

Tehnologije za Web



Kaskadni stilovi (CSS)



- ❑ HTML je izvorno projektovan kao strukturni jezik – nije napravljen da upravlja izgledom stranice, već se pretpostavljalo:
 - ❖ Autor zadaje elemente stranice
 - ❖ Čitač Weba određuje kako će oni na ekranu izgledati
- ❑ Sa podrškom za fontove, boje i druge vizuelne efekte, HTML je postao alatka za dizajniranje Web strana
- ❑ Kaskadni stilovi – uvedeni su s namerom da se jeziku HTML vrati prvobitna namena – strukturiranje Web strana
- ❑ Pomoću kaskadnih stilova odvaja se struktura od izgleda Web strane

CSS

- ❑ **Kaskadni opisi stilova (CSS)** predstavljaju skup odrednica (pravila) pomoću kojih se u potpunosti upravlja rasporedom i izgledom stranice
- ❑ Opisi stilova omogućavaju pisanje naredbi za izgled i formatiranje u zaglavlju Web stranice ili u spoljašnjoj datoteci, van HTML koda kojim se zadaje sadržaj stranice
- ❑ Nasuprot upotrebi tabela za raspoređivanje elemenata stranice, CSS pravila su jednostavni tekstualni iskazi, pomoću kojih istovremeno može da se raspoređuje i formatira jedan ili više elemenata
- ❑ Stranica na kojoj su razdvojeni sadržaj i izgled lakše se održava i ažurira

3

Specifikacije kaskadnih stilova

- ❑ Specifikacija kaskadnih stilova je veoma obimna i njoj se neprekidno dodaju nova svojstva

Verzija	Prikaz svojstava
CSS1	Upravljanje tekстом, uključujući mogućnost zadavanja fontova , veličine slova, stila i razmaka. Mogućnost zadavanja boje pozadine i pozadinske slike . Podešavanje margin a, ivica i razmaka između teksta i objekata. Podrška za stilove za liste .
CSS2	Poboljšana svojstva vezana za štampanje . Podrška za dinamičko iscrtavanje, utisnute fontove, pozicioniranje elemenata (CSSP), kaskadne stilove sa XML-om. Podrška kontroli okruženja – mogućnost izmene oblika pokazivača miša. Ograničeni postupci, na primer efekat lebdenja kod hiperveza.
CSS3	Donosi podršku za vertikalno ispisani tekst i dizajn u više kolona . Poboljšana je podrška za pridruživanje postupaka i stilova. Integrirana je tehnologija grafike, boje i fontova.

4

Metode izrade HTML dokumenata i kaskadnih stilova



Metod	Primer	Prednosti	Mane
Ručno	Pisanje koda u Notepadu	Autor ima veliku kontrolu nad kodom. Moguće je odmah pristupiti greškama, novim HTML elementima i CSS svojstvima.	Metod je spor. Podložan je greškama. Zahteva odlično poznavanje HTML elemenata i svojstava CSS. Nema mogućnost direktne vizuelne kontrole.
Prevođenje	Snimanje HTML dokumenta iz drugog programa , npr. Worda	Brz. Pojednostavljuje objavljivanje postojećih dokumenata.	Kod nastao na ovaj način obično sadrži dosta grešaka. Zahteva dodatno pročišćavanje, dodavanje hiperveza.
Izrada strana u WYSIWYG programima	Program Dreamweaver .	Autor radi direktno sa grafičkim i tekstualnim objektima. Ne zahteva dobro poznavanje jezika HTML i CSS. Pruža podršku za CSS, JavaScript, ASP.NET, ColdFusion, JavaServer Pages, and PHP.	Fino podešavanje izgleda strane najčešće zahteva ručno uređivanje HTML koda.

*WYSIWYG - *What You See Is What You Get*

[Šta želite, to ćete i videti](#)

5

Programiranje i izrada Web strana



- ❑ Programiranje za Web može se podeliti u dve osnovne velike grupe: klijentsko i serversko
 - ❖ **Klijentske tehnologije** su one koje se izvršavaju u klijentu, tj. u čitaču, iako postoje i neke klijentske tehnologije, poput JavaScripta i ActiveX kontrola, koje se izvršavaju na klijentskom računaru, ali izvan čitača
 - ❖ Programi koji se izvršavaju na serveru su **serverske aplikacije**
- ❑ Svakoj tehnologiji pripada određeno mesto i primena, njihovo kombinovanje je najbolje rešenje

6

Tehnologije za klijentsko i serversko programiranje Web strana



Klijentske tehnologije	Serverske tehnologije
ActiveX kontrole	CGI skriptovi i programi
Java programi	Serverski API programi (ISAPI/NSAPI)
<u>Jezici za pisanje skriptova</u> ❖ Java Script ❖ VBScript ❖ Dynamic HTML	<u>Serverski skriptovi</u> ❖ Active Server Pages ❖ ColdFusion ❖ PHP
Pomoćni programi	Serverski Java programi
Dodaci	

7

Klijentsko programiranje



Klijentsko programiranje

- ❑ Programi koji se izvršavaju na klijentskoj strani mnogo su brži od programa na serverskoj strani
 - ❖ Nije potrebno prenositi podatke preko mreže da bi se prikazao rezultat akcije
- ❑ Klijentsko programiranje – nedostatak kontrole
 - ❖ Teško je odrediti kakvi će korisnici pristupati lokaciji
 - ❖ Koji će čitač korisnici koristiti
 - ❖ Koja će svojstva čitača biti uključena
 - ❖ Koju vrstu procesora korisnik upotrebljava
 - ❖ Postoje korisnici koji isključuju podršku za skriptove ili koriste programe koji se tek testiraju

9

Klijentsko programiranje

Klijentsko programiranje nije uvek efikasno

Pretpostaviti: da će sve ispravno raditi, ali napraviti i rezervnu varijantu

10

Active X

- ❑ **ActiveX** je **Microsoft-ova** tehnologija za izradu malih komponenti (kontrola) u okviru Web strane – nije programski jezik, već mnoštvo integrisanih objekata napisanih na različitim jezicima u Windowsovom okruženju.
- ❑ **ActiveX** kontrole mogu biti programirane u više različitih programskih jezika, uključujući **Javu, Delphi, Visual Basic, C++ i druge.**
- ❑ **ActiveX** objekti (kontrola) omogućavaju dodavanje programa Web stranici za rešavanje mnoštva zadataka: od prikazivanja poruka do generisanja zvučnih izlaza za tekst ili npr. mogućnosti prepoznavanja govora u programima i Web stranicama.

