();

function createXmlHttpRequestObject()
{
    var xmlHttp;
    try {
        xmlHttp = new XMLHttpRequest();
    } catch(e) {
        var XmlHttpVersions = new Array('MSXML2.XmlHttp', 'Microsoft.XmlHttp');
        for (var i=0; i<XmlHttpVersions.length && !xmlHttp; i++) {
            try {
                xmlHttp = new ActiveXObject(XmlHttpVersions[i]);
            }
            catch (e) {}
        }
    }
    if (!xmlHttp)
        alert("Problem kod kreiranja XMLHttpRequest objekta.");
    else
        return xmlHttp;
}
```

Funkcija za kreiranje
XmlHttpRequest
objekta

Kreiranje
XmlHttpRequest
objekta

Kreiranje
XmlHttpRequest
objekta za IE



Ajax primjer 1. / 2

```
function dajPodatke()
{
    if(xmlHttp) {
        try {
            if (xmlHttp.readyState == 4 || xmlHttp.readyState == 0) {
                xmlHttp.open("GET", "ajax_1.xml", true);
                xmlHttp.onreadystatechange = preuzmiPodatke;
                xmlHttp.send(null);
            } else { // ako je XMLHttpRequest objekt zauzet...
            }
        } catch(e) {
            alert("Problem kod povezivanja na server:\n" + e.toString());
        }
    }
}
```

Definiranje metode i
dokumenta za poziv

Funkcija koje se periodičko poziva
tijekom izvođenja zahtjeva



Ajax primjer 1. / 3

Završen odgovor

```
function preuzmiPodatke()
{
    if (xmlHttp.readyState === 4) {
        if (xmlHttp.status === 200) { // samo ako je HTTP status "OK"
            try {
                prikaziPodatke();
            } catch (e) {
                alert(e.toString());
            }
        } else {
            alert("Problem kod preuzimanja podataka:\n" + xmlHttp.statusText);
        }
    }
}
```



Ajax primjer 1. / 4

Desila se pogreška?

```
function prikaziPodatke()
{
    var response = xmlHttp.responseText;
    if (response.indexOf("ERRNO") >= 0 || response.indexOf("error:") >= 0 || response.length === 0)
        throw(response.length === 0 ? "Prazan odgovor servera." : response);
    response = xmlHttp.responseXML.documentElement;
    var nameArray = new Array();
    nameArray= pripremiPodatke(response.getElementsByTagName("name"));
```

DOM funkcija za preuzimanje elemenata s nazivom "name"

Preuzimanje DOM dokumenta



Ajax primjer 1. / 4

```
var inHTML = '<select id="kolegij" name="kolegij" size="20">';  
  
for(i=0; i<nameArray.length; i++) {  
    inHTML += '<option value="' + nameArray[i][0] + "#" + nameArray[i][1] +  
    ">" + nameArray[i][0] + " / " + nameArray[i][2] + '</option>';  
}  
inHTML += '</select>';  
var oSuggest = document.getElementById("cilj");  
oSuggest.innerHTML = inHTML;  
}
```

Kreiranje padajućeg
izbornika

Kreiranje stavki
padajućeg izbornika

Traženje elementa s id "cilj" i
pridruživanje sadržaja za njega



Ajax primjer 1. / 5

```
function pripremiPodatke(resultsXml)  
{  
    var resultsArray= new Array();  
    for(i=0;i<resultsXml.length;i++) {  
        var $studij = resultsXml.item(i).getAttribute("studij");  
        var $sifra = resultsXml.item(i).getAttribute("sifra");  
        var $naziv = resultsXml.item(i).firstChild.data;  
        var $podaci = new Array($sifra, $naziv);  
        resultsArray[i] = $podaci;  
    }  
    return resultsArray;  
}
```

DOM funkcije za preuzimanje
atributa i prvog podčvora



Ajax primjer 1.- pogreska

```
function dajPodatke()
{
    if(xmlHttp) {
        try {
            if (xmlHttp.readyState === 4 || xmlHttp.readyState === 0) {
                xmlHttp.open("GET", "ajax_1_pogreska.xml", true);
                xmlHttp.onreadystatechange = preuzmiPodatke;
                xmlHttp.send(null);
            } else { // ako je XMLHttpRequest objekt zauzet...
            }
        } catch(e) {
            alert("Problem kod povezivanja na server:\n" + e.toString());
        }
    }
}
```



Ajax primjer 1. - pogreska – podaci: ajax_1_pogreska.xml

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<kolegijii>
  <name studij="IS" sifra="01">ALGORITMI</name>
  <name studij="IS" sifra="02">ARHITEKTURA RACUNALNIH SUSTAVA</name>
  <name studij="IS" sifra="03">AUTOMATIZACIJA UREDSKOG POSLOVANJA</name>
  <name studij="IS" sifra="04">BAZE PODATAKA</name>
  <name studij="IS" sifra="71">DIREKTNI MARKETING</name>
  ...
</kolegijii>
```

