

WEBDESIGN

1. ELŐADÁS

- A weblapkészítés ergonómiája
- Webes technológiák

TARTALOM

Bevezető

Webergonómia

Használhatóság – usability

Funkció és design

BEVEZETŐ

- A tervezési folyamat központjában a ~~design~~ van.

a felhasználók igényei



egy tervezési filozófia, ahol a felhasználók igényei, kívánságai, viselkedése, szokásai vannak a tervezési folyamat központjában.



WEBERGONÓMIA

- Kinek fejlesztünk weblapot?

ALAPVETŐ KÉRDÉSEK

Kik a felhasználók?

Milyen főbb funkciókra lesz szükségük?

Miért jó nekik a készülendő weboldal?

Vajon milyen más weboldalakat használnak?

Mi legyen a legfontosabb interakció?

Milyen folyamatokon keresztül érhetők el ezek az interakciók?

ERGONÓMIA

Ergonómia: „a dolgozó és a munkakörnyezet közötti kapcsolat tanulmányozása”. Ez a szó a görög “ergos” szóból származik, ami munkát jelent, és a “nomos” szóból, ennek jelentése törvény.



Forrás:

<http://erg.bme.hu/Emania/1999/Bara0001/ergo.html>, 2012.

Az Idegen Szavak Gyűjteménye: <http://idegen-szavak.hu/ergon%C3%B3mia>, 2012.

WEBERGONÓMIA

A webergonómia a felhasználói kényelem speciálisan az internetes világra értelmezett formája, amely az internetes oldalak használhatóságával és ennek kapcsán az internet felhasználók szokásainak vizsgálatával foglalkozik.

A célja gyakorlatilag a látogatók elégedettségének és ezzel együtt a weboldal hatékonyságának a növelése, a kezelhetőség egyszerűsítése mellett.

HASZNÁLHATÓSÁG – USABILITY

A használhatóság (usability) fogalmát a szakirodalomban, illetve a különböző szabványokban eltérően definiálják.

ISO 9241-11 (Guidance on Usability, 1998) definíció:

„Annak mértéke, ahogy a terméket meghatározott felhasználók meghatározott célokért eredményesen, hatékonyan és elégedetten használják egy adott környezetben.”

„Egy weboldal használhatósága annál jobb, minél kevesebb erőfeszítést kell tennie a felhasználónak a kezelése során.”

(Steve Krug)

HASZNÁLHATÓSÁGI FAKTOROK

U

S

A

B

I

L

I

T

Y

- **Megtanulhatóság** (Learnability): Amikor a felhasználó először találkozik a felülettel, mennyire egyszerű számára az alapvető feladatok elvégzése?
- **Hatékonyság** (Efficiency): Ha a felhasználó már megismerkedett a felülettel, akkor milyen gyorsan tud különböző feladatokat megoldani?
- **Megjegyezhetőség** (Memorability): Amennyiben a felhasználó egy ideig nem használta a felületet, a korábban megszerzett tudását mennyire gyorsan tudja újra felidézni?
- **Hibák** (Errors): Hány hibát ejt a felhasználó, milyen mértékűek ezek, és mennyire tudják a hibáikat könnyen javítani?
- **Elégedettség** (Satisfaction): Mennyire megfelelő, kényelmes a felület használata a felhasználó számára?

FELHASZNÁLÓI SZOKÁSOK

Ahhoz, hogy érthetőbbé váljon számunkra, mivel tudjuk a felhasználók számára használhatóbbá (magától érthetővé) tenni weboldalainkat ismernünk kell a felhasználók viselkedését.

1. A felhasználók nem olvassák végig az oldalakat, csak átfutják.
2. Nem a lehető legjobb döntésre törekszenek, hanem megragadják az első elfogadható lehetőséget.
3. Nem gondolják végig, hogy hogyan működik valami, csak használják azt valahogyan.

FELHASZNÁLÓI ÉLMÉNY – USER EXPERIENCE

Olyan formában adjuk az információt, amely a felhasználói szokásoknak megfelel. Cél, hogy a felhasználó jól és biztonságban érezze magát a felületen.