11

ActiveX

- ❑ **ActiveX** kontrole mogu biti izvršavane samo na **Windows** operativnim sistemima.
- ❑ **ActiveX** objekti nisu bezbedni, jer moraju da se nalaze na korisnikovom disku da bi ga bilo moguće koristiti - imaju pristup svim resursima sistema
- ❑ Zlonamerni **ActiveX** objekat može uneti virus i oštetiti ili uništiti podatke na disku korisnika

12

ActiveX



- ❑ Ako korisnik poseti Web stranicu koja koristi ActiveX objekat, a nema instaliran objekat na svom disku, Web server će pokušati da ga pošalje na korisnikov računar
 - ❖ Zavisno od korisnikovih bezbednosnih podešavanja, čitač Weba će prikazati okvir za dijalog, koji upozorava korisnika o preuzimanju ActiveX objekta
- ❑ Preporuka: koristiti samo svoje objekte, one koje su napravili poznati i pouzdani programeri i one koji su preuzeti sa pouzdanih Web lokacija (kao što je Microsoft)

13

Java



- ❑ Tehnologija kompanije Sun Microsystems
- ❑ Java je razvojni jezik nezavisan od platforme, koji omogućava pisanje programa i njegovu primenu na bilo kom sistemu, čitaču ili OS koji podržava Java Virtualnu Mašinu (JVM)
- ❑ Java jezik je po sintaksi donekle sličan jeziku C++

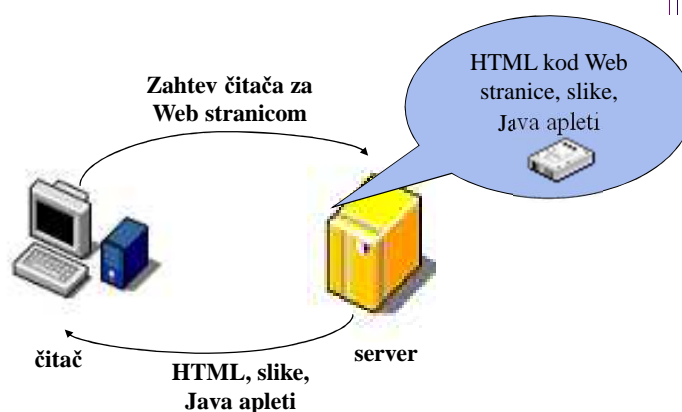
14

Java apleti

- ❑ Web strane koriste male programe pisane na jeziku Java, koji se nazivaju *apleti* i koji se preuzimaju i izvršavaju u samom čitaču proširujući njegove mogućnosti
- ❑ Apleti se pišu na jeziku Java,
 - ❖ prevode u bajt kod, koji je nezavisan od računara na kojem se izvršava,
 - ❖ u obliku datoteke tipa .class koja se automatski preuzima u čitač koji podržava Javu i
 - ❖ pokreće u okruženju samog čitača

15

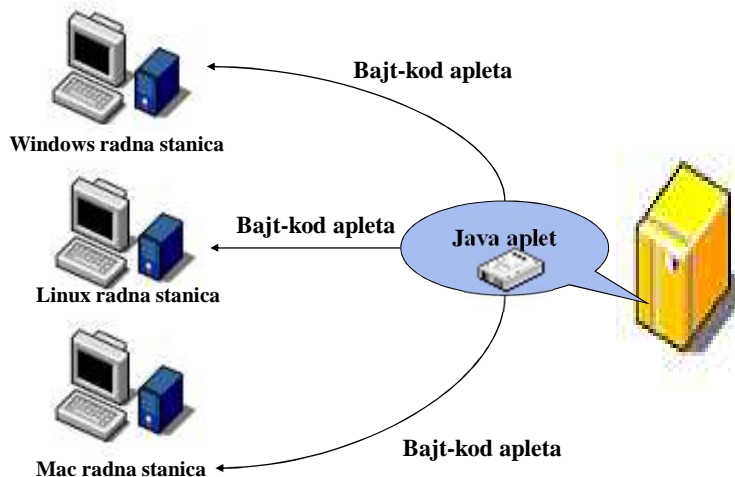
Čitač preuzima Java aplete sa Web servera



16

Java apleti

Java apleti koriste generički bajt-kod koji čitač prevodi u kod specifičan za mašinu – isti aplet radi na raznim tipovima mašina



17

Java apleti

- ❑ Čak i na računarima sa brzim procesorom može izgledati kao da se bajt kod izvršava sporije nego u normalno prevedenim programima
- ❑ JVM pre izvršavanja mora da interpretira bajt kod apleta
- ❑ Zbog toga vlada mišljenje da je Java spora
- ❑ Zapravo njena interpretacija može biti spora

18

Problemi s Javinim generičkim bajt-kodom



- ❑ Problem s Javinim generičkim bajt-kodom je što čitač mora da prevede programski kod u program specifičan za procesor svaki put kada se aplikacija pokrene
- ❑ Dodatni proces prevođenja izaziva **kašnjenje** pre pokretanja programa
- ❑ Svaki put kada korisnik pokrene aplet čitač mora da ga preuzme, dok je ActiveX objekat čitač morao da preuzme samo jednom

19

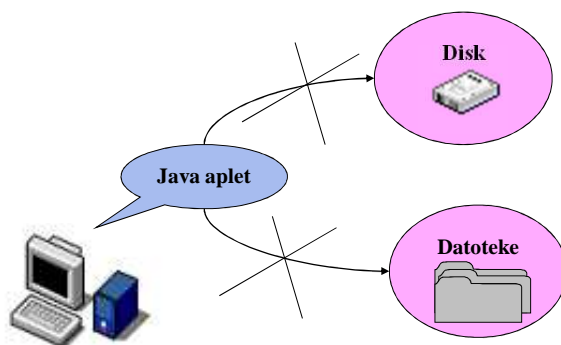
Bezbednost Java apleta



- ❑ Java apleti nisu samostalni izvršni programi – oni se pokreću samo u okviru Web čitača
- ❑ **Javin bezbednosni model** štiti korisnike od virusa i drugih opasnosti iz programa preuzetih s Weba
- ❑ Radi eliminisanja virusa i krađe podataka Java apleti:
 - ❖ ne mogu da pristupe resursima (posebno datotekama) koji se nalaze na korisnikovom disku,
 - ❖ ne mogu da pristupe svojstvima sistema,
 - ❖ ne učitavaju datoteke,
 - ❖ niti otvaraju mrežnu vezu sa bilo kojim računarom osim onog s kojeg su preuzeti.

