

Tartalomjegyzék

1. Bevezető.....	4
2. A webdesignról általánosan.....	6
2.1. A webdesign kialakulása.....	6
2.2. Oldaltervezés, valamint a megfelelő navigáció.....	8
2.3. Az adatpaca és a képszemét fogalma.....	10
2.4. Felbontás független tervezés.....	10
2.5. Néhány gondolat a válaszdíóvel kapcsolatban.....	11
2.6. Az oldalak nyelvezete.....	11
3. HTML/XHTML.....	13
3.1. A HTML története.....	13
3.2. A HTML felépítése.....	15
3.2.1. A fejrész.....	16
3.2.2. A törzs.....	17
3.2.2.1. Címek, formázó tagek, a karakterek különböző használata.....	17
3.2.2.2. A linkek.....	19
3.2.2.3. A keretek.....	20
3.2.2.4. A táblázatok.....	20
3.2.2.5. A listák.....	21
3.2.2.6. Az úrlapok.....	22
3.2.2.7. Képek használata, az oldal háttere.....	25
3.3. A HTML és az XHTML közötti lényeges különbségek.....	25
4. CSS (Cascading Style Sheets).....	27
4.1. Mi az a CSS?.....	27
4.2. A CSS története.....	27
4.3. Miért használjunk CSS-t?.....	28
4.4. CSS hozzákapcsolása HTML-hez, vagy XHTML-hez.....	30
4.4.1. Szövegek stílusok.....	30
4.4.2. Beágyazott stíluslapok.....	30
4.4.3. Külső stíluslapok.....	31

4.4.3.1. Az @import utasítás.....	31
4.4.3.2. A link elem.....	31
4.4.3.3. Az xml-styleheet feldolgozási utasítás.....	32
4.5. A stílusok felépítése.....	32
4.6. Stílusosztályok.....	33
4.7. Pszeudo-osztályok és pszeudo-elemek.....	33
4.8. Öröklődési szabályok.....	34
5. A JavaScript.....	36
5.1. A JavaScript története, a nyelv főbb jellemvonásai.....	36
5.2. JavaScript előnyei, hátrányai.....	37
6. Flash.....	39
6.1. A Flash rövid története.....	39
6.2. A Flash előnyei, hátrányai.....	39
6.2.1. A bittérképes grafika (pixel grafika).....	41
6.2.2. A vektorgrafika.....	42
7. Felhasználóbarát webdesign.....	43
7.1. Olvashatóság.....	43
7.2. Színek megfelelő használata.....	44
7.3. Grafikus elemek.....	46
7.3.1. Állandó előtérrel készülő animáció.....	47
7.3.2. Állandó háttérrel készülő animáció.....	47
7.3.3. Az objektum animáció.....	48
8. Webdesign hibák.....	49
9. Keresőoptimalizálás.....	52
9.1. Mi az a keresőoptimalizálás.....	52
9.2. A keresőoptimalizálás története.....	52
9.3. A keresési folyamat.....	54
9.4. A keresőoptimalizálás célja.....	55
9.5. Keresőoptimalizálás lépésről lépésre.....	55
9.5.1. Konkurenciaelemzés.....	55
9.5.2. Megfelelő kulcsszavak megtalálása.....	56
9.5.2.1. Hogyan gyűjtsünk lehetséges kulcsszavakat?.....	56
9.5.2.2. Hogyan válasszuk ki a fontos kulcsszavakat?.....	57

9.5.3. Domain-név kiválasztása.....	58
9.5.4. Értékes tartalom, vagy szolgáltatás hozzáadása.....	59
9.5.5. Tartalomkezelő rendszerek használata.....	60
9.5.6. A megfelelő tárhely kiválasztása.....	61
9.5.7. Keresőoldalakra való regisztrálás.....	61
9.5.8. Jól megírt weboldal létrehozása.....	62
9.5.9. Link-népszerűség növelés.....	63
9.5.10. A tartalom rendszeres frissítése.....	63
9.5.11. Nyomonkövetés.....	64
10. Tartalomkezelő rendszerek.....	65
10.1. Mik azok a tartalomkezelő rendszerek?.....	65
10.2. Tartalomkezelő rendszerek osztályozása.....	65
10.3. A Joomla! tartalomkezelő rendszer.....	66
10.3.1. Rendszerkövetelmények.....	67
10.3.2. Telepítés.....	67
10.3.3. Bővítmények.....	68
10.3.3.1. Bővítménytípusok.....	69
10.3.3.2. Alapfunkciók a bővítményekben.....	70
10.3.3.3. Bővítmények telepítése.....	71
10.3.3.4. Bővítmények adminisztrációja.....	71
10.3.3.5. Bővítmények eltávolítása.....	73
11.SportRiadó – a legfrissebb hírek egy helyen	74
11.1. Rövid leírás.....	74
11.2. Tervezési szempontok.....	75
11.2.1. Felhasználóbarát tervezés.....	75
11.2.2. Keresőoptimalizálás.....	77
11.3. A tartalomkezelő rendszer és a kész oldal.....	80
Összefoglalás.....	81
Irodalomjegyzék.....	82

1. Bevezető

Az informatika hajnalán a számítógépek csak a tudósok kiváltásagai voltak. Később, mint minden forradalmi újítás egyre nagyobb körben vált ismertté, megjelentek a személyi számítógépek, így ezeket már bárki használhatta.

Mivel az ember alapvetően társas lény, ezért újabb igény került előtérbe, a számítógépek egy olyan hálózatának megvalósítása, amely az egész világot átöleli. Megszületett a világháló, egyre nagyobb teret hódított, olyannyira, hogy már a mindennapi életünk szerves részét alkotja.

A kezdeti információ megosztás túl kevésnek bizonyult. Az emberek egyre többet és többet vártak el a weblapoktól, megjelentek a színes ábrák, képek, multimédiás alkalmazások, de ez sem volt elég. Ma már ott tartunk, hogy vásárolhatunk az interneten, beszélgethetünk ismerőseinkkel, játszhatunk idegenekkel, konferenciabeszélgetéseket is lebonyolíthatunk, sőt, akár otthonról is dolgozhatunk. Elmondható tehát, hogy a web hatalmas változáson ment keresztül. Minél szélesebb körben vált elérhetővé és népszerűvé, annál nagyobb igény lett arra, hogy emberközelibbek, barátságosabbak, látványosabbak és ergonomikusabbak legyenek az egyes weboldalak.

Szaktervezésünk célja a következő: egyrészt be szeretnénk mutatni azokat a technológiákat, amelyek a weblapszerkesztés alappilléreit jelentik, másrészt rávilágítunk azokra az alapvető szabályokra, amelyeknek egy honlap készítésénél mindenképpen be kell tartani. Ha ezekkel a szabályokkal tisztában van valaki, nyugodt szívvel foghat munkához.

Azonban nemcsak az a cél, hogy a weblapok száraz tartalommal rendelkezzenek, a majdani látogatónak valamilyen élményt is kell nyújtani. Ezért az animációk világába is betekintünk, valamint a legfontosabb webdesign jellemzőkre és hibákra is rávilágítunk.

A mai világban, ahol jóformán minden a pénz és a konkurencia mögénk utasítása körül forog, már az sem mindegy, hogy mennyien látogatják honlapunkat és milyen úton jutnak el hozzá. A keresőoptimalizálásról szóló fejezetben igyekszünk bemutatni minden olyan műveletet, trükköt, amelyet alkalmazásával a konkurens weboldalak elé kerülhetünk az internetes keresők találati listáin.

Végül, de nem utolsósorban rövid betekintést kívánunk nyújtani a tartalomkezelő rendszerek világába, amelyek segítségével szinte bárki létrehozhat profi honlapokat anélkül, hogy komoly informatikai tudással rendelkezne.

A szakdolgozat "A webdesignről általánosan", "HTML/XHTML", "A JavaScript", "Flash", illetve "Felhasználóbarát webdesign" című fejezetei Baráth Balázs-Állmos munkái, míg a "CSS (Cascading Style Sheets)", "Webdesign hibák", "Keresőoptimalizálás", illetve a "Tartalomkezelő rendszerek" című fejezeteket Tenkei Szabolcs-Levente dolgozta ki. Az utolsó fejezetben mutatjuk be a munkánk során elkészített weblapot.

2. A webdesignről általánosan

2.1. A webdesign kialakulása

A webdesign, mint fogalom, két másik fogalom kereszteződéséből jött létre. A web jelentése nem más, mint a világméretű hálózat. Az Internet egyik nagyon gyakran használt része, elsősorban HTML dokumentumokat és grafikus internetes, tehát webes felületeket foglal magába. A design, mint fogalom a következőket takarja: esztétikai érzék, egyensúly, egységesség, meglepetés.

A szó maga több értelemmel rendelkezik, mivel a következőket jelentheti: terv, minta. Ha két szót összekötjük, egy új fogalmat kapunk, melynek a jelentése nem más, mint a design használata internetes felületen. Legtöbbször a szépséggel tévesztik össze, de sokkal több ennél, a tartalom, a szerkezet és a megjelenés egységessége és harmóniája.

Egyszerre lehet szerkesztési alapelv és terv, de elsősorban értékelési szempont.

A honlap készítés és a webdesign átfedésben van egymással, de nem ugyanazt jelentik, mivel az első maga a folyamat, létrehozunk valamit, míg a második a végtermékre vonatkozik, a már kész honlapra és nem más, mint az elvárások, igények egy gyűjtőcsoportja. Ma már nagy az igény arra, hogy ne csak információt szerezzen a felhasználó egy adott oldalon, hanem a megjelenítés is igényes legyen.

A tartalom értéke nem függ a formától, de egy szép formába zárt értékes tartalom nagyobb élményt nyújthat a felhasználó számára.[1]

A webes tervezést kétféleképpen lehet megközelíteni: művészi, illetve mérnöki szemszögből. Mivel a legtöbb számítástechnikai szakember nem ért az elsőhöz igazán, gyakran előfordul, hogy művészeket kérnek fel ezen rész megvalósításához.

A weboldal nem más, mint egy szöveges dokumentum, amelyet egy megfelelő böngészőprogram értelmez.

A weboldalak vonatkozásában háromféle típust különböztetünk meg:

- statikus oldalak
- dinamikus oldalak
- dinamikusan létrehozott oldalak

A statikus oldalak elkészítése a legegyszerűbb, mivel legtöbbször csak szöveges információt tartalmaz, némely esetleg képeket is. Az ilyen oldal látogatója nem tudja befolyásolni a bemutatott információk struktúráját.

A dinamikus oldalak esete más az előzőhöz képest, mivel ezek interaktívabbak, reagálnak a felhasználó cselekedeteire. Az ilyen oldalak forráskódjában a HTML-nyelv kódján kívül más kódok is vannak, az úgy nevezett script kódok (JavaScript), illetve a stílus kódok (Cascading Style Sheets), ezek segítségével „kel életre” oldalunk. A böngészők más-más módon értelmezik és jelenítik meg az egyes oldalakat, ezért az oldal forráskódját lehetőleg úgy kell megszerkeszteni, hogy a különböző böngészőprogramok megközelítőleg azonos módon jelenítsék meg őket.

A dinamikusan létrehozott oldalak esete más, mivel az ilyen oldalak forráskódját az oldal látogatóinak igénye szerint a szerver állítja össze. A látogató megadja a keresési kulcsszót, a szerver feldolgozza, majd a keresési eredményt visszaküldi a böngészőprogramnak, a találatok formájában.

Ilyenek például az online-áruházak, a felhasználó kiválasztja a terméket, ezután a kosárba teszi, és több lehetősége van, ha szüksége van a termékre, megveszi, ha meggondolná magát, akkor kiürítheti kosarát. Másik ilyen példa az online-szótárak esete. Ezen oldalak mind olyan elven működnek, hogy a felhasználó megadja a feltételeket, a szerver pedig ezen feltételek alapján elküldi a böngészőprogramnak az összeállított honlapot. Az ilyen oldalak leggyakrabban ASP (Active Server Pages), illetve PHP (Hypertext Preprocessor) technológiával készülnek. Nagyon lényeges tulajdonságuk, hogy támogatják az adatbázisokkal való együttműködést.

A legfontosabb egy oldal tervezésekor a használhatóság figyelembevétele, mivel a weben másképp működnek a dolgok, mint a hagyományos esetekben. Hagyományos termékek, szoftverek esetén a felhasználó, vásárló előbb veszi meg az adott terméket, és aztán szembesül magával a termékkel, míg a weben ez pont fordítva működik: a felhasználó előbb megfigyeli, hogy mennyire használható egy oldal, és csak ezután fizet. Tehát az egyik legfontosabb tényező a felhasználóbarát szemlélet, ha egy oldal látogatója nem jön rá egyből, hogy hogyan működik egy oldal, tovább áll egy konkurens oldalra.

Jakob Nielsen szerint a világháló sikere az ergonómiától függ. Ha a vásárló nem találja meg a keresett terméket, szolgáltatást, hírt, stb., akkor nem fogja megvenni. A web

hatalommal ruházza fel a vásárlót, aki a kattintásaival mindent eldönthet. Ha megteheti, meg is teszi, mivel a világ összes konkurens cége egyetlen kattintással elérhető.[2]

Igazából négy fő oka van annak, ha valaki a későbbiekben visszatér egy oldalra ahelyett, hogy másikra menne. Ezek a következők:

- gyakori frissítés
- sebesség
- egyszerű használhatóság
- minőségi tartalom

Most nézzünk meg pár általános szemléletet, amely igaz egy jó webdesignnal rendelkező oldalra.

2.2. Oldaltervezés, valamint a megfelelő navigáció

A legfontosabb tényező egy oldal tervezésekor, hogy a tervezendő oldalnak egyszerűnek kell lennie, minimális zavaró tényezővel, valamint nagyon világos információ-architektúrával és megfelelő navigációs eszközzel.

A legfontosabb része egy webhelynek a főoldala, ez nem más, mint az egész webhely külső arculata, mivel a látogató ezzel találkozik legelőször, amikor a weboldalunkat látogatja. A fő oldal és a többi lap stílusa meg kell egyezzen, viszont kellene legyenek különbségek is. Nagyon gyakori hiba például az, hogy a főoldalon elhelyeznek egy „Home” feliratú gombot, holott ott vagyunk rajta. Egy főoldal két kérdésre kell tudjon válaszolni, először is a felhasználó tudja megállapítani, hogy hol van, másodszor pedig rá kell tudjon jönnie arra, hogy mire jó az általa meglátogatott oldal. Tehát a főoldal igazából nem más, mint a teljes weblap egy vázlata, a cél az, hogy aki először látogat az oldalra, az ránézésből tudja megállapítani, hogy mire jó az oldal, anélkül, hogy valaki a szájába rágná.

Tervezéskor nagyon fontos, hogy úgy viszonyuljunk főoldalunkhoz, mintha mi is látogatók lennénk, így megkönnyítjük saját dolgunkat is. Emellett át kell gondolni a nevet, valamint a logót is, mivel egy főoldal legszembeűnőbb elemei ezek kell legyenek. A site nevének minden belső oldalon szerepelnie kell, de ezen oldalak tartalmazzák inkább a konkrét információkat, mivel a főoldal csak az általános dolgok leírására szolgál.

Egy főoldalnak három fő tulajdonsággal kell rendelkeznie, amelyek a következők:

- a site tartalmát összefoglaló lista
- a legfontosabb hírek, reklámok feltüntetése
- egy megfelelő kereső

Egy website-hoz több belépési pontot kell biztosítani, mivel nem kényszeríthetjük a látogatót arra, hogy csak a főoldalon kerüljön kapcsolatba oldalunkkal. Mivel a főoldal sosem olyan részletes, mint a belső oldalak, ezért, ha csak itt biztosítanánk belépési pontot, értékes időt vennénk el a látogatótól, mivel nem kapna teljes információt az oldalról, és keresgélne kellene, a weben pedig a legfontosabb tényező a felhasználó ideje.

A világháló igazából nem más, mint egy navigációs rendszer. A legalapvetőbb interakció a kattintás, de a látogatókat támogatnunk kell különféle navigációs eszközökkel, amelyeknek a következő három alapkérdésre kell válaszolniuk:

- a) hol jár a felhasználó?
- b) hol járt már a felhasználó?
- c) valamint hova mehet tovább?

Az első kérdés megválaszolásánál fontos, hogy a látogató helyzetét két különböző módon kell nyilvántartani: először is hogy a teljes web egészéhez képest hol van, másodsor pedig hogy az adott site-hoz képest hol jár.

Elsőszámú szabály, hogy minden oldalon legyen ott a logó, amelyet következetesen kell elhelyezni, lehetőleg a bal felső sarokba. Ennek úgy kell működnie, mint egy hiperszöveg hivatkozás, azért, hogy a látogató bármelyik oldalról elérhessék a főoldalt.

Másik fontos szabály, hogy az oldalcímeknek minden egyes oldalon értelmesnek kell lenniük, abból a megfontolásból, hogy egyrészt a felhasználók könnyen tudják visszakeresni őket a könyvjelzők között, másrészt azért, hogy tudják aktuálisan hol vannak.

A második kérdés azért fontos, mert ha a látogató tudja, hogy hol járt már, akkor nem fog még egyszer ugyanoda visszamenni, csak ha szándékában áll, időt spórolunk meg neki ezáltal. A látogató az előzménylistából visszanezheti, hogy hol járt már, míg a „Vissza” gomb rögtön az előző oldalra viszi.

Ha egyértelmű mely linkek vezetnek az előzőleg már látogatott oldalakra, akkor egyrészt támogatjuk a látogatót abban, hogy könnyebben megtanulja a site szerkezetét, másrészt időt takarítunk meg neki.

A harmadik kérdésre a navigációs lehetőségek, valamint az oldalon található többi link nyújt választ. Háromféle hivatkozást lehet egy oldalon használni, amelyek a következők:

- beágyazott link
- strukturális link
- asszociatív link

A beágyazott link azt jelzi, hogy hol találhatunk bővebb információt a szöveg fő részében tárgyalt témáról. Ez a szokásos aláhúzott szöveg.

A strukturális rész az oldalszerkezet különböző részeire mutat, a testvér-, valamint a gyermekoldalak érhetőek el általa.

Az asszociatív link arra jó, hogy a látogatott oldalhoz hasonló témájú oldalakra jusson a felhasználó, egyfajta kitekintőként is felfogható.

Tehát az oldaltervezés célja az egyszerűség. A látogatók ritkán néznek egy oldalt csupán azért, hogy gyönyörködjenek a kinézetében, nekik a legfontosabb a tartalom. Biztosítanunk kell azt is, hogy oldalunk különféle platformokon is megjeleníthető legyen.[2]

2.3. Az adatpaca és a képszemét fogalma

A webdesign vonatkozásában nagyon fontos fogalom ez a kettő, el kell tudni határolni a hasznos információkat a feleslegesektől. Edward Tufte 1983-as „Mennyiségi információ vizuális megjelenítése” (The Visual Display of Quantitative Information) leírta véleményét az agyondíszített üzleti illusztrációkkal kapcsolatban, melyeknek helyigénye nagy, viszont információs tartalma alacsony. Ő vezette be az „adatpaca” fogalmát az adatok alátámasztására vonatkozó illusztráció leírására. Szerinte minden más csak képszemét. Igaz, hogy tanulmánya nem a webre vonatkozott, de igaz a webes oldalakra is. Tehát a lényeg az, hogy el kell hagynunk a felesleges dolgokat annak érdekében, hogy az oldalunk látogatója csak a lényeges információra összpontosíthasson.[2]

2.4. Felbontás független tervezés

A felbontás független tervezés legfontosabb alapelve, hogy ne kötött képszélességet alkalmazzunk, mivel nem tudhatjuk, hogy a felhasználó milyen megjelenítési eszközt használ, hanem a képernyőn lévő hely százalékában határozzuk meg a méreteket.

Másik fontos tényező az is, hogy figyeljünk oda a betűméretekre is, mivel vannak olyan felhasználók, akik a kisebb betűméretet preferálják, míg mások a nagyobbakat, ennek különböző magyarázatai lehetnek, például a csökkent látás. Nagyon fontos arra odafigyelni, hogy az oldal designja ne változzon meg különböző betűméretek mellett sem. A felbontás kérdése a betűk mellett vonatkozik a grafikus elemekre is, hisz minél nagyobb a felbontás, a grafikus elemek kisebbek lesznek.[2]

2.5. Néhány gondolat a válaszidővel kapcsolatban

Az oldalak vonatkozásában a legfontosabb tényező ez, hiszen rendelkezhet akármilyen designnal, ha túl lassan töltődik be, a látogatónak elfogy a türelme, és más oldalra megy át. A válaszidővel kapcsolatban ma is érvényes Robert B. Miller tanulmánya, melyet 1968-ban a Fall Joint Computer konferencián fejtett ki, és amely a következő gondolatokat tartalmazza:

- egy másodperc tizedrésze körülbelül az az időintervallum, amelynél a felhasználó még azt érzi, hogy a rendszer azonnal reagál
- egy másodperc az az időintervallum, ahol a felhasználó a késedelmet már érzékeli, viszont a gondolatmenete még nem szakad meg
- tíz másodperc az a felső küszöb, amíg a felhasználó a párbeszédre tud figyelni, az ezt meghaladó időtartam után már más feladatra kezd el fókuszálni.

A sebességen túl fontos tényező a bizonytalanság. Ha felhasználó meg tudja nagyjából határozni, hogy mennyi időt kell várnia egy oldal betöltésére, vagy valami letöltésére, akkor szívesebben várakozik. Ezért, hogy ha módunkban áll, tüntessük fel a várható várakozási időt.
[2]

2.6. Az oldalak nyelvezete

Összességében tehát a felhasználók számára a legfontosabb a tartalom, de egy igényes megjelenítéssel élvezhetőbbé és akár népszerűbbé válhat oldalunk.

Az is nagyon fontos, hogy hogyan írunk az oldalunkra, milyen stílusban. Jakob Nielsen szerint három fő irányelv van ezzel kapcsolatban, amelyek a következők:

- Nagyon fontos a tömörség. Egy nyomtatott anyag témájának legfeljebb 50%-át tartalmazhatja az oldal, mivel ezen felül unalmassá válik.