- Jelezni kell, ha a felhasználónak várnia kell
- Ne kelljen a felhasználónak emlékeznie korábbi oldalakon lévő információra
- Jól használható Gyakran ismételt kérdések gyűjtemény biztosítása

FUNKCIÓ ÉS DESIGN

- A design egy weblapnak csupán a kiegészítője, nem célja.
Az adott weblap céljának alárendelve, annak tartalmával összhangban kell működnie.
- A látogatók figyelme 78%-ban a szövegre, elsődlegesen a címekre, összefoglalókra és csak 22%-ban a képekre irányul.
- A designnak összhangban kell állnia a weblap tartalmával. Fontos, hogy visszaköszönjenek a termékkel, a szolgáltatással, az iparággal összekapcsolható vizuális elemek.
- A design szerepe az első pillanatokban a legfontosabb: az a feladata, hogy erősítse meg a felhasználót abban, hogy jó helyen jár.

**"Design is not
just what it
looks like &
feels like.
Design is how
it works."**

– Steve Jobs

MIRŐL LESZ MÉG SZÓ?

Web-technológiák: HTML, CSS, Javascript ...

Később:

Reszponzivitás

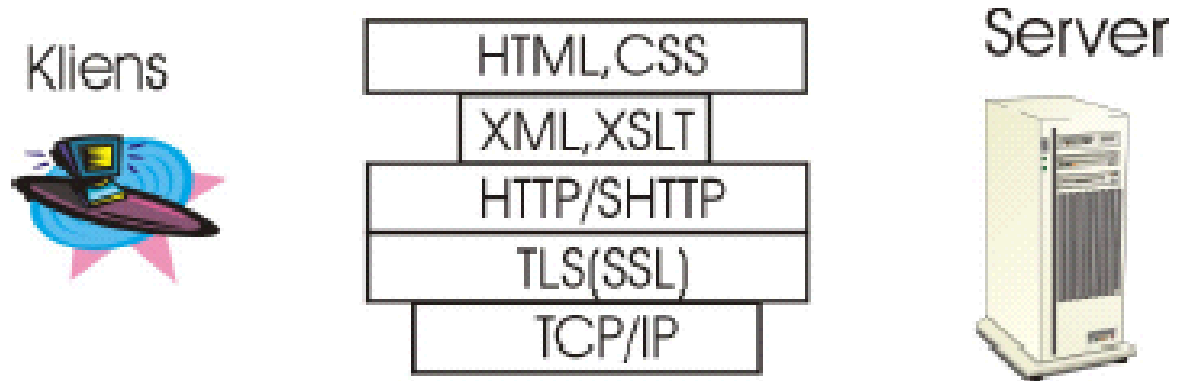
Webes űrlapok

TARTALOM

- Webes rendszer felépítése, működése
- Statikus vs. Dinamikus veboldal
- Kliensoldali programozás/technológiák
- Szerveroldali programozás/technológiák

WEBES RENDSZER FELÉPÍTÉSE, MŰKÖDÉSE

Adatforgalom:



WEBES RENDSZER MŰKÖDÉSE

A viszony kliens-szerver alapú:

- A kliens kér egy weboldalt a szervertől, mely szerencsés esetben visszaküldi azt.

