

1. feladat

Eperjesi Attila Dávid
Programtervező informatikus BSc

2021.01.05.

Tartalomjegyzék

1. Az alapok	3
1.1. A web és a látogató viszonya	3
1.1.1. Webes tipográfiai alapismeretek	3
1.1.2. Hogyan olvasunk a weben?	3
1.1.3. Kereső(re) optimalizálás	4
1.2. A web működése	4
1.2.1. Webszerver	5
2. A tartalom és a kinézet	6
2.1. HTML alapok	6
2.1.1. Mi az a HTML?	6
2.1.2. Hogyan kezdjük neki?	7
2.1.3. A head tagba írható elemek	7
2.1.4. HTML szerkesztők	8
2.1.5. Hogyan nézhetjük meg egy oldal HTML kódját?	8

1. fejezet

Az alapok

1.1. A web és a látogató viszonya

Webfejlesztőként magunk is látogatók vagyunk. Nap mint nap különböző weboldalt látogatunk meg. Ahhoz azonban, hogy jó weboldalt tudjunk készíteni, olyan módon kell látnunk a weboldalt, ahogy azt korábban nem tettük. Folyamatosan szem előtt kell tartanunk mérnöki szempontokat is.

1.1.1. Webes tipográfiai alapismeretek

Sokunkkal próbálták jól-rosszul megtanítani a szövegszerkesztési alapismereteket. Azonban a papíralapú szövegszerkesztéssel kapcsolatos tanulmányaink hátrányunkra válhatnak, ha nem értjük meg a papír és a weboldal mint különböző médiák közötti különbségeket.

A nyomdászoknak sokféle lehetőség áll a rendelkezésükre, amikor szóba kerül a tipográfia, mint például a betűkészletek pusztaszáma vagy az elrendezési lehetőségek széles skálája. A webes tipográfia ennél sokkal korlátozottabb, mivel olyan típusokkal és elrendezéssel kell dolgoznunk, amelyről tudjuk, hogy elérhető és használható lesz azokon a gépeken is, amelyeken az olvasók megnyitják a lapot, hiszen senki nem fejleszt saját magának weboldalt.¹

1.1.2. Hogyan olvasunk a weben?

Ha weboldal készítésére adjuk fejünket, akkor jó, ha tisztában vagyunk a látogatói szokásokkal. E témát egyre többen és egyre átfogóbban kutatják. Itt most csak egy rövid ajánló erejéig foglalkozunk a témával. Legalább a következő cikkek elolvasása célszerű a továbbhaladás előtt:

¹ Paul Haine: Tipográfia a weben című cikke alapján

- Kámán Veronika: A jelen forradalma: olvasás weben
- Kovács Balázs: Írás és olvasás a weben

1.1.3. Kereső(re) optimalizálás

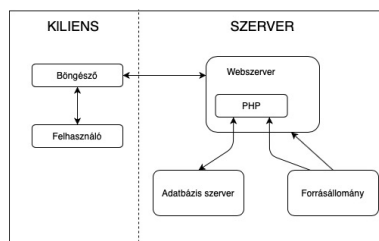
Ha egy weboldalt fáradságos munkával elkészítettünk, szeretnénk, ha minél több látogató megtalálná az oldalunkat. Aki elolvassa a cikkeinket, hozzászól a blogunkhoz, vásárol a termékeink közül.

A látogató „szerzése” minden honlapnak célja. A látogatószerzés klasszikus módja a keresőmotorokban (pl. Google) való megjelenés, mégpedig minél előkelőbb helyen az általunk hön áhított keresőszavakra. Bár a témával foglalkozó írások, weboldalak, vállalkozások nem mindig tesznek különbséget a keresőoptimalizálás és keresőmarketing között, itt ezt megtesszük.

- A keresőre optimalizálás a saját oldalunk fejlesztésével történik. Emiatt minden weboldal-tulajdonosnak szüksége van rá. Mi itt erre tudunk koncentrálni.
- A keresőmarketing sok egyéb eszközt (pl. Hírlevél, fizetett hirdetések) is felhasznál, amelyek nem képezik a weboldalunk részét. (Ez a terület nem témája a könyvnek.)

1.2. A web működése

Az 1.1. ábra sokat segíthet a további információk megértésében.



1.1. ábra. A kliens-szerver architektúra

A *felhasználó*, aki a web szolgáltatásait ki akarja használni, megteheti ezt egy tetszőleges modern *webböngészővel*. (E két „szereplő” együttesen a kliens oldalnak tekintjük.)