- Fontos az, hogy amit írunk, az könnyen áttekinthető legyen. A felhasználók nagyon kicsi százaléka olvassa el a hosszú, összefüggő szövegeket, mivel általában ha valamit keresnek, akkor azt gyorsan akarják megtalálni. Ennek érdekében jobb ha rövidebb bekezdéseket, alcímeket és pontokba szedett felsorolásokat alkalmazunk.
- Célszerű hiperszöveget használni ahhoz, hogy egy hosszabb szöveget több oldalra tördelhessünk.

Másik oka annak, hogy a szövegek rövidebbek legyenek, mint a hagyományos, nyomtatott formában megjelenítettek az, hogy a képernyőről jóval lassabban olvasunk, mint papírról.

Emiatt fontos az is, hogy egyszerűbb nyelvezetet használjunk, mivel a felhasználó könnyebben megállapítja mi fontos számára.[2]

3. HTML/XHTML

3.1. A HTML története

A HTML 1990-ben született meg, amikor Tim Bernes-Lee megalkotta a World Wide Webet, az első web-szerveret, az első böngészőt és magát a HTML-t. Bár a HTML rengeteget fejlődött az első verzió óta, nagyon sok „tag” nem változott meg azóta sem; a mai napig is érvényesek.

A HTML egy rövidítés, melynek jelentése: HyperText Markup Language, (hiperszöveg-leíró nyelv), amely elnevezést a web szülőatyja, Tim Berners-Lee adta meg. Egy olyan leíró nyelv, amelyet a weboldalak készítéséhez fejlesztettek ki. Régebben arra használták, hogy dokumentumokat tegyenek közzé, ezeknek a dokumentumoknak a tartalmát, illetve felépítését adják meg, valamint vele határozzák meg a dokumentum különböző részeit. A HTML a weblapok különböző részeinek azonosítására elemeket (tageket) használ.

Ezek segítségével adjuk meg, hogy a böngészőprogramok hogyan értelmezzék majd a dokumentumban kijelölt részeket. A HTML ismerete alapkövetelmény minden webfejlesztő számára. A HTML nem más, mint egy programozási előírás, amely szabályozza, miként kell megírni a weblapokat, hogy a számítógépek megértsék és megfelelően megjelenítsék azokat. Egyike azon három webes szabványnak, amelyeket az alapszabványoknak tekintenek, a másik kettő a CSS és a JavaScript.

A HTML fájlok szöveges állományok, amiket a legegyszerűbb szövegszerkesztővel is szerkeszthetünk.

A régi verziókban a HTML felelt egyszerre a megjelenítésért is, valamint a tartalomért is, de ez több problémát okozott. Ennek hatására jelent meg az XHTML, valamint a CSS, szétválasztották ezt a két fogalmat, amíg az XHTML csak a tartalomért, addig a CSS csak a weblap stílusáért, megjelenítéséért felelős.

A HTML 1990-es év születése óta öt fontos állomást tartanak számon, nézzük meg ezeket részletesebben.

Az első állomás

A HTML legelső verziójában alakult ki az egész nyelv alapvető szerkezete a nyitó és záró címkékkel, a fejléccel és a törzssel. Már ennél a legelső verziónál is volt zárócímke, valamint lehetőség volt a dokumentum típus deklarációra is a kezdőcímke előtt (DTD).

A tartalom, akárcsak a mai verziókban, a törzsben helyezkedett el, létre lehetett hozni bekezdéseket, hivatkozásokat, fejléceket valamint felsorolást is. A HTML 2 annyi változást hozott az elődjéhez képest, hogy innentől különült el a böngésző által szabályozott, valamint a dokumentum szerkezetében létrehozott formázás.

A második állomás

Megjelent a HTML 3.0 verzió, ami igazából a HTML 2 kiegészítése, megjelent az űrlap, valamint a kiválasztható opció fogalma. Ez a verzió szabvánnyá vált.

Harmadik állomás

A HTML 3.2 verziót 1996-ban elfogadta el a W3C. A W3C egy konzorcium (World Wide Web Consortium), amely nyílt szoftver szabályokat, ajánlásokat alkot a világhálóra. Ezt a konzorciumot az a Tim Bernes-Lee vezeti, aki a HTML, az URL és a HTTP eredeti megalkotója, ezek a technológiák képezik ma is a web alapjait. [5]

A HTML 3.2-es verzió nagyon lényeges változásokat hozott, mivel e verzió megjelenésével lehetőség nyílt a Java Appletek, valamint a különböző scriptek beágyazására. Ezeken felül lehetőség nyílt táblázat készítésre és betűtípus beállításra is. A HTML 3.2 sok funkciót elhagyott a HTML 3.0-ból és beépítették az akkor népszerű böngészők különböző fejlesztéseit (Mosaic, Netscape Navigator).

Negyedik állomás

A HTML 4.0 verziót 1997 nyarán tette hivatalos ajánlássá a W3C és ez a verzió megfelel az ISO 8879 előírásainak. Tovább fejlesztették a táblázatok és az űrlapok használhatóságát, és ebben a verzióban vált hivatalossá a keretek (frame-ek) használata. Ezen kívül támogatja a

jobbról balra olvasást, a nemzetközi karakter készleteket, valamint fejlesztésekor figyelembe vették a csökkent képességűek érdekeit is.

Ötödik állomás

Ennél az állomásnál annyi változás történt, hogy a HTML 4-et és az XML-t (eXtensible Markup Language – bővíthető jelölő nyelv) összegyűrték, és megalkották az XHTML-t, amely már XML alapokon nyugszik és kompatibilis az XML és a HTML 4-es verziójú böngészőkkel is. Az XHTML megjelenésével együtt jelent meg a CSS (egymásba ágyazott stíluslapok) is, míg az XHTML csak a tartalomért felelős, addig a CSS csak a megjelenítésért, élesen szétszedték ezt a két fogalmat.[5]

3.2. A HTML felépítése

A HTML és az XHTML különböző elemekből álló jelölő nyelvek, amelyek tartalmazhatnak attribútumokat is (egyeseket kötelezően, másokat választhatóan). Ezek az elemek jelölik a különböző típusú tartalmakat a dokumentumokban, és megadják, hogy hogyan értelmezzék a böngészők a különböző tartalomdarabokat (ilyenek például a címsorok, a bekezdések, a táblázatok, a felsorolások, stb.). Ahogy az elvárható, az elemek egy bizonyos tartalomtípust definiálnak, míg az attribútumok további információkat adnak az elemekhez, például azonosítanak egy bizonyos elemet, vagy megadják, hogy hova mutat egy adott hivatkozás. [7]

Egy HTML dokumentum tageket tartalmaz, ezek kijelölik a HTML elemeket. A legtöbb elemnek van egy nyitó és egy záró tagje, de van, amelyet nem kötelező zárni. A tagek között van a tartalom maga. A HTML nyelv nem kis- és nagybetű érzékeny.

Egy elem a HTML-ben két jelölő tagból áll. Ezek vagy egy szövegrészt fognak közre, vagy további elemeket tartalmaznak. Az elemek ezen kívül tartalmazhatnak attribútumokat is, amelyek módosíthatják az elem működését, vagy újabb értelmezést adhatnak az elemnek.

Egy HTML kód a dokumentum típusát megadó elemmel kezdődik, ami a DOCTYPE. Ezzel az elemmel azt adjuk meg, hogy a HTML melyik változatát fogjuk használni, ezáltal a böngésző tudni fogja, hogy miként kezelje és dolgozza fel majd a dokumentumot. Ezután következik a „html” nyitó tag, innen fog kezdődni a dokumentum maga, ehhez tartozik egy

záró elem, amely a dokumentum legvégén fog elhelyezkedni. A „html” elemen belül adjuk meg a „head” elemet.[5]

3.2.1 A fejrész

Ezzel adunk meg különböző információkat a dokumentummal kapcsolatban. A „head” elemen belül helyezkedik el a „title” elem, ezzel a lap címét adhatjuk meg, amely a böngésző címsorában fog megjelenni. Fontos, hogy egy jó címet válasszunk, mert ez már önmagában növelheti a weblapunk sikerességét. Csak akkor érdemes a webre írni, hogyha az oldalunkat majd meg is fogják találni.[6]

Az oldalcím jó megválasztása azért nagyon fontos, mert ezzel utalunk az oldalunkra. Nagyon lényeges, hogy ne legyen túl hosszú, legfeljebb 6 szóból álljon, de ennél is lényegesebb az, hogy lényegretörő legyen. Fontos az, hogy az oldalcímek a lehető legjobban átláthatóak legyenek. Ezt úgy lehet megvalósítani, hogy az információhordozó kifejezéseket a sor elejére tesszük, így amikor a látogató átfutja, a számára fontos információk kerülnek először a szeme elé.

Három tényező van, amit figyelembe kell vennünk:

- A címet menti a böngésző, amikor a felhasználó felveszi az oldalt a „Kedvencek” vagy a „Könyvjelzők” listára.
- A keresőprogramok a rengeteg átnézett weblap indexelése és katalogizálása során a címet használják.
- Jó cím használata esetén a felhasználó pontosan tudni fogja az oldal megnyitásakor, hogy hová került.[2]

A „head” elemben a „title” elemen kívül lényegében semmi sem fog látszani a felhasználó számára, de mégis fontos, mivel itt lehet megadni az úgy nevezett metaadatokat a böngésző számára.

Ugyancsak a fejrészben adhatjuk hozzá az oldalunkhoz a stíluslapokat, amit a „style” elem segítségével tehetünk meg, valamint a scripteket is itt adhatjuk meg a „script” tag használatával.[6]

3.2.2. A törzs

A „head” elem után következik a „body” elem, ez lényegében a dokumentum törzse. A dokumentum törzse mindig tartalmaz további beágyazott elemeket. Az oldal felosztása adja meg a dokumentum struktúráját, és további felosztásokat tartalmaz. Ebben lesznek a címsorok, a bekezdések, a listák, a táblázatok, űrlapok, keretek, háttérképek. A bekezdések ugyancsak több különböző elemet tartalmazhatnak: hivatkozásokat más dokumentumokra, idéz őjeleket, kiemeléseket.[7]

3.2.2.1. Címek, formázó tagek, a karakterek különböző használata

Hat cím van, <h1>, <h2>, <h3>, <h4>, <h5> és <h6>. Az első a legnagyobb, legfelsőbb szintű, ez a fősort jelenti. Általában 2-3 szintű címet illik használni, máskülönben túlságosan is zsúfolt lesz az oldalunk. A bekezdést a <p> taggel adhatjuk meg. A böngésző alapértelmezetten térközöket is bekapcsol.

A sortörést a
 taggel adhatjuk meg, ezt a HTML-ben nem szükséges zárni.

Ha megjegyzést szeretnénk tenni a dokumentumunkban, akkor ezt a következőképpen tehetjük meg: <!.... -->. Ez nem fog látszani az oldalon, a böngésző figyelmen kívül fogja hagyni.

A weblapok vonatkozásában fontosak a szövegek megjelenítése. Az alábbi formázó tagek vannak a HTML-ben: [7]

- : félkövér szöveget definiál
- <big>: nagyobb szöveget definiál
- : kiemeli a szöveget
- <i>: dőlt szöveget definiál
- : erősebb kiemelés
- <small>: kisebb szöveget definiál
- <sub>: alsó indexet határoz meg
- <sup>: felső indexet határoz meg
- <ins>: beszúrt szöveget jelöl
- : törölt szöveget jelöl

Nagyon lényeges webdesign szempontjából, hogy miként használjuk a betűtípusokat. Ne legyen kevesebb mint egy betűtípus egy képernyő oldalon, mert az oldal megjelenése zavarossá válhat, mivel az olvasó az oldal szerkezetét próbálja majd megfejteni, és nem a tartalomra fog figyelni. Ha egy oldalon a szövegelemeket meg kell különböztetni jobb az a módszer, ha különböző betűméretekkel, illetve stílusokkal dolgozunk, vagy más háttérrel használunk.

A szövegstílusokkal lehet javítani egy weblap megjelenésén, de alig befolyásolja az oldal tartalmát.[1]

A címsort általában nagyobb méretű betűkből állítsuk össze. A szövegek vonatkozásában jobb, ha kerüljük a csupa nagybetűs részeket, mivel ezeket nehezebb olvasni, mint a nagy és a kis betűk kevert használatát. Ha kiemelni szeretnénk valamit, akkor érdemes a félkövér betűket használni, mivel ha ezeket korlátozottan használjuk, a szövegek áttekinthetőbbé, olvashatóbbá válnak.

A formázó tageken kívül vannak az úgy nevezett számítógép kimenetet definiáló tagek, illetve az idézet, kiemelés és definíciós tagek, amelyek a következők:

Számítógép kimenetet definiáló tagek:[7]

<code>: forráskódot definiál

<pre>: a white space karaktereket direkt módon értelmezi

Idézet, kiemelés és definíciós tagek:[7]

<abbr>: rövidítés

<acronym>: mozaikszó

<address>: cím elem

<bdo >: szövegirány

<blockquote>: hosszú idézet

<q>: rövid idézet

<cite>: idézet

<dfn>: definíció

Bizonyos karaktereket nem lehet használni, csak speciális módon, mivel a HTML-ben külön jelentéssel bírnak. Ilyenkor szükséges a karakterentitás alkalmazása. Ez három részből áll: &____; a kihagyott helyen szerepel az entitásnév, vagy ezt megadhatjuk speciális módon számkóddal, de akkor elé kell rakjuk a # jelet.

A leggyakrabban használt karakterentitások a következők [7]:

| Karakter | Entitásnév | Számkóddal | Jelentése |
|----------|------------|------------|--------------------|
| < | < | < | kisebb, mint |
| > | > | > | nagyobb, mint |
| ' ' | | | nem törhető szóköz |
| & | & | & | és |
| '' | " | " | idézőjel |
| ' | ' | ' | apoztróf |
| © | © | © | copyright |

Régen problémát okoztak az ékezetes betűk, melyeket eleinte karakterentitásokkal oldottak meg, de mára ez a probléma megszűnt, mivel bevezettek a különböző karakterkódolásokat, például az UTF-8 -ét.

3.2.2.2. A linkek

A link az az eszköz, amivel oldalunk életre kelhet, egy link egy lapon bárhol elhelyezhető, szóhoz, mondathoz, szövegrészhez, képhez kapcsolva, és aki rákattint, máris új oldalt fog kapni. Ezek a linkek technikai oldalról nem másak, mint egy-egy cím, a kapcsolódó anyag URL-je, lelőhelye.

A linkeket az úgy nevezett horgony taggal <a> lehet elhelyezni. Ennek a tagnek a főbb tulajdonságai a következők:

- a href tulajdonság, amely a következőre való: az ehhez rendelt érték megmondja, hogy a böngésző miként reagáljon a link kiválasztására. Ez a tag hivatkozni tud egy tetszőleges erőforrásra, például HTML oldal, kép, zene, stb.
- a target tulajdonság, amelynek az a hatása, hogy alapértelmezetten az új oldal az aktuális helyett jelenik meg.
- a name tulajdonság a következőre szolgál: alapértelmezetten a HTML oldal teteje egy hivatkozás, de néha pontosítani kell a link célját, erre való ez a tulajdonság.[7]

Webdesign szempontjából érdemes a linkekkel a következőképpen eljárni: mivel a weben általánosan a már látogatott oldalak linkje piros vagy bordó színű, míg a még meg nem látogatottaké kék, ezért törekedjünk arra, hogy az oldalunkon található linkek is ennek megfelelően viselkedjenek. Ha nem így teszünk, akkor a felhasználók valószínűleg

összezavarodnak, mivel nem fogják tudni, hogy hol jártak már, és még hol nem, ezáltal nem fogják tudni a weblapunkat teljes egészében megtekinteni.[2]

3.2.2.3. A keretek

Keret (frame) definiálásával a böngésző ablakát több részre oszthatjuk. Ez lényegében azt jelenti, hogy több HTML oldal van egymás mellett egy ablakon belül.

A keretekre vonatkozóan legfontosabb elemek a következők:

- frameset elem: ennek a tagnek a segítségével vízszintes, illetve függőleges irányban oszthatjuk fel az ablakot; ilyenkor nincs szükség a <body> elem használatára.
- frame elem: egy-egy ablakrész milyen HTML állományt jelenítsen meg.
- no frames elem: a keretet nem ismerő böngésző esetén megjelenő tartalom.
- iframe elem: soron belül keretet definiál[7]

3.2.2.4. A táblázatok

Ha sok adatot akarunk táblázatokban közölni, akkor a table elemet kell alkalmaznunk.

A táblázat a <table> taggel kezdődik és a </table> tag zárja. Ezen belül a sorokat a <tr>, az oszlopokat a <td> határolja el egymástól. Ha a táblázat oszlopneveit ki akarjuk emelni, akkor <td> helyett <thead> jelet használunk.

A table elem következő attribútumai szolgálnak a táblázat méreteinek megadásához:

- a width attribútummal a szélességet adhatjuk meg
- a border attribútum megadásával a szegélyt lehet módosítani
- az align attribútum használatával a szöveget igazíthatjuk, mégpedig a következőképpen:
 - o ha értéke 'left', akkor balra;
 - o ha értéke 'right', akkor jobbra;
 - o ha értéke 'top', akkor felülre,
 - o ha értéke 'bottom', akkor pedig alulra igazítja a szöveget.

A táblázat cellái tartalmazhat szöveget, képet, bekezdést, listát, űrlapot, sőt akár egy másik táblázatot is.

Táblázatok vonatkozásában a következő elemek fontosak még:

- `<caption>` tag: ez kell legyen az első a `<table>` -nek, ez nem más, mint a korrekt módú megadása a tábla címének;
- `<colspan>` és `<rowspan>`: cellák egyesítése;
- `<tbody>`: ezen elem segítségével definiáljuk a táblázat törzsét;
- `<tfoot>`: lábléc megadására szolgál.[7]

3.2.2.5. A listák

A listák arra valók, hogy az egymáshoz kapcsolódó adatokat egy csoportba gyűjthessük, így áttekinthetőbbé, illetve könnyebben olvashatóbbá tehetjük oldalainkat. Mindemellett gyakran használják a listákat a navigáció megvalósításához is. A listák strukturális szempontból is kiválóak, mivel a segítségükkel egy könnyen kezelhető, könnyen hozzáférhető és jól strukturált dokumentumot hozhatunk létre. [7]

A HTML háromféle listát támogat, ezek a következők: felsorolt, számozott, valamint a definíciós-listák.

- A felsorolt lista

Gyakran rendezetlen listának is nevezik. Az ilyen listákat olyankor érdemes használni, mikor a lista elemeinek sorrendje nem fontos. A felsorolt listát az `` tagekkel jelölik, amelyen belül az elemeket az `` tagekkel fogjuk közre.

- Számozott lista

Más néven rendezett lista. Itt már számít az elemek sorrendje, a lista elemei egy jól meghatározott sorrendben követik egymást. Olyan esetben érdemes használni, ha a lista elemei csak egy bizonyos sorrendben értelmesek (például valamilyen algoritmus ismertetése esetén, ahol számít az egyes utasítások sorrendje). A számozott lista alapértelmezett számozása a decimális számozás, de betűkkel is rendezhetünk. A számozott listákat az `` tagekkel jelölik, amelyen belül az elemeket az `` tagekkel fogjuk közre.

- Definíciós lista

A definíciós listák különböznek a többi listától, mivel kifejezéseket és azok definícióit tartalmazzák egyszerű listaelemek helyett.

A definíciós listákat a <dl></dl> tagek határolják. Ezen belül meg kell adni legalább egy <dt></dt> elemet (a kifejezésnek) és egy <dd></dd> elemet (a definíciónak), mindemellett a <dt></dt> elem mindig az első kell legyen a listában.[7]

3.2.2.6. Az űrlapok

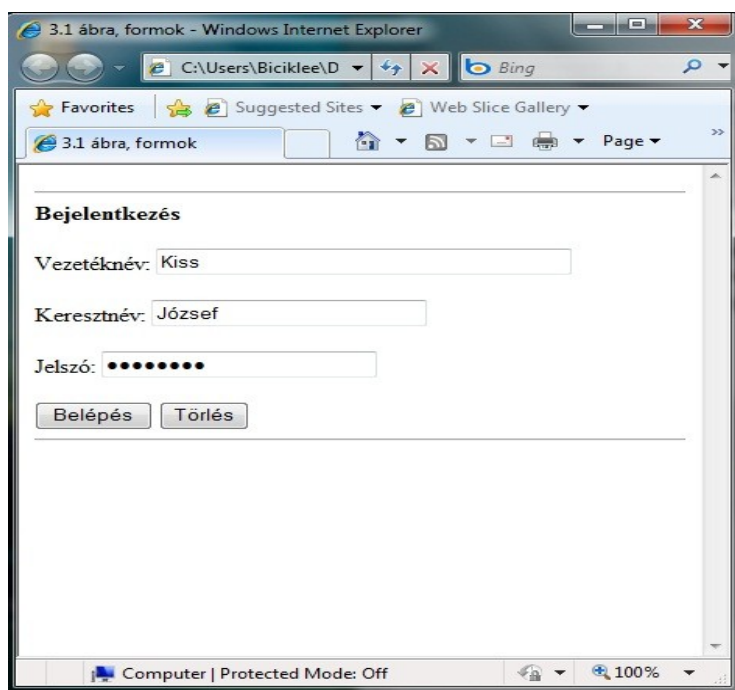
Az űrlapok arra valók, hogy különböző módokon lehetőséget nyújtsunk a felhasználóknak visszajelzés végett, vagyis adatok megadására valók; az űrlapelemek teszik lehetővé az adatbevitelt. A <form> elem jelenti az űrlapot a HTML-ben. A <form> elem más elemeket is tartalmaz, amelyek lehetővé teszik az adatbevitelt.

a) Szöveges adatbevitel

A leggyakrabban használt elem az <input> elem, amely rendelkezik a „type” tulajdonsággal. Ezzel megadhatjuk, hogy milyen adatbeviteli módot akarunk.

