

# Web programozás

2010–2011

# Áttekintés

- ▶ A web rövid története
- ▶ Kliens–szerver architektúra
- ▶ Néhány alapfogalom
- ▶ Kliens- illetve szerver oldali technológiák áttekintése
- ▶ Áttekintés: miről lesz szó (kurzus/labor/vizsga)

# A web rövid története

## WWW (World Wide Web - Világméretű Hálózat)

- ▶ 1989 – Tim Berners-Lee (CERN – Európai Részecskegyorsító Intézet) javaslata:
  - ▶ “hiperszöveg rendszer” alkalmazása kutatók közti kommunikáció javítására
  - ▶ szöveges információ, képek, (mozgóképek, hanganyag) hyperlinkek segítségével összekapcsolt rendszere
  - ▶ ötlet:
    - a Gopher protokoll-tól kölcsönzött
    - hiperszöveg (hypertext), hiperhivatkozás (hyperlink) elve ismert már a 1970-es években
- ▶ 1992 január, Genf – a WEB első nyilvános használata

# A web rövid története

## A WWW komponensei:

1. HTML-nyelven íródott *Web-oldalak* – információ, hyperlinkek
2. *Web-szerverek* – Web-oldalak rendelkezésre bocsátása (HTTP protokoll)
3. *Web-böngészők* (Web-browser, ügyfél vagy kliens program) – információ lekérése (URL segítségével), formázott megjelenítése

## A WWW elterjedése:

- ▶ NCSA Mosaic – az Illinois-i Egyetemen működő NCSA (National Center for Supercomputing Applications) által kibocsátott első *ingyenes* böngésző (Marc Andreessen és csapata)
- ▶ egy éven belül már kb. 2 millióan használták
- ▶ nyílt forráskódú

# A web rövid története

## A WWW elterjedése:

- ▶ 1993 közepén – kb. 130 web-oldal
- ▶ 1993 vége – közel 600
- ▶ 1994 – majdnem 3000
- ▶ 1996 elején – több mint 90 000
- ▶ jelenleg ... ?

## Újabb böngészők megjelenése:

- ▶ Netscape Navigator (Marc Andreessen, Jim Clark)
- ▶ Microsoft Internet Explorer (1995, '99-től ez a legnépszerűbb)
- ▶ Lynx (1993, Kansas) – szöveges terminálokra
- ▶ Mozilla (2002, nyílt forráskódú), Firefox (2004), Opera (1994, Norvégia), Safari (2003, Mac OS), Google Chrome (2008) stb.

# A web rövid története

## Igény a dinamikus tartalomra

- ▶ kezdetben: statikus HTML oldalak
  - ▶ interaktív oldalak:
    - ▶ adatok bevitele
    - ▶ adatok feldolgozása
    - ▶ aktuális adatok megjelenítése (pl. adatbázisból)
- ⇒ dinamikusan létrehozott HTML oldalak
- ▶ megjelenítési lehetőségek szélesedő skálája
  - ▶ multimédiás tartalom

## fejlődés iránya

- ▶ a Web-es felhasználói interfész közelítése a desktop-alkalmazásokéhoz

# Kliens-szerver architektúra

A *Web kliens-szerver architektúrát* alkalmaz az információk megosztására és terjesztésére.

## A rendszer működése:

- ▶ kliens program (web-böngésző)  $\xrightarrow{\text{kérés}}$  Internetre rákapcsolt web-szerver
- ▶ web-szerver: értelmezi az üzenetet  $\xrightarrow{\text{válasz}}$  kliens  
válasz = a kért dokumentum + kiegészítő információk
- ▶ böngésző: értelmezi a kapott állományt, majd megjeleníti/lementi/futtatja azt a felhasználói gépen
  