A felhasználó a böngészőt használva kezdeményezheti egyes weboldalak letöltését. A web kezdeti időszakában a webszerver azokat az álleányokat tudta kiszolgálni, amiket a háttértárain elhelyeztek. (Ez tulajdonképpen *statikus tartalmat*

eredményez, vagyis az ilyen tartalom jellemzően nem változik.) Bizonyos esetekben ez ma is így van: például egy honlapba illesztett kép nem fog megváltozni, akárhányszor töltjük is le, ezért a sebszervernek a böngésző kérésére válaszul mindössze vissza kell adni.

Később egyre nagyobb igény lett a *dinamikus tartalmak* iránt, amikor a tartalom már a látogató tevékenységei, vagy más okok miatt színesebb, változóbb lehet. Ebben az esetben a webszerver nem önmaga válaszol a böngésző kérésére, hanem PHP, vagy más nyelvű program állítja elő a választ, amit a webszerver csak továbbít.

Tovább növelheti az oldal dinamizmusát, ha a tartalmak előállításához szükséges adatokat (legalább részben) *adatbázisban tároljuk*. Ekkor a PHP nyelvű forrásprogram az adatbázisszerverrel kapcsolatot épít fel, és adatbázisból származó információkat is felhasznál a válasz elkészítéséhez, illetve a felhasználók válaszait is eltárolja az adatbázisban.

1.2.1. Webszerver

A webkiszolgáló/webszerver egy kiszolgáló, amely elérhetővé teszi a rajta tárolt weboldalakat HTTP protokollon keresztül. A sebszerverekhez sebböngészőkkel lehet kapcsolódni.

Bár a sebszerverek sok mindenben különböznek, az alapvető funkcióik azonosak. Minden webszerver HTTP kéréseket fogad a hálózatról, és HTTP válaszokat küld vissza. A HTTP válasz az esetek többségében egy HTML dokumentum, de lehet még egyszerű szöveges fájl, kép, vagy más típusú fájl is.

$$f: \mathbb{R} \setminus \left\{ (2k+1)\frac{\pi}{2} : k \in \mathbb{Z} \right\} \rightarrow \mathbb{R}, \quad f(x) := \begin{cases} \frac{\operatorname{tg}(x) \sin(x)}{x+1}, & \text{ha } x \neq -1, \\ 0, & \text{különben.} \end{cases} \quad (1.1)$$

A webszerverek a klientsől kapott kérésben többek között *URL címet* kapnak, melyet aztán két féleképpen értelmezhet a szerver a beállításaitól függően:

1. A tartománynév után álló relatív mappa és fájl struktúrát hozzárendelik egy gyökérmappához. (A gyökérmappa a webszerver beállításaiában van megadva, és az adatokat kérő kliens számára láthatatlan.)
2. A tartománynév után álló relatív mappa és fájlstruktúra (vagy akár még tartomány név is) teljesen független a kért címben szereplő struktúrától. Ebben az esetben szerver meghatározott szabályok szerint formázza a kért címet. Ennek segítségével egy mappára irányuló kérés teljesen más mappára vagy akár egy fájlra is mutathat és fordítva.

A kliens például az alábbi URL-t kéri: *http://www.pelda.com/utvonat/fajl.html*

2. fejezet

A tartalom és a kinézet

A weboldal eredeti, és máig legfontosabb célja a tartalmak közzététele. Erre a HTML mellett a CSS nyelvet használják.

A HTML a tartalom szerkezetét, a CSS pedig a kinézetét írja le. E kettő szorosan összefügg, de megfelelő tervezéssel precízen el is választható egymástól.

2.1. HTML alapok

A HTML nyelv az az alap, amivel minden webfejlesztőnek először meg kell ismerni, és alaposan tisztában kell lenni. Ez a fejezet segítséget ad a HTML lehetőségeinek megismeréséhez, de több nyelvi elem bemutatásától is eltekint. Ennek főbb okai:

1. Bizonyos HTML jellemzők a mai napra elavultnak tekinthetők. Itt előszörban a kinézet esztétikai megjelenésére kell gondolni. A CSS használatával ugyanis sokkal több és jobb lehetőségünk lesz a kinézet leírására. A HTML a mai gyakorlatban már tisztán csak az információra, és annak struktúrájára figyel. Ezt *szemantikus kódolásnak* is nevezzük.
2. Bizonyos tagok, tulajdonságok a böngészők által nem egységesen támogatott, így ezeket a gyakorlatban is csak kevesen használják.