Ha a type értéke „text”, akkor ez egy szöveges mező lesz, ahol a felhasználó megadhat betűt, számot, illetve írásjelekből álló karaktersorozatot. Ha a type értéke „password”, akkor ez jelszavak begépeléséhez való. Ebben az esetben a szöveg, ami be lesz gépelve * karakterrel fog látszani.

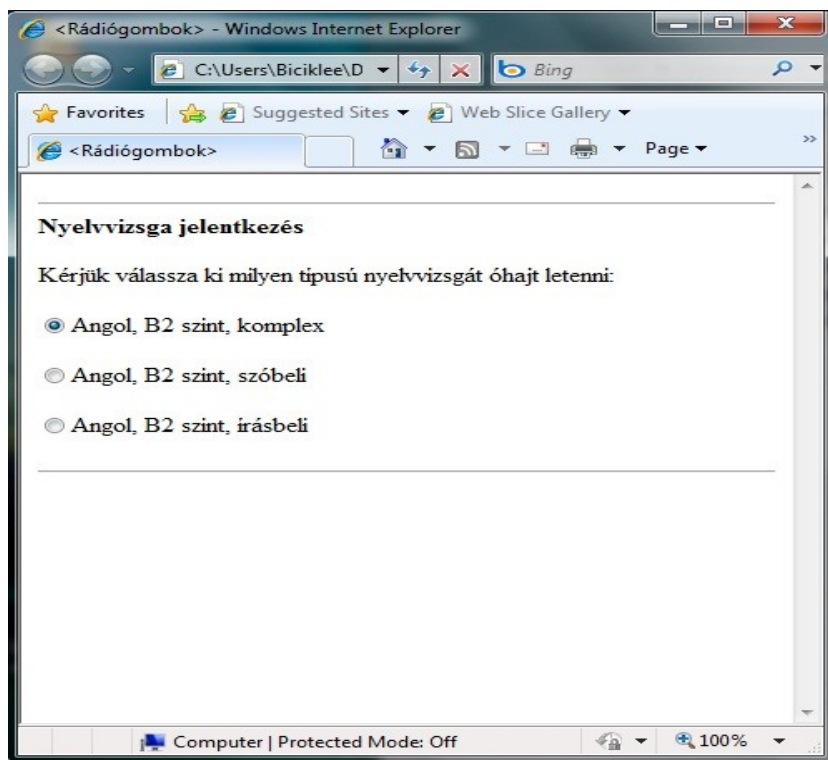
Egy űrlapon elhelyezhetünk rádiógombokat, jelölőnégyzetek, valamint lenyíló listákat is, attól függően, hogy az adatbevitel melyik formáját akarjuk használni.



3.1. ábra – Egyszerű szöveges adatbevitel

b) Rádiógombok

Rádiógombokat akkor érdemes alkalmazni, ha a felhasználónak néhány választható elem közül kell választási lehetőséget nyújtani. Az elemek közül mindig csak egy lehet aktív. Ezt a következőképpen adhatjuk meg: a type attribútum értékének a "radio" értéket adjuk meg. Rendelkezik még a name, a value, illetve a checked tulajdonságokkal is. Ha valamelyik választási lehetőséget alapértelmezettnek akarjuk megadni, akkor a checked attribútumnak a "checked" értéket kell hozzárendelni.

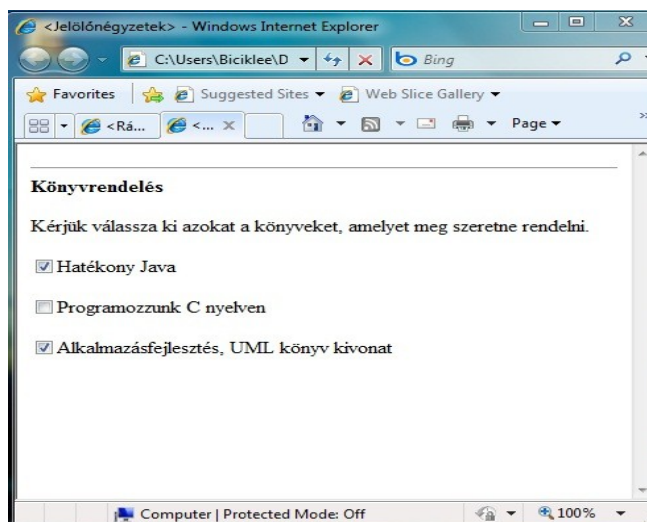


3.2. ábra - Rádiógombok

c) Jelölőnégyzetek

Itt is több elem közül választhat a felhasználó, de egyszerre akár több elem is aktív lehet, tehát egy jelölőnégyzet lehet ki-, illetve bekapcsolt állapotban, függetlenül más beviteli elemtől. Itt is, akárcsak a rádiógomboknál, lehetőségünk van arra, hogy az alapértelmezetten ki nem választott állapot helyett bejelölve jelenítsük meg a jelölőnégyzetet. Jelölőnégyzetet a type tulajdonság "checkbox" értékével adhatunk meg.

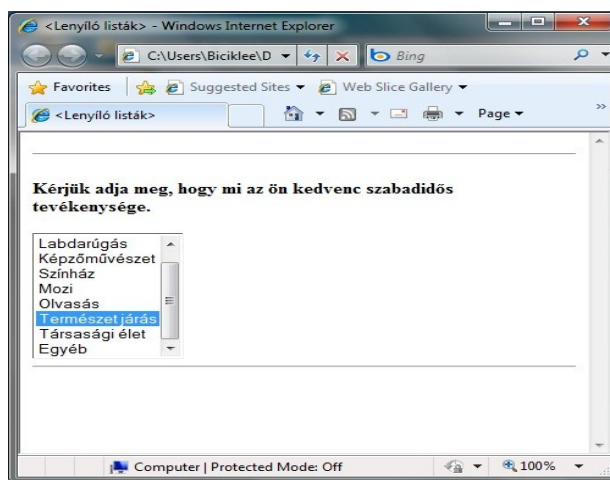
Ha a <label> elemet alkalmazzuk, akkor mind a rádiógombok, mind a jelölőnégyzetek esetén azt érhetjük el, hogy a kattintható terület ne csak egy körre, illetve egy négyzetre korlátozódjon, hanem a label elem által közrezárt szövegre is vonatkozzon.



3.3. ábra – Jelölőnégyzet

d) Lenyíló listák

Bizonyos esetben célszerűbb, ha rádiógombok helyett lenyíló listákat alkalmazunk, ennek leginkább terjedelmi oka van. A lenyíló listához két tag szükséges. Az egyik a select, amely a lista kereteit adja meg, míg a másik az option, amellyel a választható elemek adhatók meg. Rendelkezik a selected tulajdonsággal, amely az alapértelmezett elem kijelölésére szolgál, ha ez nem szerepel, akkor az első elem lesz az alapértelmezett. A <textarea > tag többsoros beviteli mezőt definiál.[3]



3.4. ábra – Lenyíló lista

3.2.2.7. Képek használata, az oldal háttere

A HTML nyelvben az `` tag segítségével tudunk képeket definiálni. Ez az elem üres, és nincs záró tagja sem.

Rendelkezik az „src” tulajdonsággal, amellyel a kép neve vagy helye adható meg. Egy kép helyét kétféleképpen adhatjuk meg, relatív illetve abszolút útvonallal.

Az „alt” tulajdonság arra szolgál, hogy ha a kép nem jeleníthető meg egy adott böngészőben, akkor az itt szereplő alternatív szöveget jelenítse meg az oldal a felhasználónak.

A „width”, illetve „height” tulajdonságokkal a beszúrandó kép méretét tudjuk megadni.

A `<body>` tag „bgcolor” tulajdonságával meghatározhatjuk az oldal háttérszínét, mégpedig háromféle módon: hexadecimális formában, RGB színkóddal, valamint a szín névvel.

A „background” tulajdonsággal megadhatjuk az oldal háttérképét, ebben az esetben is alkalmazható a relatív, illetve abszolút útvonalak használata.[7]

A háttér a weblap egyik legfontosabb eleme, mivel előkészíti az információt, meghatározza az oldal hangulatát. A háttér kiválasztása előtt mindig tisztázni kell, hogy a majdani látványnak milyen érzést kellesz közvetítenie. Egyszerű, ismertető jellegű szövegeknél célszerű az egyszerű hátterek használata, ilyenkor a tompa színek alkalmazása a javasolt. Ezzel ellentétben ha az a cél, hogy egy termék, vagy cikk köré egy meghatározott légkört teremtsünk, akkor a használhatunk a háttérben élénkebb színeket, vagy akár mintákat, képeket is.[2]

3.3. A HTML és az XHTML közötti lényeges különbségek

Az XHTML-ben írt oldalak a legnagyobb előnye az, hogy különféle böngészőkben, eszközökön közel ugyanolyan formában megjeleníthetőek. Az XHTML a következő szavak rövidítése: eXtensible Hypertext Markup Language. Az XHTML-t a HTML-ből fejlesztették ki, de több annál, mivel bővíthető. Az XHTML-ben írt weblapok könnyen együtt tudnak működni az XML-el.[3]

A következő különbségek vannak a két nyelv között:

- míg a HTML-ben a DOCTYPE meghatározása nem kötelező, addig az XHTML-ben az;
- míg a HTML-ben nem kötelező minden elemet lezárni, addig az XHTML-ben kötelező, mivel az XML nyelvtanát követi;
- az XHTML kis- és nagybetű érzékeny;
- míg a HTML-ben egyes attribútumoknak idézőjel nélkül is adható érték, addig az XHTML-ben minden értéket idézőjelben kell megadni, ami lehet egyes ('), illetve kettős ("), de ha az egyik lett választva, akkor következetesen végig azt kell alkalmazni;
- XHTML-ben minden elemet, attribútumot, illetve értéket kisbetűvel kell írni
- amíg a HTML-ben az attribútumokat bizonyos esetekben le lehet rövidíteni, addig az XHTML-ben mindig meg kell adni a teljes attribútumot.

Az XHTML szabályait HTML-ben nem kötelező betartani, de ha mégis így teszünk, akkor sem lesz semmi probléma, egy eset kivételével; ha üres elemet törtvonallal zárunk le, mivel ez HTML-ben hiba, pedig az XHTML-ben ez helyes.[3]

4. CSS (Cascading Style Sheets)

4.1. Mi az a CSS?

A CSS (Cascading Style Sheets, magyarul: egymásba ágyazott stíluslapok) a World Wide Web Consortium (W3C) egyesületnek a weblapok megjelenítésére vonatkozó szabványa, a HTML nyelv kibővítését szolgálja. Segítségével oldalainknak professzionális megjelenést adhatunk mindezt úgy, hogy egy stíluslapot rendelünk hozzájuk, így egy esetleges stílusbeli módosítást egyszerűen végrehajthatunk az összes szükséges helyen anélkül, hogy hosszú órákat töltenénk a forráskódok kézi szerkesztésével. Így hát nemcsak oldalainkat szépítjük meg, de a saját munkánkat is jelentősen megkönnyítjük. [11]

Meghatározhatjuk, hogyan és hol jelenjenek meg az egyes HTML elemek, befolyásolhatjuk méretüket, elhelyezkedésüket, színüket, stb. Az egymásba ágyazhatóság arra utal, hogy több stíluslapot is megadhatunk egyszerre, illetve egy stílus lehet több elemre is érvényes, amit egy másik stílussal felüldefiniálhatunk. A stílusok öröklődnek az oldal hierarchiája szerint: ha például egy gyökér elemre definiálunk egy stílust, akkor az többnyire az oldal összes elemére érvényes a tulajdonságok örökölhetőségétől függően.

4.2. A CSS története

A fejlesztők a HTML nyelvet a kezdetekben a tartalom egyszerű közlését megoldó leírónyelvnek tervezték. Megjelenése egyre több ember számára tette lehetővé, hogy publikálhasson az interneten, aminek terjedése több, és több új felhasználót hozott. Ezek egyre többre vágytak, így megnőtt az igény az egyre bonyolultabb, kiforrottabb elemekre és formázásokra.

Ennek következtében a HTML kiegészült bizonyos formázási lehetőségekkel, amik ugyan sok mindent lehetővé tettek, ám a nyelv elvesztette egyszerűségét. Az oldalak forrásának akár több mint a fele már nem is a tartalomról szólt, hanem a megjelenésről, ezzel megnöttek a dokumentumok méretei is. [11]

Ezen segített a CSS szabvány, melyet a böngészők egyre egységesebben értelmeznek, s sokkal szabadabban és rugalmasabban tudjuk vele befolyásolni oldalaink megjelenését, mint valaha.

A technológia már elég régóta létezik, maga a szabvány leírása 1996 decemberében látott napvilágot a World Wide Web Consortium honlapján. Kiadása óta több verzió is létrejött, 1998. május 12-n jelent meg a CSS 2, a CSS 2.1 és CSS 3 kidolgozása pedig folyamatban van.

4.3. Miért használjunk CSS-t?

Egy rövid példán keresztül szeretném bemutatni, miért olyan előnyös számunkra a CSS használata és miként könnyíti meg munkánkat. Készítsünk egy egyszerű, pár soros oldalt, ami több címsort és bekezdést tartalmaz. A címsorok betűi legyenek nagyok, színük pedig kék, míg a bekezdések betűmérete legyen kisebb, piros színnel. Előbb HTML formázással oldjuk meg a feladatot, majd utána CSS használatával.

Feladat megoldása HTML formázással:

```
<html>
<head>
  <title>Feladat megoldása HTML formázással</title>
</head>
<body>
<h1><font size="4" color="#0000FF">Címsor 1</font></h1>
<p><font size="2" color="#FF0000">Bekezdés 1</font></p>
<h1><font size="4" color="#0000FF">Címsor 2</font></h1>
<p><font size="2" color="#FF0000">Bekezdés 2</font></p>
<h1><font size="4" color="#0000FF">Címsor 3</font></h1>
<p><font size="2" color="#FF0000">Bekezdés 3</font></p>
</body>
</html>
```

Láthatjuk, hogy a fenti megoldásban minden egyes címsor és bekezdés esetében egyenként meg kellett adnunk a betűk méretét és színét. A betűk méreténél csak az előre

definiált, 1-től 7-ig meghatározott méretek közül tudunk választani, és a forrás szinte teljesen áttekinthetetlen. Pedig most csak egy egyszerű, pár soros oldalról van szó...

Feladat megoldása CSS-sel

```
<html>
<head>
  <title>Feladat megoldása CSS-sel</title>
  <style type="text/css">
    h1 {font-size: 20px; color: #0000FF; }
    p {font-size: 12px; color: red; }
  </style>
</head>
<body>
<h1>Címsor 1</h1>
<p>Bekezdés 1</p>
<h1>Címsor 2</h1>
<p>Bekezdés 2</p>
<h1>Címsor 3</h1>
<p>Bekezdés 3</p>
</body>
</html>
```

Ennek a megoldásnak a kulcsa a style elemek között van, ezeken belül határozzuk meg, hogy az oldalon szereplő összes címsor és bekezdés milyen kinézetet kapjon. Ezt csak egyszer kellett megtennünk, ezáltal a forrás is sokkal áttekinthetőbbé vált. A betűk méretét immár pixelekből is megadhatjuk, a színekre pedig nemcsak a hexadecimális alakjukban hivatkozhatunk, hanem az angol nevüknek megfelelően. Ha pedig valamikor a jövőben módosítani szeretnénk egy elem megjelenését, elég azt egy helyen megtennünk, nem kell végigpásztáznunk az egész forráson. Nem elhanyagolható szempont az sem, hogy a dokumentumunk mérete is csökkent, ami nemcsak kisebb adatforgalmat, de gyorsabb betöltődését is von maga után.

4.4. CSS hozzákapcsolása HTML-hez, vagy XHTML-hez

A fenti példa csupán egy lehetőséget mutatott be a stíluslapok használatára, ám ezen kívül több megoldás is létezik, amelyeket akár keverve is használni tudunk.

4.4.1. Szöveggözi stílusok

Az egy-egy elemre vonatkozó stílusinformációk a style jellemzővel adhatók meg. E jellemző értékét egy kapcsos zárójelek nélküli meghatározásblokkban adhatjuk meg:

```
<p style="font-size: 12px; color: red;">Ez egy bekezdés</p>
```

Nagyon hasonlít ahhoz, mintha font elemekkel határoznánk meg a megjelenést, de észre kell vegyünk, hogy sokkal több formázási lehetőség közül választhatunk. Általában nem tanácsos ez a megoldás, mert bár a jellegzetes XML dokumentumnyelvek (XHTML 1.0, XHTML 1.1) ismerik a style jellemzőt, elég valószínűtlen, hogy minden ilyen nyelvben használható lenne. Amúgy is kevés alkalommal lenne rá szükség, de akkor jól jön, ha az egyes elemekhez helyben akarunk stílust meghatározni. [11]

4.4.2. Beágyazott stíluslapok

Stíluslapot beágyazhatunk a HTML, vagy XHTML lap elejére is a style elem segítségével. Ezt az elemet a head elemen belül kell elhelyezni, ahogy azt a fenti példában is tettük.

```
<html>
<head>
  <title>Feladat megoldása CSS-sel</title>
  <style type="text/css">
    h1 {font-size: 20px; color: #0000FF; }
    p {font-size: 12px; color: red; }
  </style>
</head>
...
</html>
```

Az így meghatározott stílus az egész dokumentumra érvényes lesz.

4.4.3. Külső stíluslapok

A stílusokat külön fájlban is meg lehet adni, aminek a legfőbb előnye az, hogy ha egy fájlba gyűjtjük az általánosan használt stílusokat, azt ezt a stíluslapot használó összes fájlt egyetlen stíluslap megváltoztatásával lehet módosítani. Továbbá a külső stíluslapokat a számítógép a gyorsítótárba menti, ami csökkenti az adatforgalmat.

A használandó stíluslapot három különböző módszerrel lehet hozzákapcsolni a HTML, vagy XHTML dokumentumhoz. [11]

4.4.3.1. Az @import utasítás

A stíluslap elején elhelyezhető egy vagy több ilyen utasítás. A HTML, vagy XHTML dokumentumok esetén ezt egy beágyazott stíluslap belsejében lehet megoldani.

```
<head>
<title>Példa az @import utasítás használatára</title>
<style type="text/css">
@import url(fejlec.css);
@import url(lablec.css);
p {font-size: 12px; color: red; }
</style>
</head>
```

Egy stíluslapba be lehet ágyazni egy másikat, ami akár egy harmadikat is betölthet. A külső stílus hivatkozásának meg kell előznie bármilyen más definíciót, amennyiben már szerepel előtte más, figyelmen kívül lesz hagyva. [11]

4.4.3.2. A link elem

Ennek az elemnek a segítségével is társíthatunk stíluslapot egy dokumentumhoz, valamint több ilyen elem használata is engedélyezett. A media jellemző használatával a stíluslap egy, vagy akár több megjelenítő közegre korlátozható (all = minden, screen = képernyős megjelenítés, print = nyomtatás)

```

<head>
<title>Példa a link elem használatára</title>
<link rel="stylesheet" type="text/css" href="fejlec.css" media="screen">
<link rel="stylesheet" type="text/css" href="lablec.css" media="screen">
<link rel="stylesheet" type="text/css" href="papir.css" media="print">
</head>

```

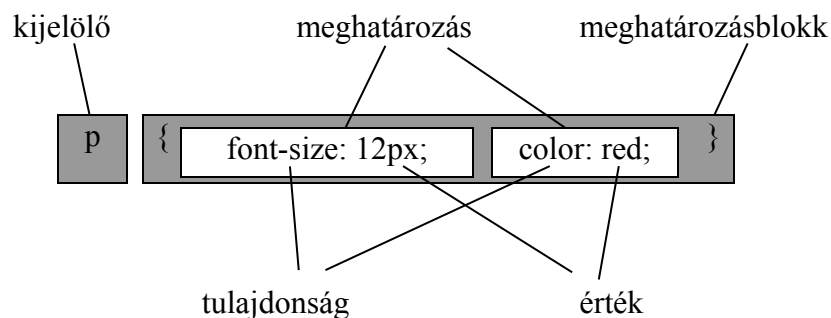
4.4.3.3. Az xml-stylesheet feldolgozási utasítás

Az XML dokumentumokban az xml-stylesheet feldolgozási utasítás segítségével lehet a stíluslapot a dokumentumhoz társítani. Minden xml-stylesheet feldolgozási utasítást az XML dokumentum bevezetése előtt kell elhelyezni, és több ilyen utasítás is használható.

```
<?xml-stylesheet type="text/css" href="stiluslap.css" ?>
```

4.5. A stílusok felépítése

Egy stíluslap egy vagy több stílust tartalmaz, amelyek azt írják le, hogy kell az elemeknek megjelenniük a lapon. Minden stílus két alapvető összetevőből áll: a kijelölőből és a meghatározásblokkból.



A stílus bal oldalán található a kijelölő, amely azt adja meg, hogy a dokumentum melyik részére vonatkozik az illető stílus. A stílus bal oldalán pedig a meghatározásblokk áll, ami mindig kapcsos zárójelek között található. Egy meghatározásblokkot egy vagy több meghatározás alkot, amiket pontosvesszővel kell elválasztani egymástól. Minden ilyen meghatározás egy CSS-tulajdonságból és az ehhez a tulajdonsághoz tartozó értékből épül fel.