20

Java apleti štite rad diska i datoteka



21

JavaScript

- ❑ **JavaScript** je skript jezik kojim Web dizajneri mogu da automatizuju postupke unutar Web stranice
- ❑ Skript jezik- jer omogućava pisanje skripta (niza naredbi) koje će Web čitač izvršiti
- ❑ Omogućava zadavanje grupe naredbi koje će čitač Weba izvršiti – ali se ne koristi za pravljenje spoljnog programa koji se izvršava nezavisno od čitača

22

JavaScript



- ❑ **JavaScript** se ugrađuje u HTML Web stranice – nije potreban poseban alat; može se koristiti isti program za uređenje teksta koji se koristi za izradu Web stranice
- ❑ Najpopularniji čitači Weba imaju ugrađenu podršku za JavaScript – posetioци lokacije ne moraju instalirati dodatne programe na svoje računare
- ❑ JavaScript iskazi postavljaju se između oznaka `<script>` i `</script>`, koje će ukazati čitaču Weba da izvrši JavaScript iskaze umesto da prikaže tekst na ekranu

23

JavaScript



- ❑ Iskazi skripta mogu se uokviriti HTML oznakama za komentar (`<!--...-->`) kako bi se sprečilo da čitači koji ne podržavaju JavaScript na ekranu prikažu iskaze skripta
- ❑ Čitači koji ne podržavaju JavaScript ignorisaće oznake skripta i sve što se nalazi između početne i završne oznake komentara
- ❑ Obrnuto, čitači koji podržavaju JavaScript ignorisaće oznake komentara i izvršiti funkciju skripta

24

Sakrivanje skripta od čitača koji ne podržavaju JavaScript



```
<script language="JavaScript">
<!--
Document.bgColor="lightyellow";
Document.fgColor="magenta";
//-->
</script>
```

25

Pomoćni programi



- Program koji će čitač pozvati u pomoć kada naiđe na tip podataka koji ne može da obradi
- Pod pretpostavkom da se programu može pridružiti MIME tip, svaki program može biti pomoćni program
- Kada se objekat postavi na Web dodeljuje mu se HTTP zaglavlje sa informacijama o tipu – te informacije date su u MIME tipu
- MIME type se određuje na osnovu ekstenzije fajla (na primer .html, ili .gif). Standardni skup MIME type je definisan u fajlu /usr/local/etc/httpd/conf/mime.types

26

Pomoćni programi



- ❑ Na primer. Svaki dokument načinjen u programu Acrobat trebalo bi da ima pridružen MIME tip **application/pdf**.
- ❑ Kada čitač primi datoteku sa određenim MIME tipom, postupiće u skladu sa tim kako je podešen:
 - ❖ Snimanje datoteke na disk
 - ❖ Brisanje datoteke
 - ❖ Prosleđivanje datoteke drugom programu
- ❑ Pomoću MIME tipova i pomoćnog programa autor Web lokacije može da postavi Wordov dokument koji će korisnici moći automatski da preuzmu i pregledaju, ako na računaru imaju instaliran Word

27

Dodatni programi za čitače



- ❑ Dodaci su uvedeni da bi se rešio problem komunikacije i integracije koji je postojao kod pomoćnih programa koji nisu integrisani u čitač Weba već se izvršavaju u zasebnim prozorima i vrlo loše komuniciraju sa čitačem
- ❑ Dodaci su komponente koje se izvršavaju u okviru čitača, pa se zato mogu lako ugraditi u Web stranu
- ❑ Lako komuniciraju sa čitačem putem tehnologije kakva je JavaScript

28

Dodatni programi za čitače

- ❑ Proširivanje čitača pomoću dodataka ima i mane – korisnici moraju pronaći i preuzeti dodatak. Instalirati ih i ponovo pokrenuti čitač.
- ❑ Za mnoge korisnike – komplikovano.
- ❑ Da bi se olakšao postupak instaliranja neki popularni dodaci poput dodatka za format Flash kompanije, postali su sastavni deo instalacije samog čitača.
- ❑ Standardni dodaci – reprodukovanju multimedijjskih sadržaja
- ❑ Primeri: Adobe Flash, Shockwave, Adobe PDF, RealVideo, QuickTime, RealPlayer...

29

http://www.blic.rs/zanimljivosti.php?id=120509

Inbox (1687) - Yahoo! ... B Blic Online | Najtražnije... METAK - Englesku rečn... Visoka poslovna škola ... Englesko-srpski rečnik ... Hipotekarna kriza - pon... My Gym Partner a

Fotografsanje je obavljeno danas uoči početka razgovora delegacije Srbije koju predvodi predsednik Boris Tadić. Ansa navodi da je Šutanovac tom prilikom dobio mesto na krajnoj levoj strani, daleko od italijanskog premijera.

