



Prirodoslovno-matematički fakultet
Matematički odsjek
Sveučilište u Zagrebu

RAČUNARSKI PRAKTIKUM II

Predavanje 01 - Uvod

3. ožujka 2020.

Sastavio: Zvonimir Bujanović



Cilj kolegija:

- Naučiti osnove web-programiranja kroz nekoliko tehnologija koje se danas najčešće koriste.
- Timskim radom izraditi nešto kompleksniji projekt.

Pretpostavljeno predznanje (iz kolegija Mreže računala):

- HTML za izradu statičnih web stranica,
- CSS za primjenu stilova u okviru web stranica.

Grubi pregled sadržaja kolegija:

- PHP – skriptni jezik za web-programiranje na strani servera,
- MySQL – veza serverskog web-programiranja s bazama podataka,
- JavaScript – skriptni jezik za web-programiranje na strani klijenta.

Razmotrit ćemo i druge aktualne web tehnologije.

Predavanja i vježbe – Zvonimir Bujanović (zbujanov@math.hr)

- Termin predavanja: utorkom, 13h-14h u (PR2)
- Termin vježbi: utorkom, 14h-16h u (PR2)
- Termin konzultacija: četvrtkom, 11h-13h u (A307)

Iz ovog kolegija se ne održavaju demonstrature.

Web-stranica kolegija: <https://www.math.hr/nastava/rp2d>

Kolegij nosi 5 ECTS bodova.

Elementi ocjenjivanja:

- Dva kolokvija, na računalima – ukupno $25+25 = 50$ bodova.
- Domaće zadaće – $4*4 = 16$ bodova (ili više).
- Aktivnost na nastavi – 4 boda.
- Projektni zadatak – 30 bodova.

Uvjet za polaganje kolegija:

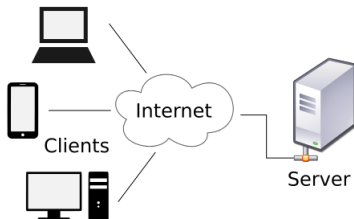
- Ukupno barem 50 bodova.
- Od toga barem 25 na kolokvijima.
- Od toga barem 5 iz zadaća i aktivnosti.

Projektni zadatak:

- Složenija web-aplikacija koja tipično treba uključivati i klijentski i serverski aspekt web-programiranja.
- Rješava se u grupi od najmanje dvoje sudionika.
- Temu treba odabrati do prvog kolokvija; predaja (obično) nakon drugog kolokvija.
- Za izradu koristiti **Git** i tzv. **MVC pattern** (osim u opravdanim i dogovorenim slučajevima).
- Prije obrane treba poslati **kratku prezentaciju** (10ak slajdova), **[link na Git repozitorij](#)**.
- Demonstracija i usmena obrana projekta; svaki student treba prezentirati svoj doprinos.
- Popis potencijalnih tema za projektne zadatke je objavljen na webu. Originalni prijedlozi mogu donijeti dodatnih 5 bodova!
- Elementi ocjenjivanja: složenost aplikacije, kvaliteta implementacije, dokumentacija, izlaganje.

Organizacija web-aplikacija: ponavljanje

Web-aplikacije poštuju klasičnu **klijent-server** paradigmu.



Statične web-stranice:

- Server čuva HTML i CSS datoteke, kao i ostale (lokalne) datoteke na koje pokazuje web-stranica (slike, PDF...).
- Klijent se spoji HTTP protokolom na adresu servera, tipično port 80.
- Klijent zahtjeva konkretne HTML, CSS i ostale datoteke.
- Server pošalje klijentu tražene datoteke.
- Web-browser na klijentu prikaže "downloadane" datoteke.

HTTP: primjer komunikacije

- Klijent želi dohvatiti web-stranicu s adrese `http://www.nekoracunalo.com/putanja/datoteka.html`.
- TCP protokolom spaja se na port 80 računala na adresi `www.nekoracunalo.com` (DNS!)
- Nakon spajanja klijent šalje zahtjev poput ovog:

```
GET /putanja/datoteka.html HTTP/1.1  
[prazna linija]
```

- Server šalje odgovor poput:

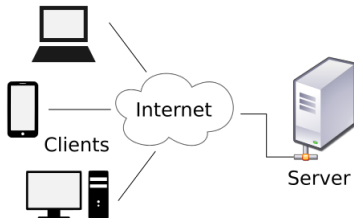
```
HTTP/1.1 200 OK  
Date: Sat, 10 Dec 2016 23:59:59 GMT  
Content-Type: text/html  
Content-Length: 1354  
<html> <body>  
<h1>Moja osobna web stranica!</h1>  
(nastavlja se sadržaj datoteke) ...  
</body>  
</html>
```

Aktualni standard protokola je HTTP/1.1.

U veljači 2015. je predložen novi standard HTTP/2.

Neki HTTP/1.1 zahtjevi:

- **GET** – klijent želi dohvatiti dokument; server odgovara slanjem statusnih informacija iz kojih slijedi kopija traženog dokumenta.
- **HEAD** – klijent želi dohvatiti samo statusne informacije o nekom dokumentu; server odgovara slanjem istih (bez kopije dokumenta).
- **POST** – klijent šalje podatke serveru, te želi da se postojeći dokument “proširi” tim podacima. Tipično će poslane podatke obrađivati neka skripta na serveru.
- **PUT** – klijent šalje podatke serveru i želi da se ti podaci spremaju na prosljeđenu adresu.
- **DELETE** – klijent želi obrisati određeni resurs sa servera.