Egy tulajdonság egy adott stílusjellemzőt képvisel, kettősponttal kell elválasztani a hozzá tartozó értéktől. [3]

4.6. Stílusosztályok

Egy elemhez készíthetünk több különböző, ún. stílusosztályt is: az elem neve után egy ponttal elválasztva adjuk meg az osztály nevét, pl. `code.html {color: yellow}`, `code.js {color: blue}`. Az osztályneveket meg lehet adni konkrét tag nélkül is, pl. `.js {color: blue}`. Ez esetben a `.js` nevű osztályt bármelyik elemmel használhatjuk. Az osztályoknak pl. a funkciójuk szerint adhatunk nevet. Az osztályokra az elemekben a `class` attribútummal lehet hivatkozni, pl. `<p class=code.html>Sárgával írt kód</p>`. Egyszerre csak egy módosítót használhatunk kiválasztó elemenként, alosztályok megadása (pl. `code.html.tulajdonság`) nem lehetséges. [11]

Léteznek ún. ID-kiválasztók is: ezek megadásánál a kettőskereszt (`#`) karaktert használjuk, és elnevezésük arra utal, hogy a HTML elemekben az `id` attribútummal lehet rájuk hivatkozni. Pl. `#ind30 {text-indent: 30}` stílusmegadásra hivatkozunk: `<p id=ind10>Ez egy 10 pixellel beljebb kezdett bekezdés</p>`

4.7. Pszeudo-osztályok és pszeudo-elemek

A CSS-t támogató böngészők automatikusan felismerik az ún. pszeudo-osztályokat és pszeudo-elemeket is. A pszeudo-osztályok megkülönböztetik az elemek típusait (pl. a linkeket és a látogatott linkeket, mint hivatkozástípusokat). A pszeudo-elemekkel az elemek által meghatározott elemek bizonyos részeire hivatkozhatunk, pl. egy bekezdés első betűjére.

Használatuk:

kiválasztó: pszeudo-osztály {tulajdonságnév: érték}

kiválasztó: pszeudo-elem {tulajdonságnév: érték}

A pszeudo-osztályok és pszeudo-elemek nem adhatók meg a `class` attribútummal, de a hagyományos stílusosztályokkal igen, pl.

Kiválasztó.osztály: pszeudo-osztály {tulajdonságnév: érték}

Kiválasztó.osztály: pszeudo-elem {tulajdonságnév: érték}

Hivatkozások pszeudo-osztályai

Az <a> elemhez használhatjuk a link, visited és active pszeudo-osztályokat, pl.

```
a: link { color: red }
a: active { color: blue; font-size: 125% }
a: visited { color: green; font-size: 85% }
```

Ezzel a hivatkozások, az aktuális és a már bejárt hivatkozások megjelenését változtathatjuk meg.

Első sor pszeudo-elem

Az első sor (first-line) pszeudo-elem segítségével a szövegblokkok első sorának adhatunk a többi sortól eltérő kinézetet, pl.

```
p: first-line {
    font-variant: small-caps;
    font-weight: bold
}
```

Első betű pszeudo-elem

Az első betű (first-letter) pszeudo-elemmel lehet iniciálékat készíteni, pl.

```
p: first-letter {
    font-size: 300%;
    float: left
}
```

4.8. Öröklődési szabályok

A stílusok öröklődhetnek, és az elemen belüli stílusdefiniókkal felülírhatjuk a globális stílusmegadásokat. Ez azt is jelenti, hogyha egy elemhez tartozik globális és lokális stílusmegadás is egy adott tulajdonságra, akkor a lokálisan megadott érték felülírja a globálisan megadottat az adott helyen. (Általában a később megadott, tehát a dokumentumban

lejjebb található stílusmegadás érvényesül.) De ha a stílusmegadás végén az !important kulcsszót is elhelyezzük, biztos, hogy az ott megadott stílus fog megjelenni, nem lehet lokálisan felüldefiniálni azt. Használata:

```
body {  
    background: url(bar.gif) white;  
    background-repeat: repeat-x !important  
}
```

A stíluslapok megjelenését a szerzők és az olvasók is módosíthatják. A szerzők beállításai mindig felülírják az olvasók beállításait (hiszen tőlük ered a dokumentum), az !important kulcsszóval jelölt beállítások pedig magasabb rendűek a normál beállításoknál. Néhány látogatónak szüksége lehet, hogy a szövegek nagyobb betűmérettel és markáns színekkel jelenjenek meg, de nem valószínű, hogy az !important használatát is ismerik. Ezért nem célszerű minden esetben az !important erőszakos alkalmazása. [11]

5. A JavaScript

5.1. A JavaScript története, a nyelv főbb jellemvonásai

A webes scriptnyelvek olyan parancsnyelvek, melyeket a böngésző futtat. A JavaScript scriptnyelv a webért született, amelyet még a fejlesztése első szakaszában LiveScriptnek neveztek, de a későbbiekben üzletpolitikai okokból JavaScriptre kereszteltek át.

A nyelvet eredetileg csak néhány böngésző támogatta. Kezdetben, 1995 végén, csak a Netscape böngésző 2.0-s verziója támogatta. Magát a nyelvet Brendan Eich írta. Eredeti neve a LiveScript volt, amely a Netscape LiveWire nevű projekt nevéből eredt. A JavaScript nyelvnek semmi köze sincs a Java programozási nyelvhez. A kezdetekben alakult ki ez az elnevezés, amikor a Sun és a Netscape közösen bejelentették a nyelvet. A JavaScript-et a Java programozási nyelv kiegészítőjének szánták, gyors programozási feladatok megoldásához a Netscape böngészőjében. A két nyelv fejlesztése azóta teljesen elvált egymástól, annyi közös van bennük, hogy mindkét nyelv szintaktikája a C nyelv szintaktikáján alapszik.[8].

A nyelv egyre nagyobb támogatottságot élvezett és látványos átalakuláson ment keresztül, a kezdeti, objektum-alapú nyelvből fokozatosan közeledett az objektum-orientált alapú nyelvek irányába.

Az alábbi táblázat összefoglalja röviden a JavaScript nyelv történetének főbb mérföldköveit:

Év	Esemény
1995	- A Netscape 2.0-ban alkalmazzák először a JavaScript 1.0-ás verzióját.
1996	- A JavaScript 1.0-ás verzió az Internet Explorer 3.0-ban is támogatott lesz, a Microsoft JScriptnek 1.0-nak nevezi. - A JavaScript 1.1-es verziót a Netscape 3.0-ás verziójában használják, az Internet Explorer 3.0-ás verziónál magasabbak támogatják, a Microsoft ezt JScript 2.0-nak nevezi el.
1997	- Megjelenik a JavaScript 1.2-es verziója a Netscape 4.0-ban. - Ezt a verziót a Microsoft elnevezi JScript 3.0-nak, valamint az Internet Explorer 4.0 támogatja.
1998	A Netscape 4.06-os verziójától van jelen a JavaScript 1.3.
1999	- A JavaScript 1.3-as beépül az Internet Explorer 5.0-ás verziójába, ezt a

	Microsoft JScript 5.0-nak nevezi el. - Az Internet Explorer 5.01-es verziójába JScript 5.1-et interpretáltak.
2000	- A Netscape 6.0-ás verziójában megjelenik a JavaScript 1.5. - Az Internet Explorer 5.5-ös verzióba a JScript 5.5-öt interpretáltak.
2001	- Az Internet Explorer 6.0-ás verziójába a JScript 5.6-ot interpretáltak.
2005	- A Firefox 1.5-ös verziójában megjelent a JavaScript 1.6
2006	- A Firefox 2.0-ás verziójában megjelent a JavaScript 1.7

A JavaScript nyelv főbb jellemvonásai a következők:

- az interpretálás, amely azt jelenti, hogy a böngésző fogja értelmezni, majd lépésenként végrehajtani a forrássorokat.
- biztonság, ez abban nyilvánul meg, hogy a JavaScriptben nincs olyan függvény vagy utasítás, melyek által elérhetővé válhatnának a kliens fájljai a kiszolgáló számára.
- hordozhatóság, a legtöbb manapság használt web-böngésző támogatja a JavaScriptet.
- interaktivitás, jól vezérelhető kódból az input mezők és egérműveletek lekezelése.
- maga a nyelv kis- és nagybetű érzékeny (case sensitive).
- a JavaScript objektum-alapú nyelv, nem rendelkezik lezármazott objektumosztályokkal, mivel minden osztály egy szinten van; viszont az objektumok között van alá- és fölrendeltségi viszony. Egy objektum részei az elemváltozók és az elemfüggvények.
- A JavaScript nyelv gyengén típusos, nem szükséges megadni egy változó típusát. Abban az esetben, ha típuskonverzióra van szükség, a JavaScript elvégzi automatikusan.[4]

5.2. JavaScript előnyei, hátrányai

A JavaScript nyelv használható arra, hogy olyan interaktív script fájlokat írjunk, amelyek különféle eseményekre reagálnak, ilyen esemény lehet az egérekattintás, illetve a billentyűnyomás.

JavaScript dinamikus funkciókat biztosíthat egy weblaphoz. Segítségével olyan kis programkódokat lehet írni egy oldalhoz, amelyek a felhasználó számítógépén fognak lefutni,

és nincs szükség semmilyen speciális szoftver telepítésére a kiszolgálón. A JavaScript lehetőséget ad arra, hogy néhány egyszerű funkcionalitás megvalósulhasson a weblapon, ezáltal azt interaktívvá tehesse, de vannak korlátai, amelyek a szerveroldali programozási nyelvekhez és a dinamikus weboldalakhoz vezetnek tovább.

A JavaScript legnagyobb előnye, hogy lehetővé teszi az alkalmazások közvetlen beépítését a HTML kódba. A JavaScript weblapon való elhelyezésére két lehetőség szolgál. Az egyik a kliens oldali, míg a másik a szerver oldali elhelyezés.

A kliens oldali JavaScript alkalmazások azok a scriptek, amelyek közvetlenül a HTML oldalakba vannak beágyazva. Ezeknek a végrehajtására akkor kerül sor, amikor felhasználó web-böngészője betölti az adott oldalt. Ennek a lehetőségnek az előnye az, hogy a script észleli és alkalmazza a felhasználó által előidézett eseményeket, és ezeket az információkat hasznosítani tudja majd a script más részeinek meghívására, elkerülve a visszatérést a webkiszolgálóra.

A szerver oldali elhelyezés esetén a JavaScript alkalmazások a web-szerveren helyezkednek el. Minden weblapnak, amely használni akarja ezeket a scripteket, külön meg kell hívnia őket.

Nagy előnyei ennek a módszernek a következők: egyrészt a szerver oldalon lévő scriptek alkalmazása esetén könnyebb hozzáférni a további, szerveren futó alkalmazásokhoz, másrészt nem terheli meg annyira a felhasználó gépét, mivel a szerveren fognak futni a scriptek.

A két módszer közül a kliens oldali az elterjedtebb, mivel a szerver oldali alkalmazások programozására a JavaScripttől hatékonyabb nyelvek állnak rendelkezésre.

Egy másik előnye a JavaScriptnek az, hogy nincs szükség plusz eszközökre JavaScriptek írásához, mivel bármely szövegszerkesztő képes kezelni, így nem kell külön fejlesztőkörnyezetet telepíteni ahhoz, hogy JavaScript kódot létrehozzunk.

A JavaScript viszont rendelkezik néhány negatív tulajdonsággal is. A JavaScript kódja végrehajtódik a felhasználó számítógépén, így rosszindulatú célokra is lehet használni. Emellett egyes böngészők másképpen értelmezik a JavaScriptet.[4]

6. Flash

6.1. A Flash rövid története

A Flash kialakulása 1996-ba tehető. A Macromedia cég FutureSplash nevű vektorgrafikus alkalmazása nagyon népszerűvé vált a honlapszerkesztők körében, mivel ezen szoftver segítségével olyan vektorgrafikus animációkat tudtak készíteni, amelyeknek nem volt túl nagy a méretük, ezáltal jól lehetett őket alkalmazni a honlapokon.

A FutureSplash tervezői, fejlesztői továbbfejlesztették ezt az alkalmazást, mivel a felhasználók egyre szélesebb körben igényelték az interaktív menürendszerek használatát, a FutureSplash azonban kifejezetten lineáris animációk létrehozására jött létre.

A Flash 5 megjelenése mérföldkő az animáció területén, hiszen a segítségével olyan interaktív filmeket tudtak készíteni, amelyek közvetlenül kapcsolatba tudtak lépni egymással.

Hang, kép, illetve videóanyagokat is képes volt beleépíteni saját formátumaiba, jelentős minőségvesztés nélkül. Legalapvetőbb tulajdonsága a fájl mérete, amely meglepően kicsi.

A Flash termékeit bármely népszerű web-böngészővel, illetve operációs rendszer alatt meg lehet jeleníteni.

A Flash technológia azért mérföldkő a webes fejlesztések területén, mert nagyon hamar elterjedt, szinte mindegyik operációs rendszer támogatja, és maga a Flash lejátszó szinte 100 százalékban elterjedt a világon.[1]

2005-ben a Macromedia és az Adobe összeolvadt, ennek legfőbb oka a Microsofttól való félelmük volt, ezáltal a Flasht a későbbiekben már Adobe Flashként ismerhetjük.

6.2. A Flash előnyei, hátrányai

Ezen technológia legalapvetőbb célterülete az Internet. Egyre több cég, illetve magánszemély dönt úgy, hogy honlapját Flash animációkkal fogja teletűzdelni.

Ennek igen egyszerű oka van, a Flash egy olyan lehetőség, amellyel multimédiás tartalmakat lehet készíteni, ezért alkalmazásával a weboldalunk változatosabb, igényesebb valamint dinamikusabb lesz.[1]

A mai weblapok többsége alkalmaz Flasht. Vannak szinte csak Flash animációt tartalmazó weblapok („full flash” weboldalak), főleg azok, amelyek valamilyen terméket mutatnak be. Emellett vannak Flash mozik, illetve önálló Flash alkalmazások (játékok).

A teljesen flash weboldalak készítése viszont kockázatos lehet, főleg a kis cégek számára, ennek oka pedig a következő: ahhoz, hogy egy honlap jól szerepeljen a keresőkben, bizonyos optimalizálási és marketing módszereket érdemes elvégezni. A teljes flash weblapok alkalmazása azért veszélyes a kis cégek esetén, kis- és középvállalkozói szektor számára, mert egy kis cégnek az Interneten megjelenni szinte kizárólag értékesítési szándékkal érdemes.

Ilyen teljesen flash alapú weblap a ”Senisoft” cég honlapja is.



6.1. ábra – Flash weblap

A flash alapú honlapok esetében azonban pont a lényeg vész el, mert a keresők, mivel nem „látják”, ezért képtelenek az adott oldalt és tartalmát beindexelni, ezért rossz esetben a weboldal meg sem jelenik a keresőkben.[9]

Ezen kívül további hátrányai is lehetnek a ”full flash” weblapoknak:

- nem minden felhasználó fogja tudni megtekinteni a weblapot, vagy ezek nem abban a formában fognak megjelenni a felhasználók számára, mint ahogy a honlapkészítő eltervezte;
- a flash honlapok költségei meghaladják a hagyományos HTML oldal költségeit;
- a különféle böngészők miatt nem fognak egységesen megjelenni ezek az oldalak;
- ha a felhasználó kis sávszélességű internet kapcsolattal rendelkezik, akkor lassabban fog betöltődni neki egy ilyen oldal, mint egy hagyományos HTML alapú oldal.
- más eszközökön (pl. mobiltelefon, PDA) a legtöbb esetben nem jeleníthetők meg

A Flash mozik alkalmasak céges bemutatókon, valamilyen termék gyártási folyamatának bemutatására.

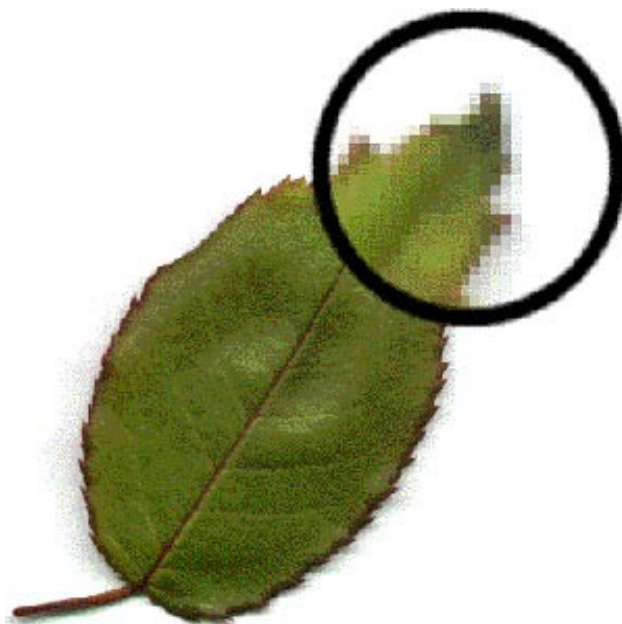
A mai reklámok nagy része Flash animációk, ennek oka igen egyszerű, a felhasználóknak a figyelmét jobban odavonzza, mint egy egyszerű szöveges hirdetés. A Flash animációk, mozik megfelelő és ötletes alkalmazása emeli weblapunk fényét, így a látogató legközelebb is visszatér. A Flash, mint program egyesíti magában a vektorgrafikus szerkesztő, az animáció készítő, valamint a multimédia-fejlesztő tulajdonságait.

Mint már említettem a Flash, mint program egyszerre három dolgot egyesít magában.

Az egyik legfontosabb a három közül a vektorgrafikus szerkesztő mivolta. Ahhoz, hogy megértsük mi is az a vektorgrafika, először is tisztázni kellene a bittérképes grafika fogalmát.

6.2.1. A bittérképes grafika (pixel grafika)

Ez a módszer a képek megjelenítésének legegyszerűbb módja. A képet függőleges és vízszintes irányban pontokra, pixelekre bontja fel, és minden pixelről eltárolja a szín és fényerősségi információit. Ennek az a hátránya, hogy sok helyet foglalnak el ezek az információk. Másik hátránya ennek a módszernek az, hogy általában torzulás történik a képen, ha valamilyen műveletet akarunk vele elvégezni. A leggyakrabban használt fájlformátumok a következők: bmp, jpeg, gif.[10]



6.2. ábra – Példa a bittérképes grafikára

6.2.2. A vektorgrafika

A vektorgrafika ezzel szemben más módszeren alapszik. A vektorgrafika vagy geometriai modellezés a számítógépes grafikában az az eljárás, melynek során geometriai primitíveket (rajzelemeket), mint például pontokat, egyeneseket, görbéket és sokszögeket használunk képek leírására. [10]

Ha egy rajzprogrammal egy olyan rajzot hozunk létre, amely vektorgrafikával készült, a rajzprogram egy nem látható hálóra rajzolja ki a képet, majd ezt utasítások halmazaként tárolja el a program egy állományban. Ezek az utasítások azt írják le, hogy milyen az alakzat színe, mérete, pozíciója, valamint további információkat tárolnak el. Ha meg szeretnénk jeleníteni ezt a grafikát, a program végrehajtja ezeket az utasításokat, és ezen utasításokból fog előállni a rajz maga.

A vektorgrafika nagy előnye a bittérképes grafikával szemben, hogy nem a képeket alkotó pontokat tárolja, hanem a megjelenítéshez szükséges információkat, ezáltal a fájlmérete jóval kisebb lesz; valamint kevésbé torzul a kép, ha különböző műveleteket végzünk rajta.

Azonban ez a módszer hátránnyal is rendelkezik, a nagyon sok utasításból előálló rajzokat több idő alatt jeleníti meg, ezáltal a fénykép minőségű képek létrehozására nem alkalmas. [10]



6.3. ábra – Példa a vektorgrafikára

A Flash másik nagy előnye, hogy animáció készítő alkalmazás is egyben. Az animációkról a „Felhasználóbarát webdesign” című fejezetben ejtek szót.

7. Felhasználóbarát webdesign

Ez a fejezet arra hivatott, hogy némi leírást szolgáltatassunk a felhasználóbarát megközelítésről.

7.1. Olvashatóság

Először is nagyon fontos a weblapok vonatkozásában az olvashatóság. Ez nagyon fontos tényező, mivel hiába lehet egy oldal nagyon gyors, hiába rendelkezik igényes megjelenéssel, és lehet modern, mit sem ér, ha a felhasználó nem tudja elolvasni az információt róla.

Érdeemes a szövegeket balra rendezni, ugyanis az olvasó könnyebben tudja átfutni a szemével az ilyen szövegeket, mint a középre, vagy a jobbra igazítottakat.

Ha lehet, akkor használjunk a nagy betűket, de jogi szövegekre, illetve lábjegyzékekre nem kell akkora hangsúlyt fektetni, mivel ezeket a szövegrészeket kevesen olvassák el, nem bírnak nagy jelentőséggel. Kerülni kell a villogó, illetve mozgó szövegrészeket, mivel ezek nagyon fárasztják a szemet, jobb a mozdulatlan szövegek használata. Az előbbieket inkább a figyelemfelkeltés eszközeként érdemes alkalmazni.

Az is növeli az olvashatóságot, ha ellentétes színeket alkalmazunk a szöveg, illetve a háttér vonatkozásában. A legjobb választás a fehér alapon fekete betűk használata, ezzel majdnem megegyezik a fekete alapon fehér betűk alkalmazása.

Ha a szöveg világosabb a feketénél, illetve a háttér sötétebb a fehérenél ez lényegesen rontja az olvashatóságot. A legrosszabb eset az olvashatóság szempontjából az olyan színek használata, amelyek elütnek egymástól, tehát kerüljük a komplementer színek egy időben való használatát. Erre kifejezetten igaz a vörös és a zöld színek alkalmazása, mivel ezen színek nem csak rontják az olvashatóságot, hanem vannak olyan felhasználók is, akik keverik őket. (piros-zöld színtévesztők)

A hátterek vonatkozásában érdemes egyszínű, vagy nagyon finom mintázatú hátteret alkalmazni.[2]

7.2. Színek megfelelő használata

A színek nagyon fontos szerepet töltenek be a webdesignban, segítségükkel nagyon igényes oldalakat lehet létrehozni, viszont a nem megfelelő használatuk teljesen tönkretetheti az oldal megjelenítését. Nagyon jók arra, hogy a fontos gondolatokat kiemelhessük egy szövegben. Figyelembe kell venni, hogy bizonyos színeket nem szabad egy időben használni (komplementer színek), illetve azt is, hogy nem minden szín webtűrő. Ezek olyan színek, melyek ugyanúgy jelennek meg minden felületen, böngészőprogramban.

