Informatika 6.

Web fejlesztés

Nagy , Gábor

Informatika 6. : Web fejlesztés

Nagy , Gábor
Lektor : Cseri , Tamás

Ez a modul a TÁMOP - 4.1.2-08/1/A-2009-0027 „Tananyagfejlesztéssel a GEO-ért” projekt keretében készült. A projektet az Európai Unió és a Magyar Állam 44 706 488 Ft összegben támogatta.

v 1.0

Publication date 2010

Kivonat

A modul bemutatja a web fejlesztéshez szükséges alapvető szabványokat, technológiákat.

Tartalom

6. fejezet - Web fejlesztés

1. 6.1 Bevezetés, alapfogalmak

Amikor „webes felületről” vagy „web alapú” alkalmazásokról beszélünk, több egymással szorosan összefüggő technológiára gondolunk, amelyek a szokásos internetes böngészésen túlmenően számos más informatikai alkalmazásban is kulcsszerepet játszanak. Elterjedtségüket jól mutatja, hogy sokszor összekeverik az Internet és a web fogalmát. Még más hálózati alkalmazásokat (pl. az e-mail vagy az azonnali üzenetküldés) is jelentős részben webes felületen keresztül használnak, és általánosnak mondható, hogy hálózati nyomtatókat vagy egyéb hálózati eszközöket webes felületen keresztül tudunk beállítani.

Tipikus alkalmazási esetben a böngészőprogram http vagy https protokoll segítségével tölt le fájlokat, egy HTML állományt és az abba beágyazott tartalmakat, egy webszerverről. A böngésző megjeleníti a letöltött lapot és a beágyazott objektumait, a tartalmat az oldalba ágyazott programok módosíthatják, ehhez a szerverrel további kommunikációt folytathatnak http protokollon keresztül. Egyes beágyazott tartalmak szintén képesek programok futtatására hasonló módon.

A HTML állományba beágyazott tartalmak lehetnek képek, hangok, mozgóképek állományai vagy a megjelenítésükhöz és kezelésükhöz valamilyen kiegészítő program telepítését igénylő objektumok.

A webes eszközkészletet alkotó technológiákat egymástól külön is lehet alkalmazni. A megjelenítendő HTML állományok lehetnek a helyi számítógépen, például egy programnak az offline is olvasható kézikönyvét biztosítva. A beágyazott tartalmakra kidolgozott formátumokat is széles körben alkalmazzák egyéb célokra.

Előszeretettel használják a http protokollt weboldalak elérése mellett más adatátviteli feladatokra is. Ennek legfontosabb okai, hogy egy adatátvitelre egyszerűen használható és jól bevált szabványról van szó, valamint hogy a http és https adatforgalom és a jellemzően hozzájuk tartozó hálózati portok általában még szigorúbb tűzfalszabályok mellett is gond nélkül használhatóak.

A HTML-t, a http-t és még egy sor további, a webhez köthető szabványt egy direkt erre a célra létrehozott szervervezet a World Wide Web Consortium, vagy röviden W3C adja ki.

2. 6.2 Az HTML nyelv

2.1. 6.2.1 A HTML állományok felépítése

A HTML (HyperText Markup Language) egy weboldalak tartalmának leírására kidolgozott SGML alapú jelölőnyelv. Tartalmazza a weblap szövegét és az annak a megjelenítésére vonatkozó különféle utasításokat, de csak hivatkozásokat tartalmaz olyan külön állományban elhelyezkedő beágyazott tartalmakra, mint például a képek.

A HTML állomány egy dokumentum típus definícióval kezdődik, ami leírja a dokumentum típusát és az alkalmazandó DTD állományt. Ez a következőképp nézhet ki:

<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01//EN" "http://www.w3.org/TR/html4/strict.dtd">

Ezt követően az állomány tartalma a <html> és </html> címkék között helyezkedik el, és két részre bontható: a fejrészre (head) és a megjelenítendő tartalmat leíró törzsre (body).

A dokumentum további tartalmára is jellemző lesz a címkék használata, melyek állhatnak önállóan vagy záró párjukkal együtt közrefoghatják a dokumentum egy részét, amit elemnek nevezünk. A HTML-ben előfordulhatnak a címkék önállóan is, ilyen lesz például a sortörést eredményező
, vagy a képek beszúrását lehetővé tévő <img>. A nyitó címke attribútumokat is tartalmazhat név vagy név=érték formában.

2.2. 6.2.2 A fejrész

A HTML állomány fejrésze, a <head> elem többféle, közvetlenül nem megjelenítendő dolgot tartalmazhat. Ilyen a dokumentum megnevezése, amit a <title> elem tartalmaz. Az ebben meghatározott szöveg általában a böngésző ablak fejlécén vagy az oldalhoz tartozó fülön jelenik meg.