Microsoft Silverlight
DOWNLOAD
Preuzmi poslednju verziju Microsoft Silverlight dodatka da pogledaš ovaj video

Facebook

Akcija - LCD TV 36.499 RSD
Šok cena LCD TV Telefunken. Dijagonala ekrana 32inča (80 cm). Vaš CBT. www.belatehnika.rs/
Čitajte u novom broju:

Team Building Agencije Da li su Vaši zaposleni u istom timu?
BC Group MegaStore Online AKCIJA: ASUS K501J-SX036L za samo 32.925 dinara

Postoji 141 komentar (53) Pošaljite komentar

ŠTAMPANO IZDANJE - 14. NOVEMBAR 2009. »

- » Mnogi prinuđeni da rade prekovremeno bez naknade
- » Dodik zabravio vrata zaposlenima u Vladi
- » Đuković odlazi ukoliko mladi ne pobeđe Norvešku

Štampano izdanje

Najčitanije Najviše komentara Najviše poslath

- » Berlusconi premestio Šutanovca zbog visine /video/
- » Dnevno krali dva automobila
- » Brisel: U Evropu bez visa
- » Policajcu sedam godina zatvora za ubistvo
- » Odbegli „zemunac“ zna ko je ubio Slavka Čuruviju

Blic online facebook

PRIDRUŽI SE BlicOVOJ GRUPI NA FEJSBUKU

Blic on Facebook
Become a Fan

Blic on Facebook

Dinamičko generisanje HTML-a



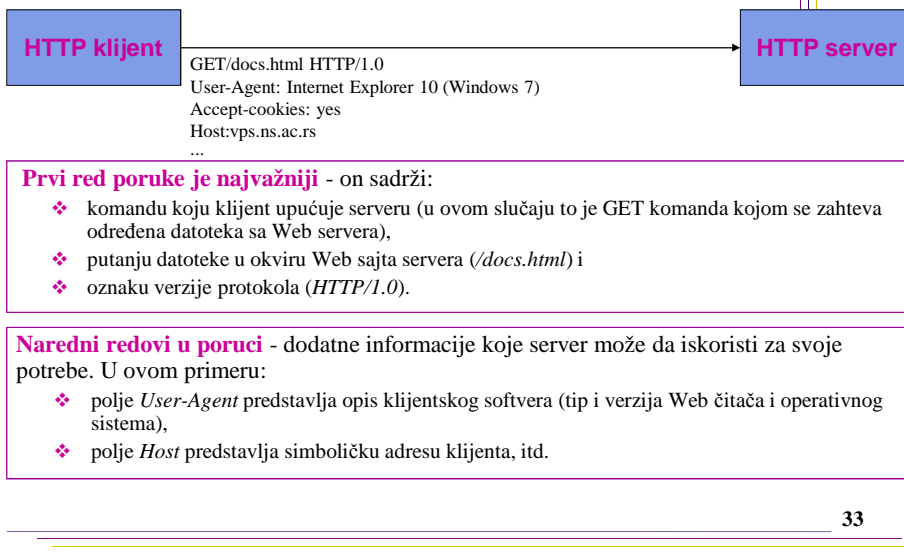
Dinamičko generisanje HTML-a



HTTP protokol

- ❑ Web čitači su namenjeni za prikazivanje Web stranica koje im isporučuju odgovarajući Web serveri.
- ❑ Struktura i izgled samih stranica se opisuje jezikom HTML.
- ❑ Komunikacija između Web klijenta (tj. čitača) i Web servera odvija se po standardnom HTTP (*HyperText Transfer Protocol*) protokolu.

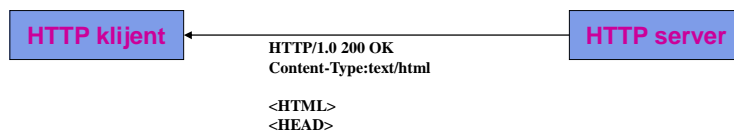
Slanje zahteva HTTP klijenta



Slanje odgovora HTTP klijentu

- ❑ Zadatak servera je da po prijemu zahteva HTTP klijenta odgovori na njega.
 - ❑ U ovom slučaju odgovor servera treba da sadrži traženu datoteku (**docs.html**), pri čemu je format odgovora takođe definisan HTTP protokolom.
- 34

Slanje odgovora HTTP klijentu

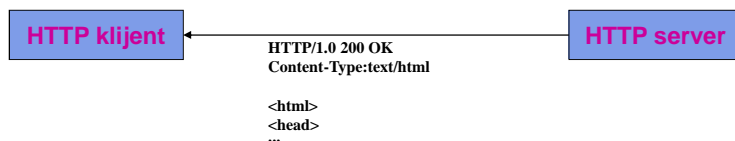


Prvi red odgovora sadrži:

- ❖ oznaku protokola, trocifreni broj koji predstavlja status izvršene operacije (200) i tekstualni opis tog statusa (OK). - Konstanta 200 označava da je zahtev uspešno izvršen i da se tražena datoteka nalazi u nastavku poruke.
- ❖ Druge konstante koje se češće sreću su:
 - 404 (tražena datoteka nije pronađena),
 - 407 (pristup datoteci nije dozvoljen), i
 - 302 (datoteka premeštena na drugo mesto).

35

Slanje odgovora HTTP klijentu



Naredni redovi u poruci sadrže:

- ❖ **Content-Type** - oznaka tipa sadržaja koji se vraća (standardizovane oznake propisane u odgovarajućim RFC dokumentima (*Request For Comments*) - zvanična serija dokumenata koja opisuje tehničke i organizacione detalje Interneta počevši od 1969
- ❖ Na primer:
 - HTML datoteke imaju oznaku **text/html**,
 - datoteke sa ASCII tekstem bez formatiranja imaju oznaku **text/plain**,
 - GIF slike **image/gif**, JPEG slike **image/jpeg**, itd.
- ❖ Jedan prazan red koji razdvaja zaglavlje odgovora od samog sadržaja datoteke.

36

Statički i dinamički Web sadržaji



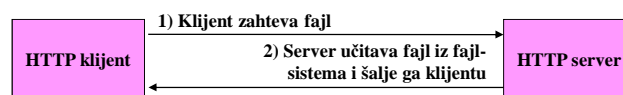
- ❑ Jedino što klijent može da zatraži od servera je **datoteka**.
- ❑ Na serveru je da tu datoteku pronade (eventualno i modifikuje!) i pošalje klijentu.
- ❑ Web sadržaji koji se smestaju na server vidljivi su klijentima kao pojedine datoteke:
 - ❖ datoteke mogu biti unapred pripremljene (npr. u editoru kakav je *Adobe Dreamweaver*) i smeštene u fajl-sistem Web servera.
 - ❖ Mogu biti i generisane “u letu” po prijemu zahteva klijenta na neki poseban način;
- ❑ U tom smislu, Web sadržaje (zapravo, datoteke) možemo podeliti na **statičke** i **dinamičke**.
- ❑ Klijent ne zna da li je datoteka koju je tražio generisana statički ili dinamički.