Figyelembe kell venni azt, hogy különböző megjelenítési eszközökön a színek másképp hatnak, például egy szín másképpen néz ki egy hagyományos CRT képernyőn, mint egy LCD monitoron, vagy egy kivetítőn.

Oldalainkat nem szabad túlszűfölni a színekkel, mert az zavaró hatású lesz. Célszerű öt színt használni, még hozzá úgy, hogy mindegyiknek legyen meg a saját funkciója, jelentése. Egy másik nagyon fontos szempont a színekkel kapcsolatban az, hogy mindig figyelembe kell venni az adott ország kultúráját, ahová az oldalunkat tervezzük. Más-más kultúrában a színek különböző jelentéssel bírnak, kifejezetten igaz ez a keleti kultúrákra, ezáltal könnyen elfordulhat az is, hogy ami az egyik országban elfogadott, máshol felháborító lesz.

A színekkel hangulatot közvetítünk, ezért ügyelni kell arra, hogy a tartalomhoz megfelelő színekombinációt használjunk.

Egyes színek jelentése, illetve megfelelő alkalmazása:[1]

Piros

A piros szín előtérbe hozza a képeket, valamint a szövegeket. Javasolt kiemelő színként használni, mivel az embereket gyors döntéshozatalra készíti.

Megfelelő a „Buy Now”, vagy a „Click Now” típusú feliratok használatára, mivel nagyon szembetűnővé teszi őket.

A piros szín széles körben alkalmazott szín a veszély jelzésére, illetve az energikus termékek reklámozására, mint például a gyors autók, energiatartalok, energiaszeletek reklámozására.

Sötétebb árnyalatai a bátorságot, erőt, haragot sugallják, míg világosabb árnyalatai a szenvedélyt, szerelmet jelképezik.

Zöld

A zöld a természet színe, nagyon nyugtató jellege van. Leginkább a gyógyszerek és orvosi termékek reklámozására használják, mivel biztonságot sugalló szín, de a környezetvédő oldalaknál is előszeretettel alkalmazzák, valamint az újrahasznosítható termékeknél. A matt sötétzöld leginkább az üzleti világra utal, pénz, bankok esetén fordul elő gyakran.

Narancssárga

A narancssárga egy nagyon meleg szín, alkalmas figyelemfelkeltésre, valamint a legfontosabb elemek kiemelésére. A fiatalok körében nagyon népszerű. Hatékony a különféle ételek, játékok reklámozására.

Kék

A kék szín nagyon univerzális szín. Rengeteg helyen használhatjuk, mint például a csúcstechnológiájú termékek reklámozására, mivel magas pontosságot és tudást sugall, de a tisztaság érzetét is kelti, ezáltal használhatjuk az ilyen jellegű termékek reklámozására (mosószeres, izzadás gátlók). Bármilyen vízi és légi témájú cikkhez, hírhez, reklámhoz megfelelően használható szín. (tengerek, légi utazások, ásványvíz, stb.)

Sárga

Elégé figyelemfelkeltő hatással bír, de a túlzásbavitele zavaróvá válhat a felhasználó számára.

Ha fekete háttérrel együtt alkalmazzuk a sárga színt, akkor válik a legfeltűnőbbé, ezért szokták a figyelmeztetéseket ilyenféleképpen megadni. Ha stabilitást, megbízhatóságot akarunk sugallni, nem célszerű ennek a színnek a használata, mivel elégé megbízhatatlan jelentéssel bír. Az arany szín viszont a bölcsesség és a vagyon színe, nagyon magas minőségű, illetve értékű termékek reklámozására kiváló.

Bíbor

Nagyon jó szín a női designhoz, ezzel magyarázható az, hogy a legtöbb hölgyeknek szánt termék, hír, cikk erőteljesen erre a színre épül (női magazinok, szépségápolási termékek, stb.) A bíbor élénkebb árnyalata a gyermek termékek reklámozására megfelelő (babaruhák, gyermekjátékok)

Fehér

A fehér szín pozitív jelentéssel bír, a tökéletesség, a tisztaság színe. Hirdetések vonatkozásában utalhat a frissességre, illetve a csúcstechnológiás termékekre. Ételek vonatkozásában leginkább a könnyűséget jelenti, mint például a diétás, zsírszegény ételek, cukormentes üdítőitalok, stb. Kiváló szín az orvosi rendelők, kórházak, orvosok oldalainak is.

Fekete

A fekete az erő, elegancia, valamint a szomorú tartalmú hírek színe.(gyász) A keleti kultúrákban más jelentéssel bír mind a fehér, mind a fekete. Hirdetések tekintetében, ha tekintélyt, eleganciát, vagy luxust akarunk sugallni, akkor a fekete tökéletes erre.

7.3. Grafikus elemek

Kezdetben csak szöveges elemek voltak a weben, de a technológiák fejlődésének köszönhetően fokozatosan jelentek meg előbb a képek, aztán a videók, később pedig a különféle animációk, grafikus alkalmazások.

Az animáció lényegében nem más, mint a rajzolt képek megjelenítése úgy, hogy a mozgás érzetét keltik a felhasználók számára. Fontos jellemzője az fps (frame per second - oldal per másodperc), amelynek az értéke 14 és 18 között kell lennie ahhoz, hogy a folyamatos mozgás érzete kialakuljon. Általa az unalmas állóképet mozgóvá tehetjük, így növekszik a felhasználói élmény. Az animációk fontos szerepet töltenek be a hirdetésekben, reklámokban, üzleti versengésekben, ez köszönhető annak, hogy erős figyelemfelkeltő hatásuk van. Igen mutatós egy-egy hirdetésben például a növekvő-csökkenő grafikonok használata.

Egy jól elkészített grafikai elem jó lehet arra, hogy a mai magas igényeknek egy site megfeleljen. Az animációkat tartalmazó oldalak jobban vonzzák a szemet, hatékonyabban kiemelik a közlendő információt.

Az animációk különféle típusúak lehetnek, vegyük szemügyre most ezeket.

Ahhoz, hogy animációt tartalmazó elemet építhessünk be alkalmazásainkba, kétféle lehetőség közül választhatunk.

Az első az, hogy animációt szerkesztő programmal hozzuk létre az animációinkat, majd a kész animációkat lejátszathatjuk ezzel a programmal. Ezen szoftverek két fajta animációt támogatnak, az állandó előtérrel készülő animációt, valamint az állandó háttérrel készülő animációt.[1]

7.3.1. Állandó előtérrel készülő animáció

Ez lényegében a hagyományos, pörgethető lapokkal egyezik meg. A kép hátere lapról-lapra változik, míg az előtér végig változatlan marad. A lapok csak egyszerű rajzot tartalmaznak. Ha gyorsan pörögnek a lapok, a mozgás érzetét keltik.[1]

7.3.2. Állandó háttérrel készülő animáció

Ez lényegében az állandó előtérrel készülő animáció továbbfejlesztése. Arra jöttek rá, hogy egyszerűbb, ha a háttér marad változatlan, és az előtér változik, mivel sokkal egyszerűbb egy állandó háttérre folyamatosan rárajzolni, mint fordítva. Az előtérben létrejövő mozdulatokat kell kisebb részekre bontani és rárajzolni a háttérre. Az animáció gyorsabban és könnyebben jön létre, mivel a háttér állandó, valamint az animáció a fájlmérete is kisebb.

A második lehetőség arra, hogy animációt tartalmazó elemet építsünk be alkalmazásainkba az úgy nevezett objektum animációk használata. Ebben az esetben saját magunk fejlesztjük ki és építjük be az animációt tartalmazó elemeket, ellentétben az elsővel, ahol az elemek külön állományokban voltak elmentve. Az animációs elemeket csak az alkalmazásban tudjuk megjeleníteni.[1]

7.3.3 Az objektum animáció

Létrehozunk egy grafikus objektumot, amelyet a képernyőn fogunk mozgatni egy program segítségével. Az ilyen animáció által kevés befektetett munkával nagy felhasználói élményt válthatunk ki, mivel igen látványos, ezért nagyon népszerűvé vált ez a technika, és az egyik legalapvetőbbé vált.

Lehetőségünk van az adott objektum kicsinyítésére, nagyítására, forgatására, illetve arra is, hogy az objektummozgás közben alakot váltson.

Kevés helyet igényel, ami nagyon lényeges, mivel nem terheli meg nagyon a felhasználó számítógépét. A kevés helyigény köszönhető annak, hogy az animációknak nincsen szükségük külső támogatásra.[1]

8. Webdesign hibák

Ha azt szeretnénk, hogy oldalunk dizájnja minden szempontból sikeres legyen, és a látogató jó érzéssel távozzon a honlapról, sok fontos szempontot kell figyelembe vennünk és ügyelnünk kell néhány alapvető hiba elkerülésére. Főleg a kezdő és tapasztalatlan webmesterek követik el ezeket a hibákat, amelyek elkedvteleníthetik, elvadíthatják a látogatót, aki amúgy pár másodperc leforgása alatt szokott döntenie arról, hogy az adott oldalon marad-e. [12]

A továbbiakban szeretnénk egy csokorba gyűjteni ezeket a hibákat, felhívva a figyelmet arra, miként kerüljük el, vagy küszöböljük ki őket.

- **Az oldal témája:** A látogatónak pár másodpercen belül tudomást kell szereznie arról, hogy miről szól az oldal. A látogatót gyorsan rá kell venni arra, hogy érdemes az oldalunk böngészésével töltse az idejét. Ha ez nem történik meg, kikapcsolja és más után néz.
- **A három klikkes szabály:** A lényeg az lenne, hogy egy weboldal sem lehet olyan bonyolult, hogy a látogató három egérgattintásból ne tudja elérni a kívánt információt. Ez a szabály annak a felhasználói problémának a következménye, hogy egyes webhelyeken egyszerűen lehetetlen megtalálni az információkat. A főoldalon jelenjen meg minden olyan információ, amivel a látogatót el tudjuk oda juttatni, ahol neki is és nekünk is jó. Ha vannak kiemelt funkcióink, mint például a hírlevélre feliratkozás, online rendelés, szállásfoglalás, akkor azt tegyük a látogató orra elé, hadd lássa, és érje el akár 1-2 másodpercen belül. E szabály betartása hasznos lehet a jobb és átlátható navigáció kialakításában. Inkább legyen egy jól strukturált oldaltérkép, mint egy bonyolult, többszintű menürendszer, mert azok sosem működnek igazán jól.
- **Flash intró:** Nagy hiba, ha egy webhely funkció nélküli Flash intróval kezdődik, mivel a látogatók türelmetlenek, azonnal információra van szükségük. Majdnem hidegen hagyja őket, hogy milyen mutatós animáció fogadja őket. Rossz választás az is, ha azt a kérdést tesszük fel a látogatónak, hogy az oldalunk Flash, vagy HTML változatát szeretné-e látni? Ha mégis arra vetemednénk, hogy ilyesmibe fogunk, mindenképp lássuk el két funkcióval: az egyik az animáció átugrása legyen, vagyis a látogató egyből a főoldalra kerül, a másik pedig a hang kikapcsolása. A legnagyobb

hiba a Flash-sel kapcsolatban, ha az egész oldal ebben készül. Ez egy eszköz, nem pedig cél. Csak akkor nyúljunk hozzá, ha valóban szükségünk van rá. A keresőrobotok ráadásul egyáltalán, vagy legalábbis csak korlátozott mértékben tudnak olvasni a Flash fájlokban, így szinte lehetetlen feladattal szembesülünk a keresőoptimalizáláskor. [9]

- **Keltezetlen tartalom:** Ha nem látjuk el dátumokkal oldalunk tartalmát, cikkeit, akkor a látogatóknak fogalmuk sem lehet arról, hogy az adott információ aktuális-e, esetleg idejétmúlt. Ajánlatos dolog archiválni az elavult tartalmat, és ezt egy külön menüpontban elérhetővé tenni. A tartalomkezelő rendszerek használatával a legegyszerűbb elkerülni ezt a hibát, velük a dátumkezelés és archiválás is néhány perc alatt megoldható.
- **Böngésző inkompatibilitás:** Nem csupán egy böngészőre kell optimalizálni az oldalt, mivel lehet, hogy a többiben nem működik normálisan. Az Internet Explorer egyeduralma már régen véget ért, egyre többen használnak Firefox-ot, Operát és Chrome-ot.
- **A látogató böngészőablakának átméretezése:** Soha ne méretezzük át a látogató böngészőablakát. Az ilyen megoldás csak a spammal teli oldalakra jellemző és ezt a látogatók is tudják.
- **Új böngészőablakok megnyitása:** A webmesterek tévesen hiszik azt, hogy ezzel a megoldással az oldalukon tarthatják a látogatót, ám ez egy öngyilkos művelet. Ne árásszuk el a látogató képernyőjét, tehát ne úgy írjuk meg a hivatkozásokat, hogy azok egy új böngészőablakban töltődjenek be. Bizzuk ezt a látogatóra. A böngészőkbe sem véletlenül építették be a „Vissza” gombot. Ha a látogatónak új böngészőablakra van szüksége, biztos nyit majd egyet magának. Ugyanakkor kerüljük a pop-up ablakok használatát is.
- **Vízszintes görgetés:** A függőleges görgetéssel nincs semmi gond, valószínűleg azért, mert nagyon megszokott, ám vízszintes görgetősávokat mindenki gyűlöli. Különösen zavaró lehet egy olyan weboldal esetén, amely kisebb felbontásra, például 800x600 pixel méretűre van optimalizálva.
- **Keresőoptimalizálás:** Az egyik legnagyobb hiba, amit a webmesterek elkövetnek az, hogy a keresőkre történő optimalizálásra csak utólag gondolnak. Itt különösen problémát jelenthet a Flash tartalom, kiváltképp, ha az egész honlap Flash alapú. [15]

- **Túl sok reklám:** A weboldal nem egy gigantikus hirdetőfal! A lényeg, hogy a látogatók számukra értékes információkhoz jussanak, a sok reklámcsík viszont magára irányítja a figyelmet, így tulajdonképpen „elvész” a tartalom.
- **Regisztráció:** Csak akkor kötelezzük a látogatót arra, hogy regisztráljon a rendszerünkbe, ha az valóban szükséges. Ő azért böngészi az internetet, hogy információkhoz jusson, nem pedig, hogy azokat adjon ki. Ha mégis regisztrálna, semmi esetre se küldjünk neki a beleegyezése nélkül hírleveleket.
- **Törött linkek:** A már elmozdult, esetleg megszűnt tartalomra mutató linkek nem csak a látogatókat hergelik fel, de a keresőrobotoknál is rossz pontnak számítanak, amik hátrásorolják majd az oldalunkat a találati rangsorban
- **Szövegaláhúzás:** A normál szöveget ne húzzuk alá és ne színezzük fölöslegesen, csak ha arra jó okunk van. A látogatók azt hiszik, hogy egy hivatkozást takar, és könnyen összezavarodhatnak, ha rájönnek, hogy mégsem így van. Továbbá 20%-kal rontja az olvashatóságot, mivel levágja, néha eltünteti a betűk lelógó részét.
- **Szöveg alapú menüpontok:** Ez a megoldás sokkal jobb, mint ha képekre írjuk az adott menüpont szövegét. CSS-sel már nagyon szépen meg lehet oldani a szöveg alapú navigációt, és gyorsabb is, mintha minden menüpontnál meg kellene várni, míg betölt a kép. A design a tartalom jó elhelyezéséről szóljon, mintsem, hogy telepakolgassuk az oldalt képekkel.
- **Látogatott linkek:** Nagyon zavaró dolog bármelyik látogató számára, ha egy már meglátogatott linknek nem változik meg a színe. Ez a felhasználhatóság egyik legfontosabb szempontja. A hiba megelőzésével segítséget nyújtunk a látogatónak az oldalunkon való navigációban, mivel nem fog véletlenszerűen még egyszer ugyanarra a hivatkozásra kattintani, aminek tartalma esetleg nem érdekelné, vagy pont ellenkezőleg.
- **Betűméret és betűtípusok:** A szöveg legyen olvasható. Az is zavaró, ha túl kicsi, de az is, ha túl nagy a betűméret. Használjunk egyszerű betűtípusokat: Verdana, Georgia, Arial, Tahoma, Helvetica, Times New Roman. Ha egy általunk használt speciális betűtípus nincs telepítve a látogató számítógépén, akkor a böngészője azt a saját alapértelmezett betűtípusára cseréli le automatikusan. Ezáltal nem tudjuk elérni azt a hatást, amit terveztünk. Egyaránt az olvashatóság és felhasználhatóság szempontjából fontos betartatunk ezt a szabályt. [12]

9. Keresőoptimalizálás

9.1. Mi az a keresőoptimalizálás?

Keresőoptimalizálásnak (angolul: Search Engine Optimization, röviden: SEO), vagy keresőmarketingnek nevezzük azt a szerteágazó tevékenységet, melynek célja, hogy a webes keresők rátaláljanak az adott weboldalra és minél előrébb jelenítsék meg azt a találati listájukon. Azon alapul, hogy a szabadszavas keresők csupán közvetítők az internetet böngésző felhasználók és a weblapok üzemeltetői között. Úgy is megfogalmazhatjuk ezt a viszonyt, hogy a keresők ingyenes hirdetési felületet biztosítanak a honlapkészítők számára, cserébe pedig az utóbbiak által szolgáltatott információk választ adnak a felhasználó kérdéseire. A keresők legfontosabb érdeke, hogy azokat az oldalakat hirdessék, amelyek a legértékesebb, legaktuálisabb, legfelhasználhatóbb választ adják a kulcsszavakkal megadott kérdésekre, mert így nő a felhasználók elégedettsége, és ezzel párhuzamosan saját látogatottságuk is. [14]

Teljesen mindegy, hogy üzleti jellegű, vagy non-profit oldalt készít az ember, az egyik legfontosabb cél, hogy minél többen találjanak rá. Azok a webhelyek szerzik meg a legtöbb látogatót, amelyek oldalai jókor és jó helyen jelennek meg. Ha egy webhely oldalai gyakran keresett kulcsszavakat tartalmaznak, akkor sokszor jelenik meg a keresők által nyújtott „hirdetési felületen”. Továbbá, minél jobban megfelel egy weboldal a kereső rangsorolási kritériumainak, annál előkelőbb és szembetűnőbb helyen fog megjelenni a találati listában.

9.2. A keresőoptimalizálás története

A webmesterek az 1990-es évek közepén az első keresők megjelenésével már elkezdték optimalizálni honlapjaikat. Ez eleinte csupán annyiból állt, hogy elküldték oldaluk webcímét a különböző keresőknek, amelyek indexelték azt. A folyamat során az illető kereső egy speciális program segítségével letöltötte a weboldalt a saját szerverére, ahol egy másik, indexelőnek nevezett program különböző információkat nyert ki belőle úgy, mint a tartalmazott szavakat és ezek sűrűségét. A honlaptulajdonosok kezdtek rájönni, nagy értékkel bír, ha a keresőknél jól szerepel oldaluk és egyre többet kezdtek foglalkozni az optimalizálással.

A keresőoptimalizálás kifejezés megszületése az 1997-es évre tehető, amikor a keresők nagyobb teret nyertek. Ekkor a keresőalgoritmusok még csak az oldal tartalmát és a META elemekben feltüntetett kulcsszavak alapján rangsoroltak. Erre hamar rájöttek a webmesterek és vissza is éltek vele, aminek következményeképp a felhasználók gyakran nem azt találták meg, amit ténylegesen kerestek. [16]

Így a keresők is változtatni próbáltak az algoritmusaikon, mivel az ő legfontosabb érdekük az volt, hogy a felhasználó elégedett legyen a találati listával. A megoldás az volt, hogy a webmesterektől független tényezők döntsenek az oldal fontosságáról. Még az évezred végén megjelent – a mai napig fennmaradó – link népszerűség, ami azon szempont alapján rangsorolja az oldalakat, hogy a többi weboldal közül hányan hivatkoznak a kérdéses weboldalra. Ennek a technikának megvolt az a hátránya, hogy a nagy, amúgy is népszerű oldalakat előre helyezte, holott lehet, hogy egy kisebb népszerűséggel bíró weboldal értékesebb információt tartalmaz. Észrevehető, hogy ez a rangsorolási eljárás teljes mértékben független az oldal tartalmától, ezáltal megnehezült az optimalizálás is, hiszen más független oldalokról érkező hivatkozásokat kellett gyűjteni.

A Stanford Egyetem két végzős hallgatója, Larry Page és Sergey Brin kifejlesztettek egy algoritmust, amely a weboldalakat értékeli. Az algoritmus által meghatározott érték a PageRank nevet kapta, amelyik az oldalra mutató külső hivatkozások számát és erősségét jelképezte. Leegyszerűsítve úgy lehetne leírni ezt a technológiát, hogy minden A oldalról B oldalra mutató hivatkozás egy, a B oldalra leadott szavazatnak minősül. A szavazat súlya függ attól, hogy a szavazatot leadó A oldalra hányan szavaztak, és így tovább. A felhasználók számára ez egy 1-től 10-ig terjedő skálán jelenik meg, tehát két ugyanolyan PageRank értékű oldal valójában egy bizonyos értéktartományon belül helyezkedik el. Ez az érték minden egyes weblapra kiszámításra kerül, vagyis ugyanazon webhely különböző lapjainak különböző PageRank értéke lehet.

Page és Brin 1998-ban megalapították a Google-t. Az egyre népesebb számú látogatóknak nagyon tetszett a kereső végletekig egyszerű kezelőfelülete. Mivel a Google nem csak az oldal tartalma (kulcsszavak sűrűsége, META elemek, oldalstruktúra, stb.) alapján rangsorolta az oldalakat, hanem külső tényezők (PageRank) szerint is, sokkal nehezebb volt trükközni a webmestereknek.

A legnagyobb keresők nem hozzák nyilvánosságra, hogy milyen szempontok szerint osztályoznak, ezzel is megnehezítve a webmesterek manipulációs próbálkozásait. A Google állítása szerint több, mint 200 különböző tényező szerint rangsorol.