Neispravan XML
dokument



Ajax primjer 2. - slanje XML-a

```
function dajPodatke()
{
    if(xmlHttp) {
        try {
            if (xmlHttp.readyState == 4 || xmlHttp.readyState == 0) {
                xmlHttp.open("GET", "ajax_2.php", true);
                xmlHttp.onreadystatechange = preuzmiPodatke;
                xmlHttp.send(null);
            } else { // ako je XMLHttpRequest objekt zauzet...
            }
        } catch(e) {
            alert("Problem kod povezivanja na server:\n" + e.toString());
        }
    }
}
```

27



Ajax primjer 2 – podaci: ajax_2.php

```
<?php
$url = "ajax_1.xml";

if(! ($fp = fopen($url,'r'))) {
    echo "Problem: nije moguće otvoriti url: " . $url;
    exit;
}

// XML data
$xml_string = fread($fp, 10000);
fclose($fp);

// create a DOM object from the XML data
$domdoc = new DomDocument;
$domdoc->loadXML($xml_string);

header("Content-Type: text/xml");
echo $domdoc->saveXML();
?>
```

važno je postaviti

28



Ajax primjer 3. - slanje XML-a

```
function dajPodatke()
{
    if(xmlHttp) {
        try {
            if (xmlHttp.readyState == 4 || xmlHttp.readyState == 0) {
                xmlHttp.open("GET", "ajax_3.php", true);
                xmlHttp.onreadystatechange = preuzmiPodatke;
                xmlHttp.send(null);
            } else { // ako je XMLHttpRequest objekt zauzet...
            }
        } catch(e) {
            alert("Problem kod povezivanja na server:\n" + e.toString());
        }
    }
}
```



Ajax primjer 3 – podaci: ajax_3.php

```
<?php
$dbc = mysqli_connect("localhost", "WebDiP2013", "FOI", "PzaWeb");
if(! $dbc) {
    echo "Problem kod povezivanja na bazu podataka!";
    exit;
}

mysqli_set_charset($dbc, "utf-8");

$output = '<?xml version="1.0" encoding="utf-8" standalone="yes"?>';
$output .= '<kolegiji>';

$query = 'SELECT studij, sifra, katedra, naziv ' .
        'FROM kolegiji ' .
        'ORDER BY naziv';
```



Ajax primjer 3 – podaci: ajax_3.php

```
$result = mysqli_query($dbc, $query);
if (mysqli_num_rows($result)) {
    while ($row = mysqli_fetch_row($result)) {
        $output .= '<name studij="' . $row[0] . '" sifra="' . $row[1] . '">' .
            $row[3] . '</name>' . "\n";
    }
    $output .= '</kolegiji>';
    mysqli_close($dbc);
    header("Content-Type: text/xml");
    print $output;
?>
```



Ajax primjer 4 – prikaz rada

```
function dajPodatke()
{
    if(xmlHttp) {
        try {
            if (xmlHttp.readyState == 4 || xmlHttp.readyState == 0) {
                xmlHttp.open("GET", "ajax_4.php", true);
                xmlHttp.onreadystatechange = preuzmiPodatke;
                xmlHttp.send(null);
            } else { // ako je XMLHttpRequest objekt zauzet...
            }
        } catch(e) {
            alert("Problem kod povezivanja na server:\n" + e.toString());
        }
    }
}
```



Ajax primjer 4 – prikaz rada

```
var window2 = window.open("", "", "scrollbars=no, width=300, height=840");
window2.moveTo(700, 0);

function preuzmiPodatke()
{
    window2.document.write("Vrijeme: " + vrijeme() + " Status: " +
        xmlhttp.readyState + "<br>");
    if (xmlhttp.readyState == 4) {
        if (xmlhttp.status == 200) { // samo ako je HTTP status "OK"
            try {
                prikaziPodatke();
            } catch(e) {
                alert(e.toString());
            }
        } else {
            alert("Problem kod preuzimanja podataka:\n" + xmlhttp.statusText);
        }
    }
}
```



Ajax primjer 4 – prikaz rada

```
function vrijeme() {
    var d = new Date();
    return(d.getDay() + "." + d.getMonth() + "." + d.getFullYear() + " " +
        d.getHours() + ":" + d.getMinutes() + ":" + d.getSeconds() + "." +
        d.getMilliseconds());
}
```