37

Statički Web sadržaji



- ❑ Statički Web sadržaji su datoteke koje su unapred smeštene u odgovarajući direktorijum fajl-sistema Web servera i spremne su za isporuku klijentima po njihovom zahtevu.

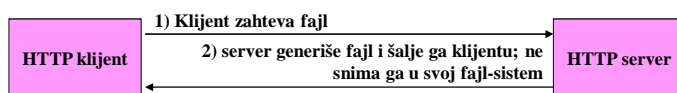


- Isporuka statičkih sadržaja -

38

Dinamički Web sadržaji

- ❑ Dinamički sadržaji nisu uskladišteni unapred već se generišu za svaki zahtev klijenta posebno.
- ❑ U ovom slučaju server neće tražiti datoteku u okviru fajl-sistema; server “zna” da je u pitanju dinamički generisana datoteka i poziva odgovarajući potprogram koji će je generisati.
- ❑ Najčešće nema potrebe ovako generisanu datoteku čuvati na serveru



- Isporuka dinamičkih sadržaja -

39

HTTP komunikacija

Ukupna sekvenca aktivnosti klijenta i servera u HTTP komunikaciji

1. klijent otvara konekciju sa serverom
2. klijent šalje zahtev serveru
3. server vraća odgovor
4. zatvara se konekcija

40

HTTP komunikacija



- ❑ Komunikacija između klijenta i servera zasnovana na **zahtev/odgovor** principu.
- ❑ Svaki par zahtev/odgovor smatra se nezavisnim od ostalih.
- ❑ Recimo, u slučaju da prvi klijent pošalje zahtev serveru i dobije odgovor, zatim drugi klijent pošalje zahtev i dobije odgovor, pa potom ponovo prvi klijent pošalje novi zahtev, nema načina da se ustanovi da je prvi klijent poslao dva zahteva (prvi i treći).
- ❑ Server svaki zahtev opslužuje nezavisno od ostalih zahteva.

HTTP je *stateless* protokol: ne omogućava praćenje stanja korisničke sesije između slanja više različitih zahteva.

41

Praćenje sesije korisnika



- ❑ HTTP protokol ne omogućava praćenje sesije korisnika.
- ❑ Kako bi se ovaj cilj ipak postigao definisano je pomoćno rešenje - mehanizam slanja “**kolačića**” (*cookies*) između klijenta i servera
- ❑ Uveo ga je *Netscape Navigator*, a kasnije je postao standardan mehanizam za ovu namenu podržan od svih Web čitača i servera.
- ❑ **Šta je jedan cookie zapravo?**
- ❑ Možemo ga shvatiti kao string od tipično 20-30 nerazumljivih znakova koji je namenjen za jednoznačno identifikovanje korisnika na serveru.

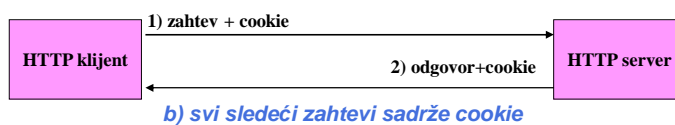
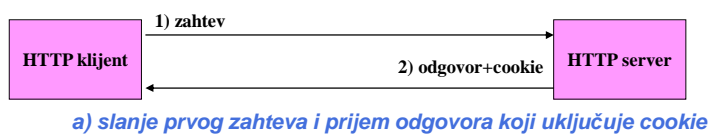
42

Praćenje sesije korisnika

- ❑ Prilikom slanja **prvog zahteva** server će ustanoviti da mu klijent nije poslao **cookie** kao jednu stavku u zaglavlju zahteva.
- ❑ U odgovor na taj zahtev server će dodati **cookie**.
- ❑ Ukoliko je Web čitač podešen tako da radi sa **cookie-ima**, on će u svim sledećim zahtevima koje bude slao tom serveru uključiti i **cookie**, tako da će server moći da prepozna klijenta koga je već ranije opsluživao.

43

Praćenje sesije korisnika



44

Svrha kolačića



- ❑ **Glavna svrha** kolačića jeste da sačuva informacije:
 - ❖ Oni se uglavnom koriste za čuvanje korisničkog imena i lozinke – korisnik ne mora da ih unosi svaki put kada pristupi lokaciji
 - ❖ Mogu se koristiti za čuvanje svih podešavanja koja je korisnik zadao kada je poslednji put pristupao lokaciji
- ❑ **Sa programerske tačke gledišta:**
 - ❖ Kolačići se koriste da reše probleme vezane za upravljanje stanjem – problem stanja je povezan sa činjenicom da se između dve posete lokaciji server ne seća korisnika

45

Upotreba kolačića



- ❑ Nekim korisnicima se ne dopada kada neko prati njihovo ponašanje na Webu, pa će oni isključiti kolačiće
- ❑ Da bi se to izbeglo i steklo poverenje korisnika, autori Web lokacije bi trebalo da sačine:
 - ❖ poseban **dokument o privatnosti** ili
 - ❖ **izjavu o upotrebi** i tako obaveste korisnika za šta će kolačići biti upotrebljeni
 - ❖ **samo jedan kolačić** koji će istovremeno i da prati korisnika i da čuva informacije o podešavanjima koja je on zadao – više kolačića zahteva od korisnika niz potvrda, jer svaki mora biti prihvaćen
 - ❖ **alternativno rešenje** za korisnike koji ne žele da prihvate kolačiće

46

Serversko programiranje



Serversko programiranje



- ❑ Serversko programiranje ima mnogo oblika, uključujući CGI skriptove, serverske API poput **ISAPI** (*Internet Server Application Programming Interface*) i Java programa, serverska okruženja za skriptove, kakva su **ASP, PHP, ColdFusion, JSP**
- ❑ Svaka tehnologija ima svoje prednosti i mane – zajedničko svojstvo: **kontrola**
- ❑ Server je jedini deo klijent-server arhitekture nad kojim autor lokacije ima kontrolu

Serversko programiranje



- ❑ Serversko programiranje ne primećuje razlike koje mogu da postoje na klijentskoj strani,
- ❑ Lokacija čija je interaktivnost bazirana na serveru, teoretski može da isporuči sadržaj bilo kom čitaču
- ❑ Razvoj interaktivnih elemenata, uprkos prednostima klijentske tehnologije, za veliki broj lokacija sa velikim protokom podataka oslanja se uglavnom na serverske tehnologije
- ❑ Glavna mana serverske tehnologije je: **brzina**
- ❑ Zbog kompletne interakcije koja se odvija na serveru može doći do kašnjenja zbog:
 - ❖ Opterećenosti servera ili
 - ❖ Zagušenosti mreže za prenos podataka