Azok után, hogy a legtöbb nagy kereső átalakult portállá és maga a keresés háttérbe szorult, megkezdődött a Google térnyerése. Megtartotta a jól bevált egyszerű formáját és egyre inkább a releváns találati listák produkálására koncentrált. Korlátozott mértékben már tudott PDF-ben és Flash-ben is olvasni.

A keresőoptimalizálás is egyre nagyobb népszerűségnek kezdett örvendeni, így gyakorlatilag egybefonódott a Google-re történő optimalizálással, amely link-népszerűség szerint rangsorolta az oldalakat. A látogatottság érték, melyet a keresőoptimalizálás segítségével nagy hatékonysággal lehet növelni. Aki odafigyel a különböző keresők szempontjaira, az előnybe kerülhet a többi webmesterrel szemben.

9.3. A keresési folyamat

A tudatosan kereső felhasználók a szabadszavas keresőket és a tematikus katalógusokat használják arra, hogy rátaláljanak olyasmire, amit keresnek. Minőségi információra van szükségük, lehetőleg olyanra, ami választ ad a kereső-kifejezésekkel megadott kérdéseikre. Több esetben szerencse kérdése, hogy a keresés végén miért éppen a mi weboldalunkon kötnek ki, ám mindez befolyásolható is. Ismerkedjünk meg hát a keresési folyamat egészével, mely során a látogató eljut az oldalunkra:

- A keresőrobotnak meg kell találnia a webhelyet a keresőoldalra regisztrálás által, vagy automatikusan az oldalra mutató külső hivatkozások által.
- Az oldalon szerepelnie kell a kulcsszónak.
- A látogatónak rá kell keresnie a kulcsszóra, kereső-kifejezésre.
- A találati listában jó helyezést (leginkább Top 10 pozíciót) kell elérnie az oldalnak.
- A találati listának olyan kivonatot kell mutatnia az oldalról, ami felkelti a látogató érdeklődését, és rákattint az oldalra mutató linkre.

A szabadszavas keresők technológiájának lényege, hogy lehetővé teszik a beindexelt információhalmazban a kulcsszavak, kereső-kifejezések alapján történő keresést. A szabadszavas keresők a céltudatosan kereső felhasználók leggyakrabban használt keresési

formáját jelentik. A keresőoptimalizálás témaköre gyakorlatilag arról szól, hogy minél több effajta, céltudatosan kereső látogatót vonzzunk az oldalunkra. [15]

9.4. Az keresőoptimalizálás célja

Az elsődleges célunk tehát az, hogy a felhasználók kereséseinek minél nagyobb hányada végződjék az oldalunkra mutató linkre kattintással. Másodlagos cél, hogy minél több olyan látogatónk legyen, aki az oldalunk célközönségébe tartozik.

A reklámköltségek nagy részben csak addig hoznak közvetlenül új látogatókat az oldalra, ameddig a reklámkampány tart. A keresőoptimalizálásnak ezzel ellentétben időben elhúzódó pozitív hatása van a látogatottságra, akár még egy év után is biztosítja a látogatottság folyamatos növekedését. Ezen kívül pedig egy jól optimalizált oldal áttekinthetőbbé, használhatóbbá és informáltabbá válik. [15]

9.5. Keresőoptimalizálás lépésről lépésre

9.5.1. Konkurenciaelemzés

Sokan abban a tudatban élnek, hogy a vetélytársak megelőzése majdhogynem lehetetlen. Keresőoptimalizálással magunk mögé tudjuk utasítani a konkurenciát. Tartalmasabb, értékesebb oldalt kell készíteni és több figyelmet kell szentelni a link-építésre.

A konkurencia weboldalainak tanulmányozása többszörös haszonnal jár. Ötletet adhat a link-építéshez, hiszen a vetélytársak elemzésével fel tudjuk kutatni, milyen tematikus katalógusokba regisztrálták weboldalaikat. A konkurens oldalak szóhasználatát elemezve megtalálhatók a legtöbbit keresett keresőszavak. Ezek az információk felhasználhatók a keresőoptimalizálás folyamatában és hozzájárulhatnak ahhoz, hogy az oldalunk hatékonyabbá váljon. A konkurencia feltérképezése többek között azért is hasznos, mert megtudhatjuk, milyen témákkal kapcsolatban van esélyünk labdába rúgni.

Keressünk rá a lehető legrészletesebben, hogy a mienkhez képest milyen hasonló témájú oldalak találhatóak meg az interneten. Erre kiváló eszközök a tematikus katalógusok, a hasonló témákról írt cikkek, vagy a szabadszavas keresők. [18]

9.5.2. Megfelelő kulcsszavak megtalálása

Keresőoptimalizáláskor az egyik legfontosabb lépés annak eldöntése, hogy mire is optimalizáljuk az oldalakat. Feladatunk azon kulcsszavak, kifejezések megtalálása, melyre optimalizálni szeretnénk oldalunkat. Ez a lépés rendkívül fontos, mert hiába jó a weboldal bizonyos szavakra a találati listán, ha azokra az emberek nem keresnek, nem hoznak látogatókat. Nagyon sokan csak maguktól kitalálják a kulcsszavakat, de nem törődnek azzal, hogy vajon az adott helyzetben tényleg azok-e a legmegfelelőbb kifejezések? Sokszor nem sikerül magunktól beletalálni és kissé csalódottak leszünk, mert nem teljesülnek a kitűzött célok. Másrészt az sem mindig szerencsés, ha olyan szavakra próbálunk kezdetben optimalizálni, melyekre sokan keresnek ugyan, de olyan nagy a konkurencia, hogy nincs reális esély a jó eredményre. A cél az, hogy művelet végén ki tudjunk választani 4-5 olyan szót, melyre érdemes lesz optimalizálni. [18]

9.5.2.1. Hogyan gyűjtsünk lehetséges kulcsszavakat?

Kiindulásként össze kell gyűjteni a weboldal témájától függően 100-200 lehetséges kulcsszót minden megkötés nélkül. Ha összeírtuk azokat, amelyek eszünkbe jutottak, érdemes a konkurens weboldalakat is felkeresni és tovább keresni kulcsszavak után. Ehhez egyszerűen keressünk csak az eddig összegyűjtött szavakra és látogassuk meg a találati lista első néhány weboldalát és bővítsük ezek alapján a kulcsszó készletünket.

Ezen felül az interneten több kulcsszó ajánló eszköz is fellelhető. Használatuk egyszerű, csak meg kell adni egy vagy több kiinduló kulcsszót és az eszköz ezek alapján felajánl néhány újat. Azokat a szavakat, melyeket relevánsnak, témába vágónak érzünk, vegyünk fel a listára. Befejezésül vegyük hozzá a listához, a kigyűjtött szavak ragozott alakjait, illetve a többes számmal ellátott szavakat is. Ekkora már minden valószínűség szerint már elég hosszú lesz a lista. [14]

A kulcsszavak keresésénél figyeljünk arra, hogy egy keresőszó lehetőleg ne legyen túl általános. Az általános szavakra túl nehéz kezdetben optimalizálni, mert túlságosan erősek a konkurensek. Például ha egy cég debreceni ingatlanokkal foglalkozik, akkor az „ingatlan” szóra nehéz lesz optimalizálni, főképp ha a weboldal még fiatal. A „Debrecen ingatlan”, „Debreceni ingatlan”, „Debreceni ingatlanok” szavak inkább szóba kerülhetnek. Ezzel kisebb

munka árán, rövidebb idő alatt, jobb eredményeket lehet elérni. Persze később meg lehet próbálkozni a nehezebb szavakkal is (főképp, hogy a „debreceni ingatlanok” kifejezéssel egyúttal az „ingatlanok” szóra is már megjelenhet az oldal a találati listán).

9.5.2.2. Hogyan válasszuk ki a fontos kulcsszavakat?

Miután összegyűjtöttünk kellően sok kulcsszót, el kell dönteni, hogy melyik 4-5 legyen az, amelyre elsősorban optimalizáljuk weboldalunkat. A megfelelő kulcsszavak helyes kiválasztása rendkívül fontos, hiszen hiába jó az oldalunk olyan szavakra, melyekre az emberek nem nagyon keresnek. De az sem jó, ha olyan szavakkal próbálkozunk, melyekre nem tudunk előre kerülni, mert annyira erősek. Tehát az optimális, népszerű kulcsszavakat kell megtalálni.

Ehhez azt kellene tudni, hogy a kigyűjtött kulcsszavaink közül, melyiket hányszor keresik az emberek és milyen nagy a konkurencia az adott kulcsszóra. Vannak az interneten ingyenes és fizetős olyan eszközök, melyek megmondják, hogy a saját adatbázisuk szerint az adott kulcsszó mennyire népszerű.

A Google AdWords (adwords.google.com) szolgáltatása segítségével elég pontos eredményt kapunk. A szolgáltatás lényege, hogy a Google oldalain hirdethetjük weboldalunkat, az általunk meghatározott kulcsszavakra. Az AdWords-ön belül pedig ellenőrizhetjük az eredményeket, láthatjuk, hogy melyik kulcsszóra hányszor jelent meg a hirdetésünk (azaz hányszor kerestek rá). A kampányt a szakterülettől függően néhány hétig érdemes futtatni. Ekkor még a cél nem az, hogy minél több látogatót szerezzünk, ezért viszonylag alacsony költség ellenében (kb. 3 000 Ft) pontosan megtudhatjuk, hogy mire lehet érdemes optimalizálni. Akik nem akarnak ennyi pénzt elkölteni, vagy nem akarják megvárni azt a néhány hetet, míg értékelhetőek az eredmények, használhatják az AdWords-ön belül a Google Keyword Tool eszközt, mely megmondja, hogy az egyes kulcsszavakat relatíve mennyien keresték és jelenleg milyen erős a hirdetési konkurencia. Bármelyik megoldást is választjuk, a művelet végén egészítsük ki a listánkat azzal, hogy az egyes kulcsszavak mennyire népszerűek. [18]

A másik tényező, amit mérhetővé kellene tenni az az, hogy milyen nagy a konkurencia az egyes kulcsszavakra, vagyis mennyire lesz nehéz dolgunk jó helyre kerülni a találati listán. Ehhez nem kell más tennünk, mint rákeresni a kulcsszavakra a Google oldalán és megnézni,

hogy hány releváns oldalt talált (természetesen ebből nem tudjuk meg pontosan, hogy a találati lista elején található oldalak mennyire erősek, de az szinte biztos, hogy ha sok oldal foglalkozik a témával, lesz néhány erős, jól optimalizált weboldal is köztük).

Ezeket az értékeket is jegyezzük fel, majd rendezzük a listát a népszerűség szerint csökkenő sorrendbe, tehát a legjobb, legnépszerűbb kulcsszavak legyenek elől. Szubjektív döntésünk kell legyen, hogy kiválasszuk azt a 4-5 kulcsszót, melyre optimalizálni fogunk. Ha optimisták vagyunk az oldalunk erősségét illetően, akkor választhatjuk a legnépszerűbb kulcsszavakat, még akkor is, ha azokra a legnagyobb a konkurencia. Ellenkező esetben egy optimális kulcsszókészletet válogassunk össze. A művelet végén ki kell jelöljünk a fontos kulcsszavainkat, melyekkel foglalkozni fogunk a továbbiakban.

Fontos szemmel tartani, hogy a kiválasztott kulcsszavakra rákeresre hol helyezkedik el az arra a kulcsszóra kihelyezett weboldal, vagy a webhely nyitólapja. Ha egy kulcsszóval kapcsolatban rosszul döntöttünk, akkor ennek megfelelően változtatnunk kell.

Ügyeljünk azonban a kulcsszóhalmazásra, azaz ha tartalomban túl sokszor használjuk a kulcsszót a magas kulcsszósűrűség eléréséhez. Ilyen esetekben a kereső büntetheti oldalunkat és hátrarangsorolja. Főképp hosszú szövegek esetén gyanús a nagyon magas kulcsszósűrűség illetve ha kulcsszavak sokszor vesszővel elválasztva szerepelnek egymás mellett.

Mindenek előtt tehát reálisan fel kell mérni a saját területünkön lévő versenyt és ehhez kell igazítani a kezdeti célkitűzéseket. Ha valaki egyből a beugrik a mély vízbe sokáig kellemetlenül fogja érezni magát és nem lesz elégedett a honlapjával. Természetesen az olyan területeken, ahol alacsony a verseny magasabb célokat is ki lehet tűzni már a honlap indulásakor. [14]

9.5.3. Domain-név kiválasztása

A jól megfogalmazott, kifejező domain-név meghatározása nagyon fontos feladat, ezért ennél a lépésnél javasolt elidőzni és jól átgondolni. A domain-név egyfajta belépő a weboldalhoz, olyan, mint a személynév. Egy szépen csengő, különleges név hallatán szinte azonnal felkapjuk a fejünket, míg az általános, gyakran használt nevekre oda sem figyelünk.

Legfontosabb választási szempont a könnyen megjegyezhetőség a leendő látogatók számára. Nagyobb látogatottságot érhetünk el, ha egy rövid címet választunk, mivel a félregépelés esélye töredékére egy 4-5 karakterből álló webcím esetén egy 10-12 karakteres

címmel szemben. Nagyon indokolt esetben válasszunk csak 10 karakternél hosszabb domain-nevet. Nagyon rövid – ügyfelek által kevésbé ismert – mozaikszavak választása sem vezet feltétlenül sikerhez.

A rövidebb webcímeknek másik nagy előnye, hogy a nyomtatott és egyéb kreatív anyagokon kisebb helyet foglalnak el, így elég egy kisebb reklámfelületet megvásárolni ugyanazon hatás eléréséhez. Ebből következik, hogy olyan reklámhordozók esetén, ahol a méretek korlátozottak (pl. sapka, autó, zászló) nagyobb betűvel – azaz távolabbról is jól láthatóan – lehet elhelyezni webcímünket.

A Google algoritmusa amely alapján a weboldalak rangsorolja, sokban változott az idők folyamán, de vannak fix, állandó dolgok. A rangsorolásban kiemelt előnyt élveznek a teljes kulcsszavas egyezéssel rendelkező domainek a domainben szereplő kifejezésre (pl. pizza.hu – pizza). [14]

Ha a weboldal témája, amelyhez domaint kell választanunk, különösen illeszkedik egy adott kulcsszóhoz vagy kifejezéshez és az adott kulcsszót tartalmazó domain valamely változata még szabad (kiemelten .com, .net domainek, magyar domain esetén: .hu) mindenképp tanácsos ezt a domain lefoglalni.

A kötőjeles nevektől lehetőleg óvakodjunk. A domain piacon egy kötőjel már nagyon nagy értékvesztést jelent egy névben, akkor is, ha egy nagyon frekvencián keresett kulcsszó párt tartalmaz.

A kötőjelek alkalmazásának azonban előnye is lehet: nyomtatott anyagokon jobban elkülöníthető két szó, így olvashatóbbá tehető és ezért megjegyezhetőbbé válik egy domain. Pl. az újsághirdetéseknél sikeresebb választás a fogyjon-le-azonnal.hu, mint a fogyjonleazonnal.hu, de egy rádióreklámban az utóbbit a célravezetőbb alkalmazni.

Ne válasszunk ismert márkanévvel azonos címet, ugyanis később a márkanév, védjegy birtokosa peres úton elveszi tőlünk a címet és a domain elvesztésén kívül a perköltség terhe is a nyakunkba hullik. [18]

9.5.4. Értékes tartalom, vagy szolgáltatás hozzáadása

Miután meghatároztuk az oldal célját, célközönségét, döntenünk kell, hogy mi legyen az a hozzáadott érték, amivel hosszútávon látogatók vonzhatunk. Ha egy weboldal nem nyújt valamilyen egyedit, értékeset, hasznosat, újat, akkor a látogatottság növelése céljából

folyamatos erőfeszítéseket kell tenni. Ha érkezni fognak is a keresők felől látogatók, nagy részük rögtön tovább is áll, magától pedig senki sem fog hivatkozást elhelyezni az oldalra.

Ha egy oldal értékes, akkor biztosak lehetünk benne, hogy a keresőoptimalizálás időszakában magától is nőni fog úgy a látogatók száma, mint a külső hivatkozásoké.

Egy olyan oldal, amin csak bemutatkozás, vagy egyéb reklám található, a látogatók közül általában keveseket érdekel. Valami pluszt kell nyújtani, ha hosszú távon, érdemben szeretnénk növelni látogatóink számát. Érdemes ezért a keresőoptimalizálásba fektetett energia mellett valamilyen értékes tartalom, vagy hasznos szolgáltatás létrehozásán elgondolkozni.

Hosszútávon ugyanis csak akkor garantált a weboldal jó helyezése a találati listában, ha az oldalon értékes tartalom, vagy szolgáltatás található. Ilyenkor olyan oldalakra is maguktól felkerülnek a mi oldalunkra mutató külső hivatkozások, melyekre soha sem gondoltunk volna. Ha jó a tartalom, szolgáltatás, akkor még akkor is növekedni fog magától a látogatottság, amikor már hónapok óta nem is frissült a weboldal. [18]

Miután elemeztük a konkurenciát kiderül, hogy mi az a témakör, vagy ezen belül mi az a specialitás, amivel kapcsolatban még nem hoztak létre tartalmat, vagy szolgáltatást, esetleg mindezt nem annyira jól, vagy számunkra és talán mások számára sem tetsző módon.

Bármit is teszünk fel az internetre, ha azt jól tesszük, biztosak lehetünk benne, hogy lesznek a téma iránt érdeklődő látogatók, akik hasznosnak találják majd a tartalmat, szolgáltatást. A potenciális látogatók száma attól függ, mennyire speciális a téma, illetve hogy mekkora a konkurencia.

9.5.5. Tartalomkezelő rendszerek használata

Rengeteg időt és energiát megspórolhatunk, ha valamilyen előre megírt tartalomkezelő rendszert használunk weboldalunk elkészítésére, testreszabására. A megtakarított időt így az értékes tartalom, illetve szolgáltatás létrehozásába, bővítésébe, tökéletesítésébe fektethetjük, valamint élhetünk a dinamikus weboldalak előnyeivel anélkül, hogy bármit is programoznunk kellene. [17] A következő fejezetben bővebben lesz szó a tartalomkezelő rendszerekről, valamint közelebbről bemutatjuk a Magyarországon legelterjedtebb és leghíresebb ilyen rendszert, a Joomla!-t.

9.5.6. A megfelelő tárhely kiválasztása

Érdemes képpen lenni a tárhely árak, szolgáltatási minőség ügyében. A legfontosabb szempont, hogy az adott szolgáltató megbízható legyen, a hosztolt oldal mindig elérhető legyen. Ez technikai okokból persze nem lehetséges, 100%-os elérhetőség nem létezik. A tárhely kiválasztását befolyásolja a választott tartalomkezelő rendszer, vagy a használni kívánt szkriptnyelv is.

Kétségkívül az ingyenes tárhelyes jelentik a legolcsóbb megoldást, ám legtöbbször egy reklámcsíkot helyeznek el weboldalunk fölé, ami különösképp üzleti weboldal esetén nagyon hátrányos lehet. Szolgáltatásaik bizonyos mértékben korlátozottak, ezért ehhez a lehetőséghez csak a legvégső esetben forduljunk. A fizetős tárhelyek előnye, hogy a szaktudás mellett kielégítő technikai háttér garantálja, hogy a pénzünkért minőségi szolgáltatást kapunk.

Amennyiben csak a hazai piacra szánjuk oldalunkat, akkor a legtöbb szolgáltató 100 MB/sec sebességű vonalat biztosít, ami tökéletesen elegendő. Azonban, ha külföldi internetet is megcélzunk érdemes megtudni, hogy mekkora sebességgel tudják az egyes szolgáltatók ezt megvalósítani. Általában a belföldi sebesség töredékét tudják csak garantálni, ezért külön vonalat kell bérelni erre a célra. Itt a költségek nagy mértékben változhatnak, ezért körültekintően válasszunk. [18]

Nem elhanyagolható, hogy a szolgáltatások között szerepeljen látogatottsági statisztikamérés, így pontos képet kaphatunk az oldalunk forgalmáról. Keresőoptimalizálás szempontjából fontos lehet továbbá, hogy lehet-e aldomaineket létrehozni és azt milyen számban tehetjük meg.

9.5.7. Keresőoldalakra való regisztrálás

Miután már működik a domain-név és elhelyeztünk valamilyen tartalmat a webhelyünkre, az első dolgunk a keresőoldalakra való regisztrálás legyen. Bár sokan azt hiszik, ez nagy jelentőséggel bír a weblapok látogatottság-növelése szempontjából, jelentősége csupán egy új webhely indulásakor lehet.

Az interneten böngészők általában kétfajta keresőtípussal szoktak információkat keresni: a szabadszavas keresőkkel és a tematikus katalógusok segítségével. Teljesen felesleges minden keresőbe regisztrálni, elég csak a legfontosabb szabadszavas keresők esetében megtenni.

Amennyiben teljes értékű linkek mutatnak oldalunkra, a keresőrobotok pár nap alatt megtalálják a webhelyet, ami ennek következtében megjelenik a találati listában.

9.5.8. Jól megírt weboldal létrehozása

Ha az oldalunkon elhelyezett szöveg megfelelő módon megfogalmazott és jól strukturált, valamint a webhely minden oldalára mutat teljes értékű szöveges link, akkor már így jó esélyünk van arra, hogy a látogatottság elfogadható mértékű legyen.

Ha a szabadszavas keresők felől várunk forgalmat, akkor a szöveges tartalomra kell helyezni a hangsúlyt, mivel a keresők szöveges oldalak között keresnek, nem pedig multimédiás alkalmazások között. Az interaktív Flash és Java alkalmazások felhasználása esetén számon probléma merülhet fel. Bármilyen nagy is legyen a webhely, ha összes funkciója egy Flash, vagy Java fájlban található, akkor gyakorlatilag egy üres oldalként tekinthető, amelybe az összes információ a keresők által nehezen, vagy egyáltalán nem feldolgozható objektum formájában van beágyazva. Bár a Google képes olvasni az .swf fájlokban, a technikai korlátai miatt egy HTML oldallal mégis sokkal több látogatót lehet vonzani.