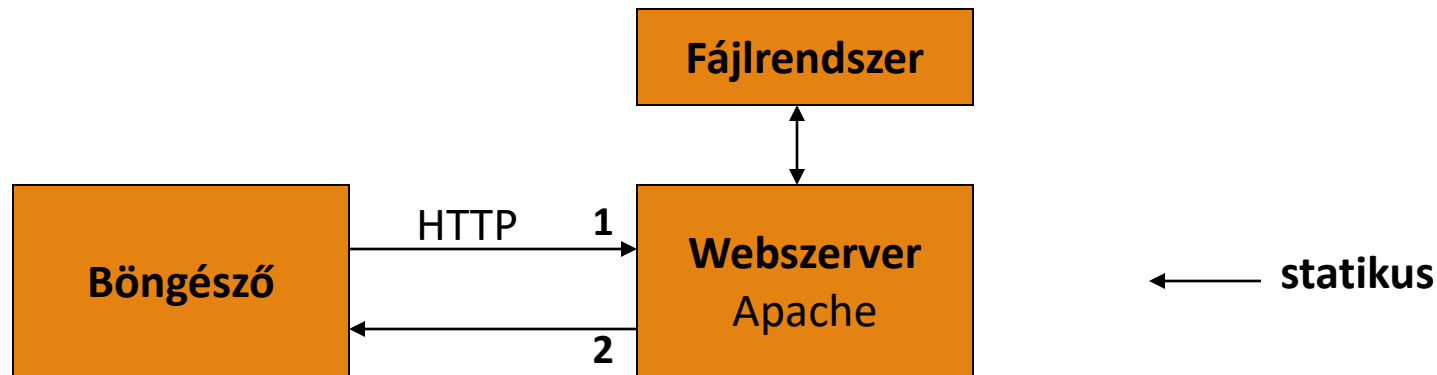
Rétegek

- Átviteli protokollként a jelenleg leginkább elterjedt TCP/IP protokollt ábrázoltuk, azonban bármely transzparens átvitelt biztosító protokoll megfelel erre a célra.
- A második réteg (TLS) biztosítja az átvitel biztonságát, megbízhatóságát, titkosságát (amennyiben az alkalmazás igényli ezt. Egy hír portálnál nincs különösebb igény az adatok biztonsága, titkossága iránt, egy online banki tranzakciónál viszont nélkülözhetetlen.).
- A harmadik réteggként szerepelő HTTP minden kapcsolatnál jelen van, gondoskodik a kliens, a szerver valamint az esetleges proxy szerverek közötti kommunikációról.

WEBES RENDSZER MŰKÖDÉSE

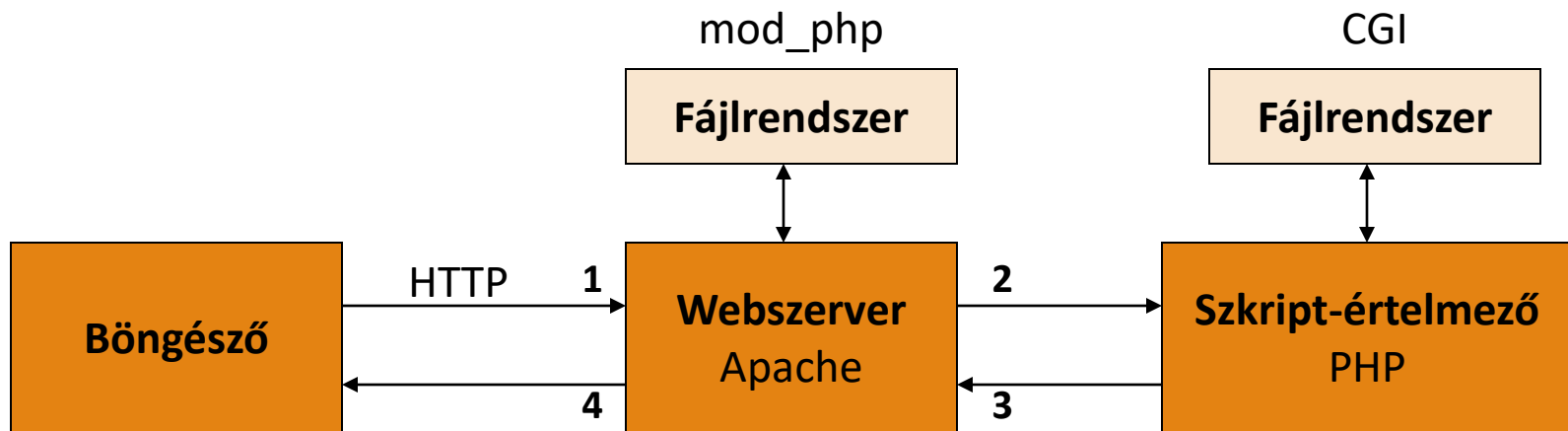
- A kommunikáció a célja a kliens által kért HTML, vagy XML dokumentumok átvitele.
- Az XML réteg (4) feladata az adatok általános leírása, az adatok megjelenítéséről nem tartalmaz információt. Különálló alkalmazások közötti adatcserére, illetve olyan esetekben szokták alkalmazni amikor a felhasználó több formában szeretné vizsgálni az adatokat (pl.: mobil készülék, pdf, html, ... stb.)
- A HTML (5) réteg szerepe az adatok megfelelő megjelenítése. A CSS nyelv segítségével megjelenítési stílusokat tudunk definiálni.

MITŐL DINAMIKUS EGY WEBOLDAL?