49

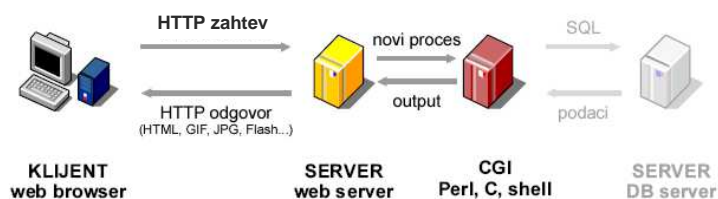
CGI programi



- ❑ Najstarija tehnologija za serversko programiranje: **Common Gateway Interface (CGI) programi**
- ❑ CGI predstavlja skup standarda koji definišu kako treba napisati dinamički dokument
- ❑ CGI omogućava Web serverima da pokreću programe kao odgovor na HTTP zahteve
- ❑ CGI specificira kako proslediti argumente programu koji će se izvršiti kao deo HTTP zahteva
- ❑ CGI omogućava da se vrati rezultat u HTML (ili nekom drugom) formatu, i to u zavisnosti od parametara konkretnog zahteva

50

CGI programi



Svaki HTTP zahtev, od strane korisnika, generiše novi proces na serveru

51

CGI programi

- ❑ CGI programi mogu da se pišu na svim jezicima koji mogu da se koriste na serveru koji izvršava program
- ❑ Najčešće se CGI programi pišu na jeziku **Perl**, zbog njegove izuzetne mogućnosti za rad sa znakovnim nizovima i podrške za Web
- ❑ CGI programi mogu se pisati u bilo kom od sledećih jezika:
 - ❖ **C/C++, Java, Visual Basic, pa čak i Pascal, Cobol ili Fortran**

CGI programe treba pisati na jeziku koji je brz, prenosiv i koji odgovara serveru što će ga izvršavati i programeru koji će ga pisati i održavati.

52

CGI programi



- ❑ CGI program mora da kreira posebni proces za svaki zahtev posetioca, koji nestaje tek kada se u potpunosti završi transfer podataka
- ❑ Kreiranje nove instance programa zahteva dodatno vreme, OS treba da uči program, alocira memorijski prostor, a nakon izvršenja da dealocira i ukloni program iz memorije
- ❑ Performanse – značajne – server opslužuje veliki broj istovremenih zahteva
- ❑ U serverskim programima veoma je važan bezbednosni aspekt

CGI programi se lakše mogu nadgledati i održavati ukoliko se postave u jedan direktorijum

53

Serverski API



- ❑ Skup pravila koje izvorni softver mora da poštuje da bi Internet isporučio podatke na odredišni softver
- ❑ Serverski API programi koriste se za važne funkcije koje treba brzo da se izvršavaju
- ❑ Omogućavaju dobru integraciju sa Web serverom
- ❑ Lako pristupaju glavnim serverskim funkcijama kao što su:
 - ❖ Provera identiteta, kontrola pristupa i brz pristup bazi podataka ili pozadinskim uslugama
- ❑ Glavna mana serverskih API programa jeste njihova **složenost i visoka cena**
- ❑ Loše napisan API mogao bi da blokira ceo Web server, dok loš CGI program može samo da zauzme previše procesorskog vremena

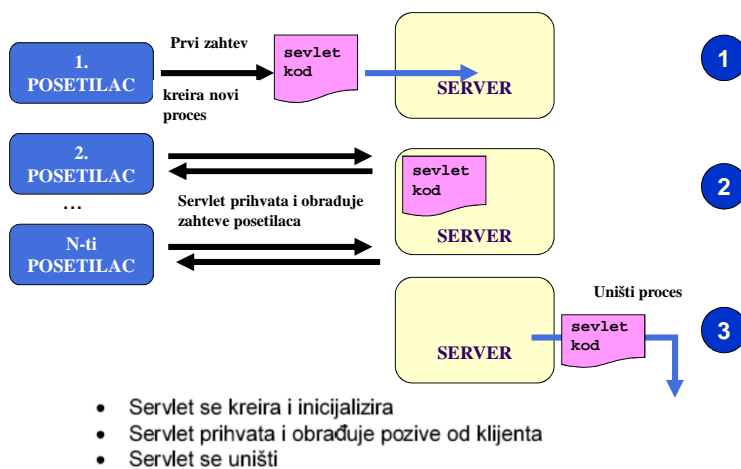
54

Java Servleti

- ❑ Servleti - jedna od tehnologija za generisanje dinamičkih Web sadržaja.
- ❑ Da bi se servleti mogli koristiti, Web server mora da ima odgovarajuću podršku za servlete. Pisanje servleta je moguće samo u programskom jeziku Java, tako da je za njihovo izvršavanje potrebna i JVM (koju najčešće obezbeđuje Web server).
- ❑ Servlet je - Java klasa koja nasleđuje standardnu klasu *HttpServlet*.
- ❑ Klase i interfejsi koji se koriste u pisanju servleta nalaze se u paketima
 - ❖ *javax.servlet* i
 - ❖ *javax.servlet.http*.