A keresőoldalakon való láthatóságot már azzal biztosíthatjuk, ha teljes értékű szöveges linkek mutatnak a webhelyünk minden oldalára. Több oldalon a téma miatt leginkább vizuális információ található képek formájában. A látvány elsődlegessége miatt legtöbbször nem hiányzik a képek mellől semmilyen leírás. Azonban, ha a képek mellé megfelelő szöveget is társítunk, akkor nemcsak a képkeresők, hanem a szabadszavas keresők felől is várhatunk forgalmat.[14]

Ha valamilyen tartalomkezelő rendszerrel írjuk meg oldalunkat, akkor csak a kulcsszavak kiválasztása és elhelyezése, webhelyünk strukturálása, valamint az oldal címének (TITLE) és leírásának (Description Meta tagok) megfelelő megadása a feladatunk.

Abban az esetben, ha saját kézzel írjuk meg az oldalt, óvakodjunk a már korábban említett problémás technológiák (Flash, Java) használatától, és figyeljünk, hogy oldalunkat a webes szabványok szerint írjuk. Az esetek többségében nem történik semmi, ha a weboldal nem követi teljes mértékben a szabványokat, néhány esetben azonban problémákat okozhat. Vagy azért, mert rosszul használják a problémás technológiákat, vagy pedig egyszerűen egy szintaktikai hiba megakadályozza az oldal további részeinek értelmezését. Előfordulhat, hogy

az oldal első felében van valamilyen hiba, és így könnyen megtörténhet, hogy a Google kereső adatbázisában csak a kiegészítő találatok közé fog kerülni. Törekedjünk így hát arra, hogy az oldalunkon található összes információ hozzáférhető legyen olyan programok számára, melyek kizárólag az alapvető (X)HTML szabványokat ismerik. [18]

Különböző online és offline validátorok vannak segítségünkre abban, hogy ellenőrizzük, oldalunk megfelel-e a szabványoknak. Többek között a W3C-nek is van ilyen szolgáltatása, ahol a már interneten elérhető oldalakat, de akár a gépünkről feltöltött fájlokat is lehet ellenőriztetni:

HTML és XHTML validator: <http://validator.w3.org>

CSS validator: <http://jigsaw.w3.org/css-validator>

9.5.9. Link-népszerűség növelés

Miután oldalunk hozzáadott értékkel bír, akkor ez a látogatók növekedésének öngerjesztő folyamatát indíthatja el, azonban nem árt ezt a folyamatot meggyorsítani, vagy támogatni kiváltképp, ha nem tudunk sok hozzáadott értéket elhelyezni az oldalon, és/vagy nem akarunk hónapokat várni arra, hogy felfedezzék oldalunk.

A külső hivatkozások nemcsak közvetlenül növelik a látogatottságot, hanem hozzájárulnak a keresőoptimalizálás hatékonyságának növeléséhez. Ekkor viszont már nemcsak a bejövő linkek száma, hanem milyensége is fontos szerepet kap.

Érdeemes a tematikus katalógusokba regisztrálással kezdeni, de egyéb módszerek is vannak: a hasonló témájú oldalakról mutató linkek számának növelése érdekében kérjük meg az első lépésben már feltérképezett oldalakat, melyek nem feltétlenül számítanak konkurenciának és minden más, a témához kapcsolódó oldal üzemeltetőit, hogy helyezzenek el oldalaikon ránk mutató hivatkozásokat, továbbá adjuk hozzá az oldalunkra mutató linkeket olyan oldalakon ahol hozzászólások, szerkesztések formájában magunk is hozzáadhatunk tartalmat (fórumok, blogok, közösségi oldalak, wikik). [18]

9.5.10. A tartalom rendszeres frissítése

A régóta változatlan tartalmú oldalak hajlamosak hátrébb csúszni a találati listákon. A keresőrobotok rendszeres időközönként látogatják az oldalunkat és figyelik a változásokat,

ezért törekedjünk a frissítésre. Ez történhet a tartalom aktualizálásával, bővítésével, de akár a látogatók hozzászólásával, vagy bármilyen automatikusan generált tartalom hozzáadásával. Tartalomkezelő rendszerek esetén például egyszerűen építhetünk be olyan komponenseket, amelyet lehetőséget biztosítanak a látogatók számára, hogy hozzászóljanak az adott témához.

9.5.11. Nyomonkövetés

Mielőtt azt gondolnánk, hogy az oldal elkészültével véget ért minden feladatunk, a java még csak most következik. Rendszeresen nyomon kell követnünk a látogatói statisztikákat, növelnünk kell a külső hivatkozások számát, valamint figyelni a találati listák változásait. Az eredmények függvényében kell módosítanunk, tökéletesítenünk az oldalt. Némi idő elteltével kiderül, milyen kulcsszavakkal kapcsolatban szükséges erősíteni, milyen kulcsszavakat kell használni az oldal kulcsfontosságú részeiben. El kell dönteni, hogy a sikeres kulcsszavak irányába mozdulunk, vagy az általunk kitűzött irányt tartva a kevésbé sikeres kulcsszavakat erőltetjük.

Statisztikailag minél több a látogatónk, annál többen fogják meglínelni az oldalt, és minél többen hivatkoznak az oldalra, annál több látogatónk fog érkezni a keresők felől, és így tovább... [18]

10. Tartalomkezelő rendszerek

10.1. Mik azok a tartalomkezelő rendszerek?

Amikor valaki a saját weblapjának építésébe fog, általában valamilyen WYSIWYG-szerkesztőhöz nyúl, mint például az Adobe Dreamweaver, vagy a Microsoft FrontPage. Ez meg is teszi egy néhány oldalból álló bemutatkozó honlaphoz, ám amikor később előjönnek az igények, az újdonsült webmester rájön, komoly programozói ismeret nélkül ezeket nem tudja megvalósítani.

Az egyszerű és gyors megoldást a tartalomkezelő rendszer (angolul: Content Management System, röviden: CMS) jelenti. Ez egy olyan szerveren futó programcsomag, ami lehetővé teszi a tartalom szerkesztését egy távolról elérhető adminisztrációs felület segítségével, és publikálását, általában hypertext formátumban. Alkalmas strukturált és/vagy strukturálatlan tartalom kezelésére, tárolására.

A tartalomkezelő rendszerek egyik legfontosabb tulajdonsága és nagy előnye a dinamizmus. Egy hagyományos FTP csatornán keresztül csupán egyszer, az első alkalommal kell feltölteni a fájlokat, amikor üzembe helyezzük a szerveren a honlapot. Ezek után a tartalmat bárhol és bármikor meg lehet változtatni. Különböző felhasználókat azonosítva és hozzájuk különféle jogosultságokat rendelve lehetővé tehetik akár több ember együttes munkáját is. Használatukkal bárki létrehozhat profi dinamikus honlapokat anélkül, hogy komoly informatikai tudással rendelkezne. [17]

Keresőoptimalizálás szempontjából azért lehetnek fontosak a tartalomkezelő rendszerek, mert ezek segítségével a tulajdonos az oldal értékes tartalommal való feltöltésére koncentrálhat, és élhet a dinamikus weboldalak minden előnyével anélkül, hogy bármit is programoznia kellene. Egy honlap sikere már nem azon múlik, hogy a tulajdonosa mennyi pénzt fordít rá, hanem azon, hogy mennyire értékes tartalmat tud elhelyezni az oldalaira. [15]

10.2. Tartalomkezelő rendszerek osztályozása

A tartalomkezelő rendszerek többsége MySQL adatbázist használ és PHP szerveroldali programozási nyelven íródott (Joomla!, Drupal, E107), de vannak Java-ban (Dspace, OpenEdit) és ASP-ben (Allinta) írt rendszerek is. A piacon több száz CMS létezik, a legtöbb

nyílt forrású és szabad licencű, ám vannak egyedi fejlesztésű és kereskedelmi termékek is. A továbbiakban a Joomla! nevű ingyenes tartalomkezelő rendszert fogom bemutatni. [17]

10.3. A Joomla! tartalomkezelő rendszer

A CMS-t először használók, miután kipróbáltak néhányat, általában a Joomla! mellett döntenek, mivel egyszerű a telepítése és használata is.

A Joomla! egy ingyenes, GNU/GPL alatt kiadott tartalomkezelő rendszer, melynek használata nagyon könnyen elsajátítható és segítségével profi színvonalú weblap építhető. Mint a nyílt forráskódú tartalomkezelők piacvezetője, kiváló választás úgy magánszemélyek, mint kis-, de akár nagyvállalkozások weblapjaihoz is. Verzióról verzióra új szolgáltatásokkal és funkciókkal bővül és a világon több millió weboldal motorjául szolgál.

Támogatja a keresőbarát URL-eket, az OpenID és LDAP-hitelesítést, RSS- és Atom-hírcsatornák kezelését, lehetővé teszi a cikkek kategóriákba, és szekciókba sorolását. Ezekon kívül rendelkezik felhasználó- és hirdetés-kezelővel, van beépített menürendszere, lehetőség van tömeges üzenetküldésre, de akár szavazásokat is kiírhatók a weboldal látogatói számára.

A Joomla!, mint keretrendszer lehetőséget nyújt az alapfunkciók kibővítésére, így a könnyen telepíthető kiegészítőkkel pár perc alatt új, hasznos és látványos funkciókkal bővíthető egy weblap. Egyszerűen telepíthető egy jól mutató képtár, videó lejátszó, vendégkönyv, fórumrendszer, dokumentumkezelő, de akár egy bevásárló rendszer is. [13]

A Joomla! itthoni terjesztését és támogatását jelenleg a 2008-ban alapított Magyar Joomla! Felhasználók Nemzetközi Egyesülete (www.joomla.org.hu) végzi. A nonprofit és közhasznú egyesület a Joomla! projekt jogi pénzügyi ügyeit is bonyolító Open Source Matters (OSM) által elismert szervezet, az egyik legnépszerűbb hazai nyílt forráskódú közösség.

Az egyesület vezetősége és tagjai önkéntesek, akik a szabadidejüket áldozzák a közösség híreinek továbbadására, a honosítások készítésére és gondozására, az oktatásra (wiki.joomla.org.hu), valamint a fórumban (forum.joomla.org.hu) a problémák megválaszolására.

Dokumentációs, fejlesztési és fordítási munkacsoportjuk nyilvános, bárki csatlakozhat, aki szeretne hozzájárulni a Joomla! itthoni sikeréhez és elterjedéséhez. Az egyesület irodát nem tart fenn, ezért személyes tanácsadásra nincs lehetőség.

10.3.1. Rendszerkövetelmények

A Joomla! futtatásához szükséges az Apache webservert 1.3-as, a PHP-motor legalább 5.2-es és a MySQL adatbázis-kezelő 3.23-as, vagy újabb verziója. A PHP4 már rég nem támogatott és adatbázis tekintetében is ajánlott olyan szolgáltatót keresni, ahol MySQL5 fut, ugyanis a Joomla! 1.5 már teljes körűen támogatja az UTF-8 karakterkódolást.

Sok tárhelyszolgáltató bekapcsolja a Register Globals PHP-funkciót, aminek a fejlesztők általi helytelen használata rengeteg veszély forrása lehet, így nem tanácsos a használata. Mostanra már sem a Joomla!, sem a bővítmények nem igényelik, sőt a PHP 6-ból el is távolították.

A megosztott szerverek biztonsági problémáit kijavítani megkísérlő Safe Mode PHP-beállításról sokan azt hiszik, hogy megoldja szerverük biztonságát, így aktiválják. Ám ez sok kellemetlenséget okozhat például a bővítmények telepítésekor.

Sokan próbálkoznak a Joomla! ingyenes tárhelyre történő telepítésével, ami több hiba forrása lehet, mivel az ingyenes szolgáltatók biztonsági okokból letiltanak számos PHP- és adatbázis-funkciót. Manapság már nem drágák a tárhelyek, így ajánlott fizetős választani és Magyarországon is egyre több Joomla!-specializált szolgáltató ajánlatából van lehetőség válogatni.

10.3.2. Telepítés

A Joomla! letöltése (www.joomla.org/download.html) után csomagoljuk azt ki a helyi gépünkre, majd az archívum tartalmát FTP-klienssel töltjük fel a tárhelyünk gyökerébe. Ha ezzel készen vagyunk, írjuk be a böngészőnk címsorába leendő weblapunk címét (például: <http://www.sajatweblapom.hu>), s ha mindent jól csináltunk, akkor az asztali alkalmazások telepítőprogramjaihoz hasonlóan betöltődik a Joomla! telepítő 1. lépése, amelyben kiválaszthatjuk a folyamat nyelvét, így a magyart is.

A telepítés 2. lépésére érdemes több figyelmet fordítanunk: a Joomla! ugyanis itt ellenőrzi a szerver beállításait. Ha valami pirossal jelölt, annak ajánlatos azonnal utánanézni, mivel a később felmerülő problémák ettől függenek. Ha például a már korábban említett PHP „Safe Mode” beállítást bekapcsolta a rendszergazda, nem fogunk tudni bővítményeket telepíteni.

Ezután olvashatjuk el a licenc-szerződést, majd 4. lépésként az adatbázis konfigurálása következik. Az *Alapbeállítások* rész kitöltéséhez a kiszolgáló és az adatbázis nevét, a hozzáféréshez szükséges felhasználónevet és jelszót a tárhelyszolgáltatótól kapjuk. A Joomla! alapértelmezés szerint a *jos_* táblanév előtagot ajánlja fel a *Speciális* beállítások részben, amelyet tetszés szerint megváltoztathatunk.

Az 1.5-ös Joomla! telepítőjének újdonsága az FTP-réteg kezelése, amelyet a fentebb említett „Safe Mode” PHP-beállítás miatt vezettek be. Az FTP-hitelesítési adatok és elérési út birtokában ugyanis lehetőség van a fájlkezelésre. Ezeket a paramétereket az 5. lépésben adhatjuk meg.

A 6. lépésben kell beállítanunk a weboldal nevét, a főadminisztrátori jelszavát, illetve e-mail címét. Kezdőknek ajánlott, hogy kattintsanak rá a *Mintaadatok telepítése* gombra is, mert a telepítés befejezése után az adminisztrációba bejelentkezve áttekinthetőbb képet kapnak a rendszer felépítéséről.

Az utolsó, 7. lépésben látható kérésnek az /installation könyvtár eltávolításának tegyünk eleget, mert különben nem tudunk hozzáférni a felhasználói oldalhoz. A jobb felső részben elhelyezett megfelelő gombra kattintva léphetünk be az adminisztrációba, illetve tekinthetjük meg a tényleges weboldalt.

A telepítéssel néhány perc alatt elkészülhetünk, s tulajdonképpen kész is a dinamikus, PHP/MySQL-alapú weblapunk váza, amelyet csak fel kell töltenünk saját tartalommal, képekkel, videókkal.

10.3.3. Bővítmények

A Joomla! népszerűségének egyik tagadhatatlan indoka a bővítmények széles választéka. A Joomla! Extensions Directory (röviden: JED, magyarul: Joomla! Bővítmények Tára, extensions.joomla.org) gyűjteménye a fejlesztői közösség jóvoltából több mint 4000 kiegészítőt tart nyilván.

A Joomla!-bővítmények elvileg mindenféle igényt ki tudnak elégíteni. Létezik hozzá fórum, apróhirdető, közösségépítő, ügyfélkapcsolat-kezelő, hogy csak a legnépszerűbbeket említsük. A sokak által használt VirtueMart e-kereskedelmi megoldás a phpShop Joomla!-ba építéséből nőtte ki magát, az Agora fórum pedig a punBB Joomla!-integrációja.

Igaz, hogy a Phoca Gallery képgaléria-komponens nem büszkélkedhet ilyen felmenőkkel, ám támogatja a PicLens diashow-t, illetve a Shadowbox és a Highslide JS függvénykönyvtár beépítésével különleges effektusokkal nézegethetik a látogatók a képeket.

A Joomla! rugalmasságának köszönhetően akár egy külön telepített phpBB fórum tagjainak adatbázisát is összekapcsolhatjuk a Joomla! regisztrált tagjainak adatbázisával.

A bővítmények széles választékának köszönhetően szinte bármilyen elképzelés megvalósítható komolyabb programozói ismeret nélkül.

10.3.3.1. Bővítménytípusok

A Joomla!, mint keretrendszer, a következő bővítménytípusokat támogatja:

- A **komponens** az oldal fő törzsében leképezésre kerülő, legnagyobb és legösszetettebb bővítménytípus, minialkalmazásnak is tekinthetjük. A komponensekkel történik az adatok kezelése, ők állítják be a megjelenítéseket, s különféle funkciókat biztosítanak. A komponensek teszik lehetővé a rendszer képességeinek nagymértékű bővítését.
- A **modul** megjelenítése az ún. modulpozíciókban történik, bárhol elhelyezhetjük a weblapunkon. Nem olyan összetett, mint a komponens; a kisalkalmazás szerepkörét tölti be. Kiegészítheti a komponensek szolgáltatásait (például egy fórum legfrissebb hozzászólásait megjelenítő modul), de önálló funkciója is lehet (például az oldal aktív látogatóinak számát megjelenítő modul).
- A legspeciálisabb bővítménytípus a **beépülő modul** (plug-in). A Joomla! és a felhasználó között működik, egy előre definiált esemény megtörténésekor fut le. A Joomla! 1.5-ben például beépülő modulokkal valósítják meg az OpenID- és LDAP-hitelesítést.
- A **nyelvi kiegészítők** az alaprendszer kezelőfelületének a telepített nyelvre történő átváltását teszik lehetővé.
- A felhasználói és a kiszolgáló oldalra is készíthető **sablon** a weblapunk grafikai terve, ez felelős az oldal elrendezéséért.

10.3.3.2. Alapfunkciók a bővítményekben

Már a Joomla! Alaprendszerével együtt kapunk több hasznos bővítményt, amelyekkel kezelhetjük a hirdetéseket, a szavazásokat, a webcímgyűjteményeket, az RSS-/Atom-hírcsatornákat.

- A *Hirdetések* komponens biztosítja a reklámkategóriák létrehozását, a hirdetők nyilvántartását és a hirdetések kezelését.
- A *Kapcsolat* komponenssel vihetjük fel és listázhatjuk ki a weblap szerkesztőinek elérhetőségi adatait, illetve a *Kapcsolattartó* képernyőn, jobb oldalt az *E-mail paramétere*i részben engedélyezhetjük az e-mail űrlap megjelenítését, amivel a látogatók üzenetet küldhetnek. Így elkerülhetjük, hogy az e-mail címeket begyűjtő robotok megszerezzék elérhetőségünket, valamint letilthatunk az e-mail címbe, az üzenet tárgyában vagy szövegben előforduló szavakat, ami némi védelmet nyújt a spammerek ellen.
- A *Hírcsatornák* komponenssel kategóriákba csoportosított RSS-olvasót adhatunk a látogatóinknak. Az egyes csatornák beállításában megadhatjuk, hogy hány hírelemet listázzon ki, illetve a gyorsítótárzás időtartalmát is meghatározhatjuk.
- A *Szavazások* komponenssel némi interaktivitásra készíthetjük látogatóinkat azáltal, hogy a feltett kérdésre választ adnak a legfeljebb 12 lehetőség valamelyikével. A *Késleltetés* paraméter értékének növelésével, vagy csökkentésével szabályozhatjuk, hogy milyen hosszú időtartamnak kell eltelnie ugyanazon felhasználó két szavazása között. A modulkezelőből hozzáférhető *Szavazások* (mod_poll) modul paramétereiben választhatjuk ki, hogy abban a modulpozícióban melyik szavazást kívánjuk megjeleníteni a felhasználói oldalon.
- Amennyiben van olyan szolgáltatásunk, amit csak a regisztrált felhasználók számára kívánunk hozzáférhetővé tenni, akkor a látogatóink a *Belépés* modulban található *Fiók létrehozása* hivatkozás követésével hozhatják létre felhasználói fiókjaikat. Az adminisztrációban, a *Globális beállítások* képernyő *Rendszer* fülén, a *Felhasználó beállításai* részben tanácsos engedélyezni az *Új felhasználói fiók aktiválása* beállítást, mert új felhasználó regisztrálásakor a Joomla! aktiváló hivatkozást küld a látogatónak. Erre a linkre kattintva erősíthetik meg a felhasználók a regisztrálási szándékukat, amivel megakadályozhatjuk a spammerek térnyerését.

- A Joomla! 1.5 már támogatja az OpenID- és LDAP-hitelesítést is, illetve a Gmail e-mail fiókkal rendelkező felhasználók is használhatják gmailes bejelentkezési adataikat, amennyiben a *Beépülő modul-kezelőben* aktiváltuk ezeket a plugineket.

10.3.3.3. Bővítmények telepítése

A Joomla!-bővítményeket többnyire .zip formátumú archívumokba csomagolva terjesztik. Elő szokott fordulni még a .tar, .gz, .tar.bz formátum, ugyanis a Joomla! telepítője ezeket az archív fájlokat is ki tudja bontani. Tartalmuknak a szerver megfelelő könyvtárába történő feltöltése nem elég. Ahhoz, hogy a Joomla! használni is tudja őket, az adatbázis megfelelő tábláiba is be kell őket jegyeznünk. Megoldható ez kézzel is, de van egy sokkal egyszerűbb és elegánsabb megoldás: a telepítés.

A telepítést a *Bővítmények* menü *Telepítés/Eltávolítás* menüpontjából megnyitható *Bővítménykezelő* képernyőn végezhetjük el. Ez a művelet valamivel előrébb jár az asztali programokhoz képest, mert a telepítendő fájlt le sem kell töltenünk magunknak. Mindössze csak a távoli állomány webcímére van szükségünk, amelyet a *Telepítési URL* mezőbe kell bemásolnunk, majd a *Telepítés* gombot megnyomnunk.