A fejléc tartalmazhat <meta> elemeket, amelyek metaadatok megadására használhatóak a name és a content attribútumaik segítségével. Tipikus metaadatok a leírás (name=”description”), a kulcsszavak (name=”keywords”) vagy a szerző (name=”author”).

Elhelyezkedhetnek még a fejlécben stíluslapok, szkriptek, illetve ezeket tartalmazó külső állományokra történő hivatkozások.

2.3. 6.2.3 Tartalom meghatározása, formázás

A HTML állomány törzse tartalmazza a megjelenítendő tartalom leírását, ami a szöveg mellett az annak megjelenítésére vonatkozó utasításokat is jelenti.

A HTML állomány fej (head) részében többféle karakterkódolást használatát is be tudjuk állítani. Azokat a karaktereket, amelyeket az adott kódlap nem tartalmaz vagy a HTML nyelvben betöltött szerepük miatt nem használhatóak közvetlenül, karakter egyedhivatkozások segítségével használhatjuk. Ezek a hivatkozások „&” és „;” karakterek között tartalmazzák a karakter megnevezését (az SGML entitás nevét). Az „á” betűt például a „á” formában lehet ilyen módon megadni. Használhatunk numerikus karakter referenciát is, ilyenkor „&#” és „;” között kell megadni a kívánt karakter Unicode kódját decimálisan vagy hexadecimálisan. Az „á” betűt így a „á” vagy a „á” alakban is meg tudjuk adni.

A <p> elem a szöveg egy bekezdését tartalmazza. A szövegben a
 elem segítségével tudunk tetszőleges helyen sortörést elhelyezni. (A HTML kódban található sortörések nem eredményeznek sortörést a megjelenített szövegben)

A szöveg tartalmazhat különböző szintű címeket, amiket a <h1>, <h2>, ... <h6> elemekben tudunk elhelyezni, a szintbéli rendűségnek megfelelően. A fejezetek címeit például a <h1>-be, az alfejezetekét pedig a <h2>-be.

Több okból is célszerű törekedni a tartalom és a formázás szétválasztására. A HTML és a hozzá kapcsolódó CSS ezt az elvet teljes mértékben támogatja, de találkozhatunk még a formázásra vonatkozó utasítások közvetlen megadásával, ami például a <b>, <i> vagy a <font> címkék segítségével történhet.

A elem a dokumentum egy részét a elem pedig a szövegnek egy (bekezdésen belüli) szakaszát tartalmazhatja. Mindkét elem alkalmazásakor fontos a más elemeknél is használható „class” attribútum megadása, amivel a dokumentum adott részének az osztályát lehet meghatározni.

A CSS (Cascading Style Sheet) egy stílusleíró nyelv, a segítségével meghatározható, hogy a weboldal különféle részei milyen formázási jellemzőkkel jelenjenek meg. A nyelv felépítése egyszerűnek mondható, érvényességi kört meghatározó szelektorokból és az azokat követő deklarációs szakaszokból áll. A kapcsos zárójelek között megadott deklarációs szakaszt pontosvesszővel elválasztott deklarációk alkotják, melyek a szelektor segítségével meghatározott rész egy tulajdonságát adják meg.

A szelektoroknál fel lehet használni a class attribútum segítségével megadható osztályt, így az egyes elemekből többféle stílusú is lehet az oldalon. A szelektorban hivatkozhatunk olyan jellemzőkre is, mint például egy link volt-e már látogatva, vagy hogy éppen a dokumentum adott része felett tartózkodik-e az egérkurzor.

A CSS kódot el lehet helyezni közvetlenül a fejrész <style> elemében, vagy hivatkozhatunk ugyanott egy külön állományban elhelyezett kódra a <link> elem segítségével.

2.4. 6.2.4 Felsorolások, táblázatok

A HTML dokumentum többféle felsorolást is tartalmazhat. A számozott felsorolásokat a <ol>, a számozás nélkülieket a <ul> elem tartalmazza. Ezeken belül a felsorolt tételeket <li> elemek adják meg.

Táblázatot a <table> elem segítségével adhatunk meg, amin belül a táblázat sorai <tr> elemekbe kerülnek. A sorokon belül az egyes cellákat <td>, vagy fejléc típusú cellák esetében a <th> elemek tartalmazzák.

A táblázatokat egy időben előszeretettel felhasználták a weblap szerkezetének kialakítására is. Napjainkban ez egyértelműen kerülendőnek minősül.

2.5. 6.2.5 Linkek, beagyazott tartalmak

A HTML egyik legfontosabb tulajdonsága, hogy az oldalakon hivatkozások helyezhetőek el. Ezek segítségével az oldal egy része (tipikusan egy rövidebb szöveg) egy másik oldalra vezető ugróponttá tehető.