55

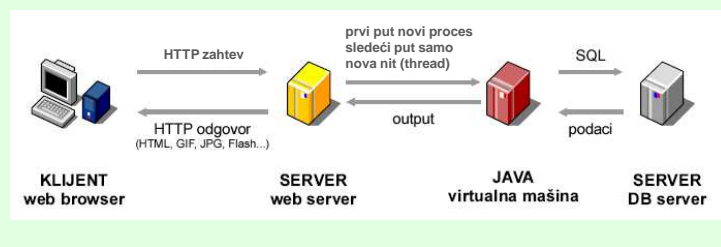
Životni ciklus servleta



56

Prednosti Java Servleta

- ❑ **Efikasnost** – prilikom prvog zahteva prema servletu, server mora pokrenuti novi proces unutar Java virtualne mašine, ali za svaki sledeći zahtev prema tom servletu, JVM kreira samo malu Java nit (eng. Thread)
- ❑ **Portabilnost** – zbog prirode napiši jednom pokreni, bilo gde (eng. Write once, run anywhere) Java idioma, servleti su portabilni na različitim SW/HW platformama



57

Nedostaci Java servleta

- ❑ Java servleti mogu biti **sporiji** od CGI skripta, u situacijama gde servlet poziva veliki broj korisnika
- ❑ Vidljiv nedostatak je i vezivanje za samo **jedan programski jezik** – Java (što je važno samo za programere i ne smatra se nedostakom tehnologije)
- ❑ HTML kao izlaz iz servleta je priličan problem, jer je reč o internom štampanju HTML koda iz servleta – **teško čitljive i složene izlazne naredbe**, koje sadrže HTML i programski kod ostatka aplikacije

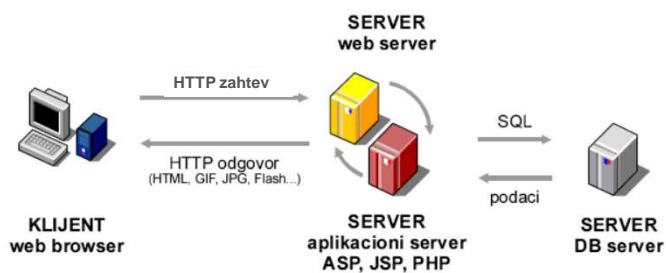
58

Serverski skriptovi

- ❑ Često nazivan “**serverski HTML**” ili “**serverski skriptovi**” – pridruživanje programiranja Web stranama na relativno jednostavan i brz način
- ❑ Cilj serverskih skriptova: izrada posebnih HTML datoteka ili šablona koji sadrže smešu skripta i HTML elemenata, koje će server po potrebi učitati
- ❑ Postoje mnoga okruženja za serverske skriptide:
 - ❖ **PHP –Personal Home Page**
 - ❖ **ASP - Active Server Pages**
 - ❖ **ASP.NET**
 - ❖ **JSP - JavaServer Pages**
 - ❖ **ColdFusion..**
- ❑ Razlike su veoma male – glavna razlika između okruženja jeste sintaksa

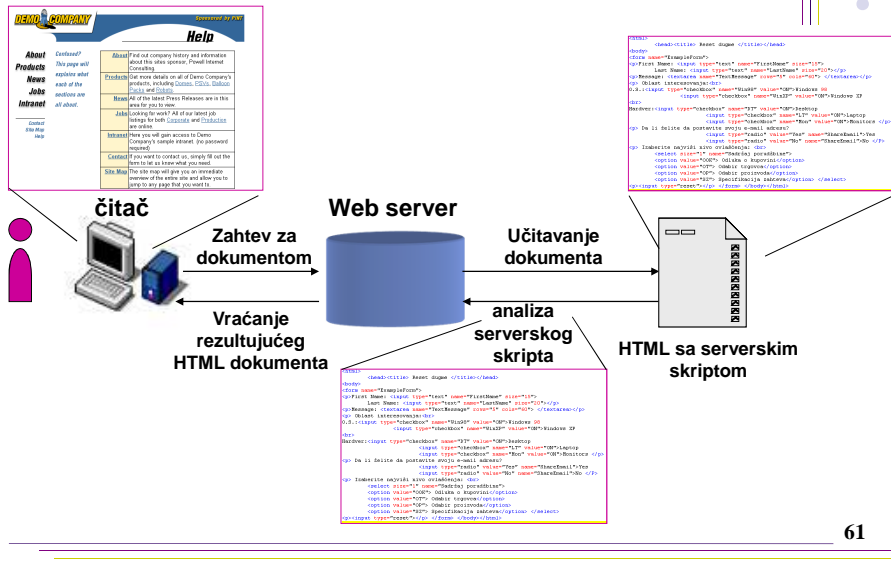
59

Primer serverske analize HTML dokumenta



60

Primer serverske analize dokumenta



ASP

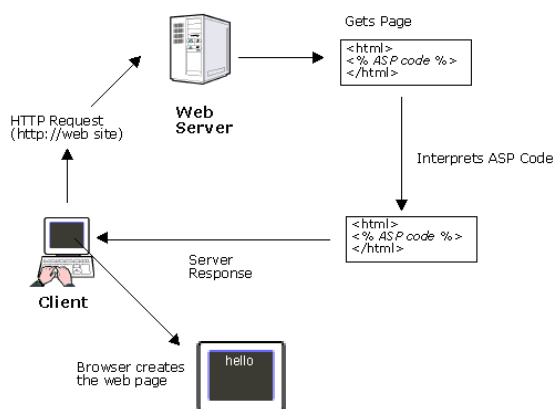
- ❑ ASP ("Stranice aktivne na serveru", eng. Active Server Pages) predstavlja jedan od skriptnih jezika za razvijanje Internet sajtova, razvijen od strane kompanije Microsoft.
- ❑ Njegov glavni takmac na Internet tržištu je PHP, koji ima tu osobinu da je besplatan i slobodan, dok su ASP i svi proizvodi potrebni za njegovo korišćenje komercijalni i u vlasništvu kompanije Microsoft.
- ❑ Sa druge strane, Microsoft konstantno održava i dodaje nove mogućnosti svom paketu, na taj način ga čineći modernim, čvrstim i pružajući svojim klijentima svu moguću tehničku podršku.