Aktivne web-stranice = skriptiranje na strani klijenta:

- Server pošalje klijentu tražene HTML, CSS i ostale datoteke. Te datoteke sadrže i neki program.
- Web-browser na klijentu prikaže "downloadane" datoteke, te izvršava program unutar web-browsera na klijentskom računalu.
- Zbog interakcije programa s korisnikom, korisnik ima dojam da je web-stranica "aktivna".

Osnovna varijanta:

- Program se u potpunosti izvršava na klijentu.
- Program nakon "downloadanja" nema više nikakvu interakciju sa serverom.

Tipični programski jezici za klijentsko skriptiranje:

- **JavaScript**, ~~ActionScript (Adobe Flash), Java applet~~i

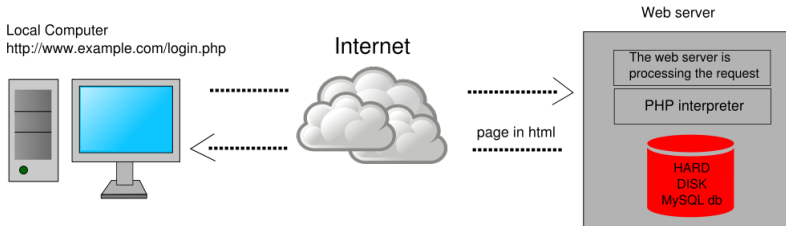
Uoči: **svaki klijent** treba imati podršku za programski jezik!

- Svi moderni browseri podržavaju JavaScript *out-of-the-box*.
- ~~Adobe Flash i Javu najčešće treba manualno instalirati.~~

Skriptiranje na strani klijenta: JavaScript primjer

```
1 <!DOCTYPE html>
2 <html lang="hr">
3 <head>
4   <script>
5     function zbroji()
6     {
7       var broj1 = document.getElementById( "broj1" );
8       var broj2 = document.getElementById( "broj2" );
9       var rezultat = document.getElementById( "rezultat" );
10
11       var br1 = new Number( broj1.value );
12       var br2 = new Number( broj2.value );
13
14       rezultat.value = br1 + br2;
15     }
16   </script>
17 </head>
18
19 <body>
20   <h3>Zbrajanje</h3>
21
22   <form>
23     <p>
24       <input type="text" id="broj1" />
25       <input type="button" id="plus" value="+" onclick="zbroji()" />
26       <input type="text" id="broj2" /> = <input type="text" id="rezultat" />
27     </p>
28   </form>
29 </body>
30 </html>
```

Skriptiranje na strani servera



Dinamičke web-stranice = skriptiranje na strani servera:

- Klijent nakon spajanja na server šalje i izvjesne **parametre** (na primjer, korisničko ime ili lokaciju).
- Server **na temelju dobivenih parametara generira** HTML, CSS i ostale datoteke, te ih pošalje klijentu.
- Web-browser na klijentu prikaže "downloadane" datoteke.
- Zbog činjenice da za različite parametre server generira različite datoteke, govorimo o "dinamičkim" web-stranicama.

Osnovna varijanta:

- Skripta koja generira HTML se u potpunosti izvršava na serveru.
- Nakon što klijent “downloada” generirane datoteke, više nema nikakve interakcije sa serverom.

Tipični programski jezici za serversko skriptiranje:

- **PHP**, ASP.NET, Java, Ruby, Python (Django), JavaScript (Node.js)

Uoči: klijenti dobivaju generirani HTML i ne trebaju podršku za jezik sa servera!

Skriptiranje na strani klijenta: PHP primjer

index.php:

```
1 <!DOCTYPE html>
2 <html lang="hr">
3 <head></head>
4 <body>
5     <p>
6         <?php
7             echo "Hello, " . $_GET['username'] . "!";
8         ?>
9     </p>
10 </body>
11 </html>
```

Generirani HTML dokument za index.php?username=Ana:

```
1 <!DOCTYPE html>
2 <html lang="hr">
3 <head></head>
4 <body>
5     <p>
6         Hello, Ana!
7     </p>
8 </body>
9 </html>
```

U tipičnoj web-aplikaciji situacija je složenija:

- Serverska skripta obično dohvaća podatke iz baze podataka:
 - popis korisnika, provjera lozinke
 - popisi proizvoda i njihovih cijena
- Klijentska skripta obično više puta kontaktira server:
 - Google search suggest
 - GMail
 - Google Maps

Za komunikaciju klijentske skripte i servera koristi se tehnika zvana *Ajax*.

Skriptiranje na strani servera: PHP

- 1 Osnove PHP-a kao programskog jezika
- 2 Superglobalne varijable
- 3 PHP sessions
- 4 Rad s MySQL bazama u PHP-u
- 5 Struktura moderne web-aplikacije (MVC)

Skriptiranje na strani klijenta: JavaScript

- 1 Osnove JavaScripta kao programskog jezika
- 2 HTML Document Object Model (DOM)
- 3 Ugrađene JavaScript biblioteke: Geolocation/Canvas/Web Workers/Local Storage
- 4 Eksterne JavaScript biblioteke: jQuery, OpenStreetMap
- 5 Ajax i XML/JSON