Ha valamilyen okból kifolyólag mégsem működne ez a módszer, még mindig van lehetőség a saját gépünkre letöltött bővítmény csomagként történő telepítésére.

10.3.3.4. Bővítmények adminisztrációja

- **Komponensek**

Valamely komponens beállításaihoz és adataihoz a *Komponensek* menü vonatkozó menüpontjából férhetünk hozzá. A funkcióban gazdag komponensek saját vezérlőpulttal is rendelkezhetnek. Egyes összetevők beállításainak testreszabását a hozzájuk készített menüpontban végezhetjük el.

- **Modulok**

A modulokat a *Bővítmények* menü *Modulkezelő* menüpontjából megnyitható képernyőn kezelhetjük. Sorrendjüket a cikkekhez hasonló módon rendezhetjük át, állíthatjuk be a hozzáférést, illetve engedélyezhetjük a megjelenítést. A modulokból a modulkezelő eszköztárán elhelyezett *Másolás* gombbal készíthetünk másolatot. Miért lehet erre szükség?

Vegyük például a hirdetések. Egy modult csak egy modulpozícióban tudunk megjeleníteni, tehát ahhoz, hogy az oldalon máshova is kerülhessen hirdetés, szükségünk van a *mod_banners* modul újabb példányára, amelyben azt a modulpozíciót választhatjuk ki, ahová a másik hirdetést tenni akarjuk.

A moduloknak eltérők a paramétereik, amelyeket ha módosítani szeretnénk, akkor a modulkezelőben kattintsunk rá, majd a betöltődő oldalon tegyük meg a változtatásokat. Felmerülhet olyan szituáció, hogy a modult csak néhány oldalon akarjuk megjeleníteni. Erre a *Hozzárendelés menüpontokhoz* részben van lehetőség. Több modul speciális paraméterei közt van lehetőség a gyorsítótárzás engedélyezésére. Ezzel érdemes eljátszadózni, mert a felhasználói oldalon felgyorsíthatja az oldalak betöltődését, ami kisebb sávszélesség fogyasztással is jár.

- **Beépülő modulok**

Az ugyancsak a *Bővítmények* menüből megnyitható Beépülő modul-kezelő felépítése hasonló a modulkezelőjéhez. Itt engedélyezhetjük, vagy tilthatjuk le a használatukat, illetve valamelyikre rákattintva testre szabhatjuk annak paramétereit.

- **Sablonok**

Egy sablon megtervezéséhez ismernünk kell a Joomla!-sablonok szerkezetét. Az alap-csomag tartalmaz 3 sablont, de ha valami különlegesre vágyunk, akkor az interneten is találunk néhány lelőhelyet, ahonnan ingyenes sablonokat tölthetünk le, valamint működik vagy 20 sablonklub, amelyekben 50 dollártól kezdődő tagsági díj fejében kiváló minőségű, sok modulpozíciót tartalmazó, keresőbarát, XHTML- és CSS-érvényes sablonokhoz juthatunk hozzá.

A sablonokat ugyancsak a *Bővítmenykezelőben* telepíthetjük. Sikeres telepítés után a *Bővítmények* menü *Sablonkezelő* pontjából megnyitható képernyőn válasszuk ki, majd az eszköztáron kattintsunk az *Alapértelmezett* gombra. Ha ezután felkeressük a felhasználói oldalt, az már az új színekben fog pompázni. Ne lepődjünk meg, ha némely modul nem ott van, mint ahol eddig volt, mert a sablonokban eltérő helyre kerülhetnek a modulpozíciók. A legtöbb sablonhoz szoktak mellékelni egy modul-térképet, ami az illető sablon modulpozíciót mutatja.

- **Nyelvi kiegészítők**

Ha a *Bővítmények* menüben a *Nyelvkezelő* pontot választjuk, akkor egy hasonnevű képernyő tárul elénk. Itt állíthatjuk be a felhasználói oldal és az adminisztráció kezelőfelületének

nyelvét. Ha esetleg az eredeti angol kiadáshoz telepítjük a magyar kiegészítőt, akkor a modulkezelőben a modulnevek, valamint a menüpontok angolul maradnak, nekünk kell lefordítanunk manuálisan, ugyanis ezek nem a kezelőfelülethez, hanem a tartalomhoz tartoznak, aminek lefordításához külső komponens szükséges.

10.3.3.5. Bővítmények eltávolítása

Ha egy bővítésre már nincs szükségünk, akkor nyissuk meg a *Bővítéskezelőt* (ahol a bővítést telepítettük). Váltunk át a bővítéstípusnak megfelelő lapra. A betöltődő oldalon jelöljük ki a bővítést, majd az eszköztáron kattintsunk az *Eltávolítás* gombra.

Nem minden összetevő dobja el a folyamat során a tábláit az adatbázisból, ezért ellenőrizzük az adatbázis-kezelőnkben, s ha már valóban nincs szükségünk a tárolt adatokra, távolítsuk el manuálisan ezeket.

11. SportRiadó – a legfrissebb hírek egy helyen

11.1. Rövid leírás

Ebben a fejezetben bemutatjuk a munkánk során fejlesztett weboldalt. A "SportRiadó" egy olyan hírportál, amely összegyűjti a legnagyobb magyar és külföldi sportoldalak legfrissebb híreit, lehetővé téve a látogató számára, hogy egy helyen értesüljön a sportvilág legaktuálisabb történéseiről.



11.1. ábra – SportRiadó főoldal

11.2. Tervezési szempontok

Munkánk során igyekeztünk a legfontosabb webdesign technikákat alkalmazni, különös figyelmet szentelve a felhasználóbarát kialakításnak és a keresőoptimalizálásnak.

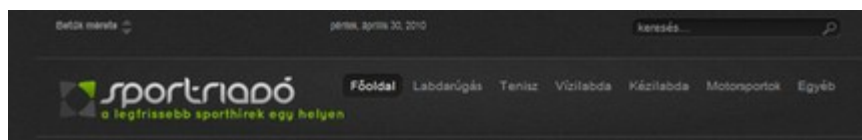
11.2.1. Felhasználóbarát tervezés

Az oldal létrehozásakor odafigyeltünk arra, hogy egyszerű legyen, valamint nagyon világos információ-architektúrával és megfelelő navigációs eszközzel rendelkezzen.

Fő hangsúlyt fektettünk a weblap főoldalára, mivel ez nem más, mint az egész webhely külső arculata, mivel a látogató ezzel találkozik legelőször, amikor a weboldalunkat látogatja.

Egy jó főoldalnak két fontos jellemzője van: az egyik az, hogy a felhasználó tudja, hol van, a másik pedig, hogy azonnal tudomást szerezzen róla, miről szól az oldal. Az oldalunk logójában szereplő szlogen ennek teljesen eleget tesz, a látogatóban rögvest tudatosul, hogy a legfrissebb sporthírek birtokába juthat.

A főoldal három vízszintes blokkra tagolódik. A legfelső egy úgynevezett eszköztárat biztosít a felhasználó számára, ami magában foglalja a navigációs menüt, továbbá egy kereső modult és egy olyan funkciót, amivel az oldalon szereplő betűk méretét lehet csökkenteni, illetve növelni. Ez utóbbi egy nagyon fontos tényező, ugyanis vannak olyan felhasználók, akik a kisebb betűméretet preferálják, míg mások a nagyobbakat, ennek különböző magyarázatai lehetnek, például a csökkent látás.



11.2. ábra – Legfelső blokk

A középső blokk a legnagyobb magyar sporthírpportálok legfrissebb híreit gyűjti össze, külön alblokkokba csoportosítva az oldalak szerint.



11.3. ábra – Középső blokk

A legalsó blokk az előzőhöz hasonló módon épül fel, azzal a különbséggel, hogy itt a külföldi sportoldalak hírei jelennek meg.



11.4. ábra – Legalsó blokk

A jobb átláthatóság és olvashatóság kedvéért minden blokk különböző háttérszínnel párosul. Mivel elsősorban a magyar nyelvű hírekre koncentráltunk, ezért a középső blokkon fehér alapon fekete betűket használtunk, ugyanis ez a legjobb választás olvashatóság szempontjából.

A tervezésnél négy alapszínt alkalmaztunk, különböző árnyalatokban amelyek mindegyike önálló funkcióval bír és a hátterek vonatkozásában is egyszínű háttereket alakítottunk ki, hogy ne keltsünk zavaró hatást a látogatóban.

A felbontás független tervezés technikájával élve nem kötött képszerűséget alkalmaztunk, mivel nem tudhatjuk, hogy a felhasználó milyen megjelenítési eszközt használ, továbbá az oldal designja nem változik meg a különböző betűméretek mellett sem.

Az oldalak tömörek, mégis könnyen áttekinthetők és nagy információ tartalommal bírnak. A felhasználók nagyon kicsi hányada olvassa el a hosszú, összefüggő szövegeket, mert ha valamit keresnek, akkor azt gyorsan akarják megtalálni. Ennek érdekében pontokba szedett felsorolásokat alkalmaztunk, továbbá a szöveg balra rendezésével az olvasó is könnyebben tudja átfutni a tartalmat, mint ha középre, vagy jobbra igazítottuk volna.

Igyekeztünk kihasználni a piros szín kiemelő hatását, ezért a különböző hírblokkok fejléceinek szövegét ezzel tarkítottuk.



11.5. ábra - Hírblokk

A tervezésnél törekedtünk elkerülni a leggyakoribb design hibákat, ezért nem használtunk Flash intrót, nem lép fel böngésző inkompatibilitás, a látogató böngészőablaka nem méreteződik át és nincs is szüksége vízszintes görgetésre. A kialakítás olyannyira egyszerű, hogy bármely pontból visszajuthatunk a főoldalra, tegyük ezt a logóra kattintással, esetleg a menü "Főoldal" elemére kattintva.

11.2.2. Keresőoptimalizálás

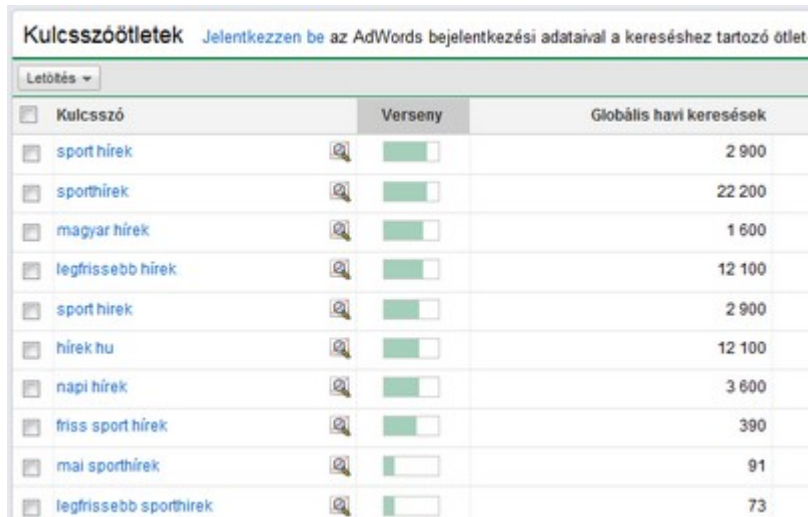
Ugyan az oldal témája egyedi ötlet volt, mégis alaposan igyekeztünk foglalkozni az esetleges konkurens elemzésével. A választás mégis jónak bizonyult, ugyanis a legtöbb hasonló jellegű magyar hírgyűjtő portál nem megfelelő kialakítású, átláthatatlan és reklámokkal zsúfolt.

A konkurens oldalak szóhasználatát elemezve megtaláltuk a legtöbbet keresett keresőszavakat, ezért tudtuk, hogy mire érdemes és mire nem optimalizálni az oldalt. Ezek az információk hozzájárultak ahhoz, hogy az oldalunk hatékonyabbá váljon.

A megfelelő kulcsszavak, kifejezések megtalálásának igyekeztünk nem fejetlenül nekimenni, mert hiába jó a weboldal bizonyos szavakra a találati listán, ha azokra az emberek nem keresnek, nem hoznak látogatókat.

Másrésről az sem lett volna szerencsés, ha olyan szavakra próbálunk optimalizálni, melyekre sokan keresnek ugyan, de olyan nagy a konkurencia, hogy nincs reális esély a jó eredményre. Célunk 4-5 olyan szó, kifejezés kiválasztása volt, melyre érdemes optimalizálni.

Ebben nagy segítségünkre volt a Google AdWords Keyword Tool nevű segédeszköze, amelynek legújabb verziója már azt az információt is tartalmazza a kiválasztott kulcsszavakról, hogy azokra milyen nagy a konkurencia és átlagban hányan keresnek rá. Sikerült 5 olyan kifejezést találnunk, amelyek esetében van esélyünk labdába rúgni, így sikeresen teljesíthetjük a keresőoptimalizálás ezen lépését. A következő kifejezéseket választottuk: mai sporthírek, legfrissebb sporthírek, sport oldal, friss sport, magyar sport.



Kulcsszó	Verseny	Globális havi keresések
sport hírek	<input checked="" type="checkbox"/>	2 900
sporthírek	<input checked="" type="checkbox"/>	22 200
magyar hírek	<input checked="" type="checkbox"/>	1 600
legfrissebb hírek	<input checked="" type="checkbox"/>	12 100
sport hírek	<input checked="" type="checkbox"/>	2 900
hírek hu	<input checked="" type="checkbox"/>	12 100
napi hírek	<input checked="" type="checkbox"/>	3 600
friss sport hírek	<input checked="" type="checkbox"/>	390
mai sporthírek	<input type="checkbox"/>	91
legfrissebb sporthírek	<input type="checkbox"/>	73

11.6. ábra – Google AdWords Keyword Tool

Az oldallal igyekeztünk értékes információkhoz juttatni a jövőbeli látogatókat, mivel minden oldal kialakításánál ez az egyik legfontosabb szempont.

Mivel az oldalunkon elhelyezett szöveg jól strukturált, valamint a webhely minden oldalára mutat teljes értékű szöveges link, már így jó esélyünk van arra, hogy a látogatottság elfogadható mértékű legyen.

Mivel leginkább a szabadszavas keresők felől várunk forgalmat, ezért a szöveges tartalomra kellett helyeznünk a hangsúlyt, ugyanis a keresők szöveges oldalak között keresnek, nem pedig multimédiás alkalmazások között. Így többek között kerültük az interaktív Flash és Java alkalmazások felhasználását azon okból, hogy ezek számos problémát szülnek. Ha egy oldal összes funkciója egy Flash, vagy Java fájlban található, akkor gyakorlatilag egy üres oldalnak tekinthető, amelybe az összes információ a keresők által nehezen, vagy egyáltalán nem feldolgozható objektum formájában van beágyazva. Egy HTML oldallal sokkal több látogatót lehet vonzani.

Mivel a Joomla! tartalomkezelő rendszerrel írtuk meg oldalunkat, csak a kulcsszavak kiválasztása és elhelyezése, webhelyünk strukturálása, valamint az oldal címének (TITLE) és leírásának (Description Meta tagok) megfelelő megadása volt a feladatunk.

Metaadatok beállításai

A hely globális meta leírása	A legfrissebb sporthírek egy helyen
A hely globális meta kulcsszavai	mai sporthírek, legfrissebb sporthírek, sport oldal, friss sport, magyar sport
A Cím metaadat megjelenítése	<input type="radio"/> Nem <input checked="" type="radio"/> Igen
A Szerző metaadat megjelenítése	<input type="radio"/> Nem <input checked="" type="radio"/> Igen

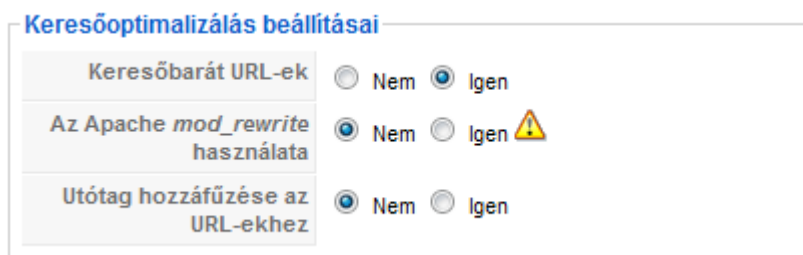
11.7. ábra – Joomla! metaadatok beállításai

A tartalom rendszeres frissítését automatikusan biztosítottuk, mivel folyamatosan frissülő RSS csatornákon keresztül érkező adatokat dolgozunk fel, amivel mindig a legaktuálisabb információkat szolgáltatjuk a látogatóinknak.

11.3. A tartalomkezelő rendszer és a kész oldal

Az oldal megírását a Joomla! tartalomkezelő rendszerrel végeztük el. Az egyik legnagyobb sablonklub, a RocketTheme.com egy ingyenes sablonját (Quasar) dolgoztuk át, ezzel is biztosítva magunkat, hogy a site megfelel majd minden technikai szabványnak (CSS, XHTML, stb.).

Keresőoptimalizálás szempontjából is jó választásnak bizonyult a tartalomkezelő rendszer, mert támogatja a keresőbarát URL-eket segítségével az oldal értékes tartalommal való feltöltésére koncentrálhattunk, és élhattünk a dinamikus weboldalak minden előnyével.



11.8. ábra – Joomla! keresőoptimalizálás beállítások

Az oldal hírblokkjai RSS csatornákon keresztül olvassák be a különböző portálok legfrissebb híreit.

A főoldalon a legnagyobb sportportálok vegyes híreit helyeztük el, de lehetőséget biztosítunk a látogatóknak, hogy oldalanként friss híreket kapjanak a különböző sportágakról.

Ezeket a kategóriákat a fenti menü megfelelő pontjára való kattintással érhetjük el.

A leírtak függvényében úgy érezzük, az általunk elkészített weboldal eleget tesz a legalapvetőbb webdesign elvárásoknak, amelyek nélkül egy mai oldal nem tekinthető modernnek és elfogadhatónak.

Összefoglalás

„Nem hiszek egy olyan világban, ahol az ember mást se csinál, mint az interneten szörfözik, ahogy nem hiszek abban a világban sem, ahol csak autóval változtathatunk helyet.”

(Umberto Eco)

A szakdolgozat befejezéseként összegezni szeretnénk a kialakult eredményeket. A munkánk folyamán megpróbáltuk bemutatni a legalapvetőbb technológiákat és a leggyakrabban használt webdesign fogásokat, melyek nélkülözhetetlenek egy modern weblap elkészítéséhez.

A kezdetekben ugyan különböző álláspontokat képviseltünk a témával kapcsolatban, de úgy érezzük, hogy ezek az eltérő nézőpontok a közös munka folyamán fokozatosan közeledtek egymáshoz, mi több sikerült mindkettőnk elképzeléseit egymás mellé illeszteni.

A kezdeti nehézségek ellenére mindketten úgy érezzük, hogy kifejezetten hasznunkra vált a közös projektmunka, aminek a jövőben is nagy hasznát vehetjük majd. A vázlat elkészítésekor pontos célokat tűztünk ki magunk elé mind a szakdolgozat tartalmával, mind a később megvalósított weboldallal kapcsolatban. Ennek érdekében ütemeztük munkánkat és osztottuk fel egymás között a kidolgozandó fejezeteket.

A legnagyobb probléma, amivel szembesültünk egy saját domain-név lefoglalása volt, amit sajnos nem sikerült bejegyeztetnünk.

Amennyiben időnk, valamint lehetőségünk adódik arra, hogy a választott témában további kutatásokat végezzünk mindenképpen megteesszük, mivel nagy örömmel lettünk mindabban, amit megvalósítottunk.

Az elkészített weblap online publikálása is a jövőbeli céljaink között szerepel, mint ahogy az is, hogy akár a leglátogatottabb magyar nyelvű sporthírgyűjtő portált fejlesszük ki belőle.

Irodalomjegyzék

- [1] Ludik Péter - Ósz Rita – Váraljai Mariann: Web-design és animáció
Dunaújvárosi Főiskolai Kiadó, 2008;
- [2] Jakob Nielsen: Web-design, Typotex Kft. Elektronikus Kiadó, 2002;
- [3] Virginia DeBolt: HTML és CSS, weblapszerkesztés stílusosan, Kiskapu kft, 2005;
- [4] Sikos László: JavaScript 1.5 kliens oldalon, BBS-INFO Kiadó, 2004;
- [5] Mark Norman Francis: A HTML alapjai,
<http://dev.opera.com/articles/view/12-a-html-alapjai/>
- [6] Chris Heilmann: A HTML <head> eleme,
<http://dev.opera.com/articles/view/13-a-html-head-eleme/>
- [7] Nagy Gusztáv: Web programozás I. 0.7. verzió, 2008,
http://nagygusztav.hu/sites/default/files/Web_programozas_jegyzet_0.7_0.pdf
- [8] Mark Wilton-Joens: JavaScript history,
<http://www.howtocreate.co.uk/jshistory.html>
- [9] Proszenyák Zoltán: Miért kockázatos flash honlapot készíttetni?,
<http://www.profitguru.hu/blog/index.php/2007/12/15/miert-kockazatos-flash-honlapot-keszittetni/>
- [10] Vajdasági ismeretterjesztő és tudománynépszerűsítő portál,
<http://www.fokusz.info/index.php?cid=1173942465&sid=1981012301>
- [11] Eric A. Meyer: CSS zsebkönyv, Webtartalmak formázása stíluslapokkal
- [12] Steve Krug: Ne törd a fejem! Felhasználóbarát webdizájn
- [13] Steven Holzner - Nancy Conner: Joomla! for dummies
- [14] Brad Callen: Search Engine Optimization Made Easy
- [15] Stefan Weinenberger: A sikeres keresőoptimalizálás weboldala,
<http://keresooptimalizalas.co.hu>
- [16] Weblabor: Keresőoptimalizálás ,
<http://weblabor.hu/cikkek/keresooptimalizalas>
- [17] Jároli József: Tartalomkezelő rendszer ,
http://webni.innen.hu/Tartalomkezel_c5_91Rendszer
- [18] Jároli József: Keresőoptimalizálás ,
http://webni.innen.hu/Keres_c5_91optimaliz_c3_a11_c3_a1s