- ▶ az információ-csere a HTTP protokoll segítségével történik

## Statikus HTML oldal lekérése

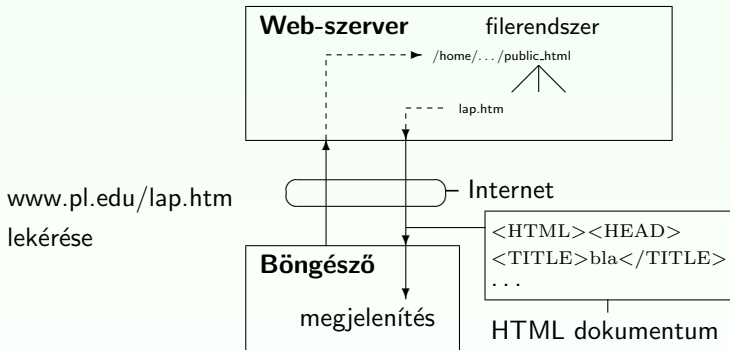


Figure: Egyszerű HTTP kérés



# Dinamikusan létrehozott HTML oldal

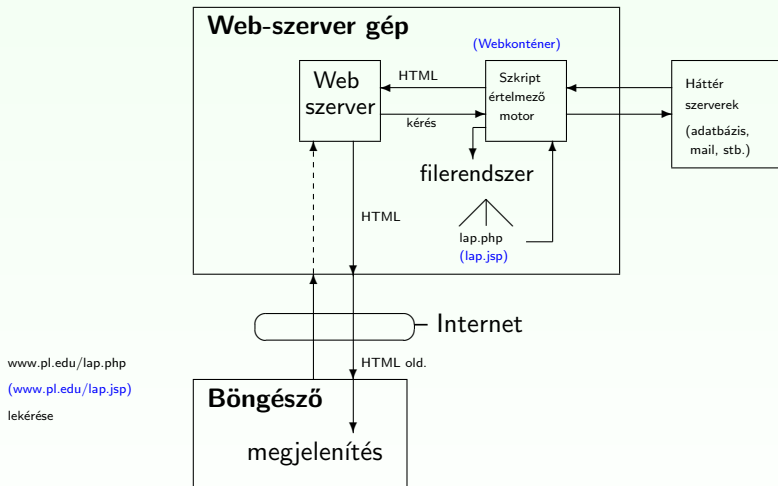


Figure: HTTP kérés web-alkalmazás esetén

# URL

Minden erőforrás a WEB-en egy címmel rendelkezik, amelynek formátumát a Uniform Resource Locator (URL) határozza meg

Egy URL a következő részekből áll:

[protokoll://] [gazdagép ] [:port ] [kérés útvonala ]

- ▶ **protokoll**: meghatározza a kommunikáció módját (pl. http, https, ftp, mailto, stb.)
- ▶ **gazdagép**: azonosítja a számítógépet, amelynek a kérést címezzük [gépnév ] . [altartomány(ok) ] . [tartomány ]
- ▶ **port**: http-nél alapértelmezett a 80-as
- ▶ **kérés útvonala**: a kért erőforrás neve, elérési útként megadva (esetleg paraméterekkel)

<http://www.cs.ubbcluj.ro/~laura/webprg/index.htm>

# HTTP protokoll

## HyperText Transfer Protocol (Hipertext Átviteli Protokoll)

Böngésző  $\overset{\text{HTTP}}{\rightleftharpoons}$  Web-szerver

### tulajdonságai:

- ▶ állapot nélküli
- ▶ általános
- ▶ egyszerű, gyors

### HTTP üzenetváltás lépései:

- ▶ kliens és szerver közti kapcsolat felépítése (80-as port)
- ▶ a kliens egy kérést (**request**) küld a szervernek
- ▶ a szerver a kérés fogadása és a kért feladatok elvégzése után egy választ (**response**) küld vissza a kliensnek
- ▶ mindkét fél bontja a kapcsolatot (HTTP 1.0) vagy a kapcsolat újrahasználható (HTTP 1.1)

# HTTP protokoll

## A kérés (request) tartalma:

- ▶ információ a böngészőről (pl. ennek típusa és verziója)
- ▶ az alkalmazott formátum (pl. az alkalmazott HTTP protokoll verzió)
- ▶ **paraméterek** (pl. form-ban megadott vagy a hivatkozásban szereplő)  
`http://cs.ubbcluj.ro/jegyek.jsp?diakID=100&felev=1`
- ▶ metódus - megadja az elvégzendő művelet típusát

## metódusok:

- ▶ **GET** – a paraméterek az URL-ben lesznek elküldve (inkább lekérdezésre ajánlott használni)
- ▶ **POST** – a paramétereket az üzenet törzsében küldi a böngésző a szervernek (ajánlat: használható adatmódosításra)
- ▶ más, kevésbé használt: HEAD, PUT, DELETE, OPTIONS, TRACE, CONNECT, PATCH

# HTTP protokoll

## Visszajelzés:

- ▶ a visszaküldött objektum jellemzői:
  - ▶ kódolási formátum (sima szöveg, UUENCODE, zip, stb.)
  - ▶ típus (szöveges információ, kép, video, hang, bináris állomány, stb.)
  - ▶ alkalmazott nyelv
- ▶ “HTTP/1.1 **200 OK**” - siker esetén
- ▶ **a kért objektum** (pl. HTML oldal)
- ▶ hibaüzenet (ha szükséges)