- A böngésző elküldi a kérést
- A webszerver ellenőrzi a jogosultságokat, beolvassa a fájlt, majd visszaküldi a böngészőnek
- A böngésző fogadja a fájlt, és megjeleníti a weboldalt

MITŐL DINAMIKUS EGY WEBOLDAL?



- A böngésző elküldi a kérést
- A webszerver ellenőrzi a jogosultságokat, **beolvassa a fájlt és továbbítja az értelmezőnek VAGY megmondja az értelmezőnek, hogy mit kell beolvasnia**
- **A értelmező feldolgozza a fájlt, és visszaadja a webszervernek**
- **A webszerver kiküldi a generált fájlt**
- A böngésző fogadja a fájlt, és megjeleníti a weboldalt

STATIKUS VS. DINAMIKUS WEBOLDAL

- Sok olyan feladat létezik, melyeket nem lehet, vagy nem célszerű statikus oldallal megoldani.
- A cégek adataikat többnyire adatbázisokban tartják. Ezekből az adatbázisokból le lehet ugyan képezni az adatokat különböző eljárásokkal statikus oldalakra, azonban ez több olyan problémát is felvet, mely az információ közzétételének ezen módját kizárja a lehetséges megoldások közül.
- Vannak olyan egyszerű feladatok is, melyek ugyancsak szükségessé teszik a szerver oldali programfuttatást. Ilyen lehet például egy egyszerű letöltés-számláló az adott oldalon, de lehet például egy internetes portál friss hírek rovata, mely igen gyakran frissül. Ezért nem érdemes statikus weboldalként kezelni.

KLIENSOLDALI PROGRAMOK

- A dinamikus weboldal egyik változata, mely kliensoldali programok (client-side scripts) segítségével kommunikál a felhasználóval.
- Ilyen feladat lehet például egy, a felhasználó által kitöltött kérdőív elemeinek ellenőrzése a szerver bevonása nélkül, de ide tartozhat a webes számológép is.
- kliensoldali programozási technológiák:
 - JavaScript
 - VBScript.

SZERVER OLDALI PROGRAMOK

Szerver oldalon a tartalom frissítését korábban a CGI (Common Gateway Interface) technológiával oldották meg, ma erre sokkal fejlettebb eszközök vannak.

- Pl. PHP, ASP, JSP

Ma a vállalatok olyan komplex weben alapuló rendszereket használnak, melyek igen erőteljesen használják a

- Java nyelvet és a nyelvet támogató servlet technológiát.
- .NET technológiát

SZERVER OLDALI PROGRAMOZÁS- PHP

- A PHP (PHP: Hypertext Preprocessor) egy széles körben használt, nyílt forráskódú, általános célú programozási nyelv, különösen jó web fejlesztés támogatással és HTML-be ágyazási képességekkel. A szintakszisa a C, Java és Perl nyelvekre épül, könnyen megtanulható.
- A nyelv fő célja lehetőséget teremteni dinamikus generált weboldalak gyors készítésére.
- A PHP főleg szerver-oldali szkriptek írására készült, azaz bármire képes, amit más CGI programok el tudnak végezni, ilyen funkciók az űrlap adatok feldolgozása, dinamikus tartalom generálása, vagy süti küldése és fogadása.

AJÁNLOTT IRODALOM

- Jennifer Niederst Robbins: **Learning Web Design**, Fourth Edition, O'Reilly, 2012
- Jason Beaird , James George: **The Principles of Beautiful Web Design**, 2014
- RUNG, András-Kiss, Orhidea Edith: *Felhasználóbarát honlapok – Webergonómiai tanulmányok.*, Gondolat Kiadó, 2006
- Timothy Samara: **A grafikai tervezés kézikönyve - Elemek, összefüggések és szabályok**, Sclolar, 2015

AJÁNLOTT LINKEK

Első labortémához:

- Egy példa: [Archivio Mario Russo](#)
- Top webdesign trendek?
- <https://designstreet.hu/blog/melyik-a-jobb-a-cms-vagy-a-sajat-fejlesztesu-rendszer>
- Kódszerkesztők:
 - [Visual Studio Code](#)
 - [Sublime Text](#)
 - [Brackets](#)
 - [Notepad++](#)
- <https://www.w3schools.com/>