Ajax primjer 4 – podaci: ajax_4.php

```
$output = '<?xml version="1.0" encoding="utf-8" standalone="yes"?>' . "\n";
$output .= '<kolegiji>' . "\n";

$query = 'SELECT studij, sifra, katedra, naziv ' .
        'FROM kolegiji ORDER BY studij, naziv';

$result = mysqli_query($dbc, $query);
if (mysqli_num_rows($result)) {
    ob_start();
    header("Content-Type: text/xml");
    print $output;
    while ($row = mysqli_fetch_row($result)) {
        $output = '<name studij="' . $row[0] . '" sifra="' .
                $row[1] . '">' . $row[3] . '</name>' . "\n";
        print $output;
    }
    $output = '</kolegiji>';
    print $output;
}
mysqli_close($dbc);
?>
```



Ajax primjer 5. - slanje HTML-a

```
function prikaziPodatke()
{
    var response = xmlhttp.responseText;
    if (response.indexOf("ERRNO") >= 0
        || response.indexOf("error:") >= 0
        || response.length == 0)
        throw(response.length == 0 ? "Prazan odgovor servera." : response);
    var inHTML= response;

    var oSuggest = document.getElementById("cilj");
    oSuggest.innerHTML = inHTML;
}
```

Nije u izvornom
AJAX duhu jer se
prenosi HTML
sadržaj



Ajax primjer 5. - podaci: ajax_5.php

```
<?php
echo "<select id=\"kolegij\" name=\"kolegij\" size=\"20\">
  <option value=\"BS#01\">BS / ENGLISKI JEZIK I</option>
  <option value=\"BS#24\">BS / ENGLISKI JEZIK II</option>
  <option value=\"BS#02\">BS / FINANSIJSKE INSTITUCIJE I TRŽIŠTE
  KAPITALA</option>
  <option value=\"BS#03\">BS / INFORMATIKA I</option>
  <option value=\"BS#04\">BS / INFORMATIKA II</option>
  <option value=\"BS#05\">BS / IS U UREDSKOM POSLOVANJU</option>
  <option value=\"BS#06\">BS / IS ZAŠTITE OKOLIŠA</option>
  <option value=\"BS#07\">BS / KOMUNICIRANJE U ORGANIZACIJI</option>
  <option value=\"BS#08\">BS / MATEMATIKA I</option>
  <option value=\"BS#09\">BS / MATEMATIKA II</option>
  <option value=\"BS#10\">BS / NJEMACKI JEZIK I</option>
  <option value=\"BS#11\">BS / NJEMACKI JEZIK II</option>
</select>";
?>
```



Ajax primjer 6. - pretraživanje podataka

```
<body>
  <form name="forma">
    Naziv kao: <input id="naziv" name="naziv" maxlength="20"
    onkeyup="dajPodatke(this.value);">
  </form>
  <div id="cilj"></div>
</body>
```

Nakon otpuštanje tipke poziva se funkcija dajPodatke kojoj se prenosi upisani naziv za traženje podataka



Ajax primjer 6. - slanje XML-a

```
function dajPodatke(naziv)
{
    if(xmlHttp) {
        try {
            if (xmlHttp.readyState == 4 || xmlHttp.readyState == 0) {
                xmlHttp.open("GET", "ajax_6.php?naziv=" + naziv, true);
                xmlHttp.onreadystatechange = preuzmiPodatke;
                xmlHttp.send(null);
            } else { // ako je XMLHttpRequest objekt zauzet...
            }
        } catch(e) {
            alert("Problem kod povezivanja na server:\n" + e.toString());
        }
    }
}
```



Ajax primjer 6 – podaci: ajax_6.php

```
<?php
$dbc = mysqli_connect("localhost", "WebDiP2013", "FOI", "PzaWeb");
if(! $dbc) {
    echo "Problem kod povezivanja na bazu podataka!";
    exit;
}

$result = mysqli_query($dbc, $query);

$output = '<?xml version="1.0" encoding="utf-8" standalone="yes"?>';
$output .= '<kolegiji>';

$query = 'SELECT studij, sifra, katedra, naziv FROM kolegiji ';
if(isset($_GET["naziv"])) {
    $query .= ' WHERE naziv like "%" . $_GET["naziv"] . "%" ';
}
$query .= 'ORDER BY studij, naziv';
```



Ajax primjer 6 – podaci: ajax_6.php

```
$result = mysqli_query($dbc, $query);
if (mysqli_num_rows($result)) {
    while ($row = mysqli_fetch_row($result)) {
        $output .= '<name studij="' . $row[0] . '" sifra="' . $row[1] . '">' .
            $row[3] . '</name>' . "\n";
    }
    $output .= '</kolegiji>';
    mysqli_close($dbc);
    header("Content-Type: text/xml");
    print $output;
?>
```



JSON

- JSON:
 - ⇒ JavaScript Object Notation - JavaScript-no označavanje objekta
 - ⇒ Lagani tekstu orijentirani format podataka
 - ⇒ Neovisan o jeziku
 - ⇒ Samoopisujući
 - ⇒ Lagan za razumijevanje



JSON

- Sintaksa:

- ⇒ podaci su u obliku parova naziv/vrijednost
- ⇒ podaci su odvojeni zarezom
- ⇒ vitičaste zagrade sadrže objekte
- ⇒ pravokutne zagrade sadrže nizove



JSON

- JSON vrijednosti mogu biti:

- ⇒ brojevi (cjeloborni ili decimalni)
- ⇒ string (u dvostrukim navodnicima)
- ⇒ Boolean (true, false)
- ⇒ niz (u pravokutnim zagradaama)
- ⇒ object (u vitičastim zagradaama)
- ⇒ null



JSON

```
[
  {"studij": "BS", "sifra": "01", "naziv": "ENGLESKI JEZIK I"},
  {"studij": "BS", "sifra": "24", "naziv": "ENGLESKI JEZIK II"},
  {"studij": "BS", "sifra": "02", "naziv": "FINANCIJSKE INSTITUCIJE I TR\u0017dI\u00160TE KAPITALA"},
  ...
]
```



Ajax primjer 7. - slanje JSON-a

```
function dajPodatke(naziv)
{
  if(xmlHttp) {
    try {
      if (xmlHttp.readyState == 4 || xmlHttp.readyState == 0) {
        xmlHttp.open("GET", "ajax_7.php", true);
        xmlHttp.onreadystatechange = preuzmiPodatke;
        xmlHttp.send(null);
      } else { // ako je XMLHttpRequest objekt zauzet...
      }
    } catch(e) {
      alert("Problem kod povezivanja na server:\n" + e.toString());
    }
  }
}
```



Ajax primjer 7. - slanje JSON-a

```
function prikaziPodatke()
    var response = xmlhttp.responseText;

    if (response.indexOf("ERRNO") >= 0
        || response.indexOf("error") >= 0
        || response.length === 0)
        throw(response.length === 0 ?
            "Prazan odgovor servera." : response);

    nameArray = JSON.parse(xmlhttp.responseText);

    var inHTML = '<select id="kolegij" name="kolegij" size="20">';

    for (i = 0; i < nameArray.length; i++) {
        inHTML += '<option value="' + nameArray[i].studij +
            "#" + nameArray[i].sifra + ">' +
            nameArray[i].studij + " / " +
            nameArray[i].naziv + '</option>';
    }
    inHTML += '</select>';
    var oSuggest = document.getElementById("cilj");
    oSuggest.innerHTML = inHTML;
}
```

Pretvaranje
dobivenog
teksta u JSON
podatak



Ajax primjer 7 – podaci: ajax_7.php

```
<?php
$dbc = mysqli_connect("localhost", "WebDiP2013", "FOI", "PzaWeb");
if(! $dbc) {
    echo "Problem kod povezivanja na bazu podataka!";
    exit;
}

mysqli_set_charset($dbc, "utf-8");

$query = 'SELECT studij, sifra, katedra, naziv ' .
    'FROM kolegiji ' .
    'ORDER BY naziv';
```



Ajax primjer 7 – podaci: ajax_7.php

```
$odgovor = array();
$result = mysqli_query($dbc, $query);
if (mysqli_num_rows($result)) {
    while ($row = mysqli_fetch_row($result)) {
        $redak["studij"] = $row[0];
        $redak["sifra"] = $row[1];
        $redak["naziv"] = $row[3];
        $odgovor[] = $redak;
    }
}
$output .= '</kolegiji>';
mysqli_close($dbc);
header("Content-Type: application/json");
print json_encode($odgovor);
?>
```



AJAX biblioteke

- Važnost AJAX može se vrednovati i u velikom broju biblioteka koje su nastale u relativno kratkom razdoblju
 - ⇒ jQuery
 - ⇒ MooTools
 - ⇒ jMaki
- Također postoji veliko broj aplikacijskih okvira (framework) za AJAX kojima se pojednostavljuje razvoj RIA
 - ⇒ Prototype
 - ⇒ Angular



Korištena i dodatna literatura

- ⇒ <http://www.w3.org/TR/XMLHttpRequest/>
- ⇒ <http://www.ajaxmatters.com/>
- ⇒ <http://java.sun.com/developer/technicalArticles/J2EE/AJAX/DesignStrategies/>
- ⇒ <http://vyk1.spaces.live.com/Blog/cns!EBE3A761F939F926!1051.entropy>
- ⇒ <http://blog.coderlab.us/rasmus-30-second-ajax-tutorial/>
- ⇒ http://www.w3schools.com/dom/dom_http.asp
- ⇒ <http://developer.mozilla.org/en/docs/AJAX>
- ⇒ http://developer.mozilla.org/en/docs/Gecko_DOM_Reference
- ⇒ <http://www.xul.fr/en-xml-ajax.html>
- ⇒ <https://ajax.dev.java.net/>
- ⇒ <http://www.telerik.com/help/desktopbeta/>