ASP

- ❑ Nakon uvođenja .NET tehnologije, Microsoft je i ASP prilagodio istom, tako da je u upotrebi verzija ASP.NET 2.0, objavljena 2005. godine.
- ❑ ASP je **skriptni jezik**, kao i PHP, i potreban je interpreter na serverskoj strani da bi isti mogao da radi.
- ❑ ASP nije programski jezik, već tehnologija koja omogućava programiranje Web servera, dok programski jezik zavisi od izbora autora:
 - ❖ **VBScript** – najčešće korišćen
 - ❖ **JScript** – Microsoftova verzija JavaScript jezika
 - ❖ **C#** (samo za ASP.NET)
 - ❖ **Više od 19 jezika** (samo ASP.NET) razvijenih od nezavisnih firmi

63

ASP



64

Kvantitativno poređenje serverskih tehnologija

	CGI - Perl	PHP	ASP	ASP.NET	JSP	ColdFusion
Jednostavnost razvoja	2 vrlo složena sintaksa	3 relativno jednostavan jezik	4 VBScript i JScript	4 VBScript i JScript, odličan razvojni alat	3 neophodno poznavanje programskog jezika Java	5 tag orijentisan
Skalabilnost	1 loša	3 Prihvatljiva	4 Dobra, ali zavisi od OS-a	4 Dobra, ali zavisi od OS-a	5 Dobra - JVM	5 Izvršna, ali skupa
Osobine programskog jezika	4 ogromna biblioteka dobro testiranih funkcija	4 Brojne korisne standardne funkcije (PHP4)	4 Prilično jasan (VBScript i JScript)	5 brojni programski jezici (VBScript, JScript, C# i dr.)	5 Mogućnost upotrebe snage programskog jezika Java	5 Velika biblioteka postojećih tagova, plus mogućnost definisanja novih (CFML)
Portabilnost	4 razne verzije UNIX-a i Windowsa	5 Linux, Windows, Mac OS, OS/2	3 Windows 9x, ME, NT, 2000, XP, .NET	4 Windows 9x, ME, NT, 2000, XP, .NET	5 UNIX, Microsoft Windows, Mac OS, Linux	4 Windows NT, Solaris, Linux
Podrška za komponente	1 -	4 COM, Java klase	3 COM komponente	3 COM komponente	4 JavaBeans, Enterprise JavaBeans	5 COM, CORBA, JavaBeans
Cena	5 Besplatan	5 Besplatan	4 Besplatan sa Windowsom NT, 2000, XP	4 Besplatan sa Windowsom NT, 2000, XP	5 Besplatan	2 \$799 (Pro) \$4999 (Enterprise)

65

Kvantitativno poređenje serverskih tehnologija

	CGI - Perl	PHP	ASP	ASP.NET	JSP	ColdFusion
Podrška za baze podataka	5 MySQL, ODBC, Interbase, Oracle, PostgreSQL i sl.	5 MySQL, mSQL, ODBC, Oracle, Informix, Sybase, i sl.	5 bilo koja ODBC kompatibilna baza podataka	5 bilo koja ODBC kompatibilna baza podataka	5 bilo koja ODBC i JDBC kompatibilna baza podataka	5 ODBC, OLE DB, DB2, Oracle, Informix, Sybase, i sl.
Održavanje kôda	1 vrlo složeno	3 dovoljno čitljiv kôd	3 jednostavna sintaksa	4 uvođenje event modela	3 može dovesti do vrlo komplikovanog kôda	5 tag orijentisano
Podrška za web servere	5 praktično svi serveri imaju podršku za CGI Perl	5 Apache iPlanet/Netscape Enterprise Server, MS IIS, Zeus, httpd, i sl.	4 IIS, Personal Web Server (PWS), ostali serveri sa paketima drugih proizvođača	4 IIS sa .NET frameworkom, ostali serveri sa paketima drugih proizvođača	5 Bilo koji web server, uključujući Apache, Netscape i IIS	5 IIS, Netscape Enterprise Server, Apache, Website Server (WSAPI), CGI
UKUPNO	28	37	34	37	40	41

66

Tehnologije distribuiranih objekata



Osnovni koncepti



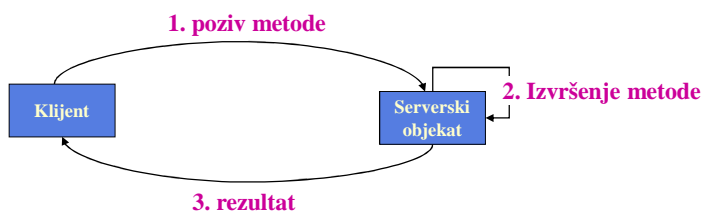
- ❑ **Koncept distribuiranih objekata** - neki objekat (instanca klase) sa svojim metodama i atributima može da “postoji” na nekom računaru i da drugi programi, odnosno objekti, koji se izvršavaju na drugim računarima mogu da mu pristupaju.
- ❑ Pristup tom objektu – mogućnost pozivanja metoda i pristupa njegovim atributima.
- ❑ Pristup tom “serverskom” objektu bi, sa strane ostalih učesnika (“klijenata”), trebalo da bude što jednostavniji.
- ❑ U idealnom slučaju pristup njemu ne bi trebalo da se razlikuje od pristupa lokalnim objektima – ostalim objektima koji se nalaze u sastavu programa.

Osnovni koncepti

- ❑ Poziv metode serverskog objekta podrazumeva izvršavanje te metode na onom računaru na kome se nalazi sam objekat.
- ❑ To znači da se klijentski program izvršava na više računara – inicijalno na onom na kome je pokrenut, ali i na svim računarima na kojima se nalaze serverski objekti koje on koristi.

69

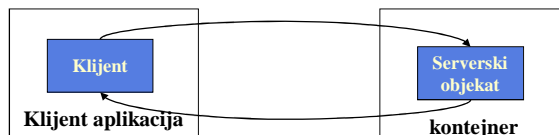
Poziv metode serverskog objekta



- ❖ **Komunikacija dva objekta:**
 - ❖ klijentski objekat je deo klijentske aplikacije, a serverski objekat je deo serverske aplikacije.
 - ❖ Serverska aplikacija se često naziva “kontejner” za objekte jer je njena osnovna funkcija da obezbedi mrežne i druge servise koji su neophodni za ovakav serverski objekat.

70

Klijentski i serverski objekat u okviru svojih programa



- Posmatrano sa stanovišta autora klijentske aplikacije, i klijentski i serverski objekat su sastavni deo **jednog programa**.
- Osobina tog programa je da se izvršava na više računara u mreži.
 - ❖ Serverski program, logički posmatrano, i ne postoji; postoji samo serverski objekat koji se ponaša/koristi kao da je deo klijentskog (jedinog) programa.

71

Tehnologije distribuiranih objekata

- ❖ **COM/DCOM** - *(Distributed) Component Object Model*
- ❖ **RMI** - *Remote Method Invocation*
- ❖ **CORBA** - *Common Object Request Broker Architecture*
- ❖ **EJB** - *Enterprise JavaBeans*

72