## Néhány gyakoribb hibakód:

- ▶ 400 Bad Request  
The request contains bad syntax or cannot be fulfilled.
- ▶ 404 Not Found
- ▶ 414 Request-URI Too Long
- ▶ 500 Internal Server Error

# Kliens oldali technológiák

- ▶ a statikus HTML kliens oldali kiegészítései
- ▶ böngészőbe beépített (pl. JavaScript) vagy hozzáadott segédprogram használata szükséges (pl. JVM-plugin, Java Applet futtatásához)

## Konkrét kliens oldali technológiák:

- ▶ Egymásba ágyazható stíluslapok (Cascading Style Sheets)
- ▶ Kliens oldali script nyelvek: pl. JavaScript, JScript, VBScript
- ▶ Java Appletek
- ▶ Rich Internet Application (RIA) platformok: Adobe Flash, JavaFX, Microsoft Silverlight

AJAX - kliens és szerver közti aszinkron módon történő kommunikáció

# Kliens oldali technológiák

## CSS

- ▶ HTML tag-ek megjelenítési stílusát adhatjuk meg a segítségével
- ▶ *egységes megjelenítést* kölcsönöz a web-oldalnak
- ▶ a HTML oldal kinézete *könnyen módosítható* a CSS állomány módosításával

## Kliens oldali script nyelvek – JavaScript

- ▶ a HTML oldalak készítői számára egy programozási eszközt biztosít
- ▶ módosíthatja a HTML tartalmát, kinézetét (a HTML DOM objektumaihoz való hozzáférés által)
- ▶ eseményekre tud reagálni
- ▶ a bevitt adat helyességének ellenőrzésére ad lehetőséget

# Szerver oldali technológiák

## Általános jellemzők, feladatok:

- ▶ Weboldal dinamikus létrehozása
- ▶ a felhasználótól jövő lekérdezésekre adott válasz, a felhasználó által bevitt adatok (HTML form) feldolgozása
- ▶ adatok lekérdezése (adatbázisból, XML fájlból vagy bármilyen más adatforrásból) illetve az adatok megfelelő formában való elküldése
- ▶ különböző Weboldalakhoz való hozzáférés ellenőrzése (pl. bizonyos oldalakat csak bejelentkezett felhasználó érhet el)
- ▶ a válasz testreszabása a böngésző típusának függvényében
- ▶ a hálózati forgalom csökkentése (pl. az elküldött tartalom be-/kitömörítése)



# Szerver oldali technológiák

## Konkrét szerver oldali technológiák:

- ▶ CGI
- ▶ Szerver-oldali script alapú technológiák (pl. PHP, ASP)
- ▶ Java alapú technológiák (servlet, JSP)

## Szerver oldalon gyakran használt egyéb technológiák

- ▶ SQL – adatbázis lekérdező nyelv, séma módosítás
- ▶ XML – Extensible Markup Language

### webalkalmazás:

alapgondolata: bizonyos webcímek mögött nem statikus tartalom van (pl. HTML), hanem a szerver a böngésző kérésére dinamikus tartalmat hoz létre, és küld el a böngészőnek.

## Tervezett kurzus-témák:

- ▶ HTML, (XHTML), HTML DOM
- ▶ CSS
- ▶ Kliens oldali technológiák - Script nyelvek - JavaScript
- ▶ AJAX - Asynchronous JavaScript and XML
- ▶ Szerver oldali technológiák - PHP
- ▶ JAVA-alapú webalkalmazások:
  - ▶ Servletek
  - ▶ JSP
  - ▶ JSP - Tag-ek
  - ▶ MVC elv
  - ▶ JDBC adatbázishozzáférés
  - ▶ (XML feldolgozás JAVA-ban)
  - ▶ Struts keretrendszer
- ▶ Rich Internet Applications (RIA) – ismertető

## Labortémák:

- ▶ HTML + CSS
- ▶ HTML Form + JavaScript
- ▶ PHP
- ▶ Servlet
- ▶ JSP
- ▶ JSP Tag-ek
- ▶ JDBC
- ▶ AJAX
- ▶ Projekt

## A tantárgy honlapja

[www.cs.ubbcluj.ro/~laura/webprg](http://www.cs.ubbcluj.ro/~laura/webprg)

## Vizsga:

- ▶ elmélet
- ▶ gyakorlat – projektbemutató

Kérdések...?!?