2.1.1. Mi az a HTML?

- A HTML a Hyper Text Markup Language rövidítse
- A HTML állományok egyszerű szövegállomány, amely rövid jelölő tagokat tartalmaz
- A jelölő tagok alapján tudja a böngésző, hogyan kell megjeleníteni az oldalt

- A HTML állományok html kiterjesztéssel rendelkeznek
- A HTML állományt egyszerű szöveges (editor) programokkal (pl. Jegyzet-tömb) is létrehozhatunk

2.1.2. Hogyan kezdjük neki?

Windows operációs rendszer alatt indítsuk el a Jegyzettömböt, majd gépeljük be a következő szöveget:

```
1 <html>
2   <head>
3     <title>Az oldal címe</title>
4
5   </head>
6   <body>
7     Ez az első honlapom. <strong>Ez a
8       ↳ szöveg kiemelt.</strong>
9   </body>
10 </html>
```

Mentsük el a szöveget index.html néven!

Ebben és a következő néhány fejezetben még nincs feltétlenül szükségünk web-szerver elérésre a tanuláshoz. Később majd FTP kapcsolaton keresztül a webszerverre fogjuk az oldalainkat feltölteni, majd webszerver elérésével tesztelni azokat.

Nyissuk meg a böngészőnket, majd a *Fájl* menü *Webcím megnyitása* parancsát választva keressük meg az előbb elmentett index.html állományt! (További lehetőségünk, hogy a *Windows intőzőben*, vagy *Total Commanderben* duplán kattintunk az állomány nevére. De az állomány böngészőre vonszolásával is célt érhetünk.)

A dokumentum első tagja a html. A böngésző erről fogja tudni, hogy hol kezdődik a HTML oldal. A párjával fogja tudni, hogy itt ér véget a dokumentum a böngésző számára. A head tagok közötti rész a fejléc információ. Az itt leírt szöveget a böngésző nem jeleníti meg közvetlenül. A title tagok közötti szöveget a böngésző címsorában jeleníti meg, ahogy az ábrán is láttuk. A body tagok közötti szöveg jelenik meg a böngésző ablakában. A strong tag hatására a szöveg kiemelten (félkövéren) jelenik meg.

2.1.3. A head tagba írható elemek

- Kötelező
 - title:

- Ide írjuk a weboldal címét ami a fülön fog megjelenni
- Nem kötelező
 - meta:
 - Ide írhatjuk az oldal karakter kódolását
 - style:
 - Segítségével formázható szöveg jeleníthető meg
 - script:
 - Segítségével valamilyen beépített kódot futtathatunk
 - link:
 - Ezzel az elemmel HTML lapok közötti kapcsolatot jelölünk
 - base:
 - Az URL tulajdonképpen az aktuális oldal pontos helyét rögzíti. Ezzel biztosítható, hogy a környezetéből kiragatott oldalunkon belül is működő relatív linkek legyenek
 - isindex:
 - Használatával jelezhető, hogy a dokumentum csak információkérő oldal. Kiír egy adatkérésre figyelmeztető szöveget, ami a PROMT paraméter segítségével felülírható. Elég ritkán használt elem.

2.1.4. HTML szerkesztők

2.1. Definíció. Amit látsz azt kapod, (what you see is what you get).

Léteznek olyan szerkesztőprogramok, amelyekkel tényleges HTML ismeretek nélkül is lehet HTML oldalakat létrehozni. Ezeket a programokat *WYSIWYG* (jelentése a 2.1. definícióban olvasható) editoroknak hívjuk. Ezek azonban kerülendőek, ha minőségi HTML oldalakat szeretnénk létrehozni. (Legalábbis a tanulás kezdeti fázisában.) Ezek a programok ugyanis kisebb-nagyobb mértékben „teleszemetelik” a kódot.

2.1.5. Hogyan nézhetjük meg egy oldal HTML kódját?

Gyakran előfordul, hogy a weben böngészve megtetszik egy oldal és szeretnénk megnézni a forrását. (A szerző véleménye szerint ez az egyik legjobb módszer a tanulásra.) Hogyan tehetjük ezt meg?

Keressük meg a böngészőnk *Nézet* menüjét, majd *Forrás*, vagy *Oldal forrása* (vagy valami hasonló nevű, böngésző függő) menüpontot.

A szerző javasolja a fejlesztéshez a *Firefox* nevű böngészőt, amely eleve webfejlesztők számára lett kifejlesztve és több ezer kiterjesztése (plug-in) közül jó néhány HTML forrás könnyen áttekinthető megjelenítését szolgálja. Lásd [1, 33. oldal]

Irodalomjegyzék

- [1] NAGY GUSZTÁV: *Web programozási alapismeretek*, Budapest, 2011, Ad Librum Kiadó.