Az <a> elem tartalma a weboldalnak az a része, amelyik az ugrópont felületét szolgáltatja, a „href” attribútuma pedig megadja az ugrópont célját. A hivatkozás lehet abszolút vagy relatív. Az abszolút hivatkozás a protokoll megadásával kezdődően tartalmazza a cél teljes címét. A relatív hivatkozás a HTML állományhoz (vagy a fejrész <base> elemében meghatározott útvonalhoz) képest adja meg a célt.

Képet elhelyezni a HTML állományban az <img> elemmel tudunk. A szöveg kérdéses pontjára az elem „src” attribútumában megadott elérési útvonalú képfájlban tárolt kép kerül. Az elérési útvonal ugyanúgy lehet abszolút vagy relatív, mint a linkek esetében.

A kép formátuma többféle lehet, a hatékonyság érdekében érdemes tömörített formátumokat használni. Veszteségmentes tömörítésnél a .gif vagy a .png, veszteséges tömörítésnél pedig a .jpg használata terjedt el.

Képek mellett a weboldalon elhelyezhetünk még kiegészítő program segítségével kezelhető objektumokat is. Ezek például növelhetik az oldal interaktivitását, vagy a grafikus megjelenítés és mozgóképek lejátszása terén nyújthatnak új lehetőségeket.

A napjainkban leginkább jelentős beépülő tartalom a Flash. Az eredetileg a Macromedia által fejlesztett eszköz fejlesztése felvásárlás következtében az Adobe-hoz került. A legtöbb webes felületen videólejátszást lehetővé tévő vagy interaktívabb felületet nyújtó (pl. nagyon sok játék esetében) weboldal beépülő Flash tartalmak segítségével működik, de sokszor használják webes reklámokhoz is.

A Microsoft kidolgozott egy .NET alapú, a Flash-hez hasonló szolgáltatásokat nyújtó eszközt, a Silverlight-ot. Az ehhez készült tartalmakat a Moonlight nevű nyílt forráskódú eszköz segítségével is lehet használni.

Inkább régebben volt népszerű a Java applet, ami nem keverendő össze a JavaScript-tel. Ez Java alapú beépülő tartalmak elhelyezését teszi lehetővé.

2.6. 6.2.6 Űrlapok

Az űrlapok segítségével nyílik lehetőség arra, ahogy egy weboldalt adatbevitelre használjunk. Az oldalon elhelyezhető egy vagy több űrlap a <form> elem használatával, amelyen belül a szokásos HTML tartalmon túl adatbeviteli feladatokat ellátó elemeket is meghatározhatunk.

A legtöbb ilyen beviteli mezőt az <input> elem segítségével lehet létrehozni. A beviteli mező típusát a „type” attribútum határozza meg. Ennek függvényében az űrlapon el tudunk helyezni szövegbeviteli mezőket (type=”text”), jelszavak bevitelére használható mezőket (type=”password”) vagy nyomógombokat (type=”button”). Lehetőség van összetettebb elemek, mint például jelölő négyzetek (type=”checkbox”) és választógombok (type=”radio”) vagy vagy akár fájlok feltöltését lehetővé tévő mezők (type=”file”) használatára is.

Tartalmazhatnak az űrlapok több soros szöveg bevitelére alkalmas mezőket, amelyeket a <textarea> elem segítségével lehet elhelyezni. Az elem tartalma a szöveg kezdeti értéke.

Listákat a <select> elem segítségével lehet elhelyezni. Az elemen belül az <option> elemmel lehet meghatározni a lista pontjait, amelyek közül a felhasználó majd választhat.

2.7. 6.2.7 XHTML

Az XML az SGML egy szűkített, egyszerűsített részhalmaza. Szintén egy általános célú leírónyelv, de egyszerűségének köszönhetően könnyebb hozzá feldolgozó programokat fejleszteni, aminek következtében nagy népszerűségre tett szert.

Az XML alapú formátumok terjedésével felmerült az ötlet, hogy a HTML nyelvet is át lehetne úgy alakítani, hogy megfeleljen az XML előírásainak. Így született meg az XHTML formátum, ami nagyban hasonlít a HTML-re, csak az XML szabványnak való megfelelés érdekében tér el valamennyire.

Az XHTML-ben az egymásba ágyazott elemeknek egy szigorú hierarchiát kell meghatároznia. Minden elemnek zártnak kell lennie, ezért minden címkéhez tartoznia kell egy lezáró párnak, vagy önmagában zártnak kell lennie. (Ezért kell a
 helyett a
 címkét használni a sortöréshez.)

Lezáró címkével kell rendelkezniük azoknak az elemeknek is, amelyeknek a végét a HTML-ben a következő hasonló szintű nyitó címke is jelölheti. Használni kell ezért a </tr> és </td> címkéket a táblázatok sorainak és celláinak, valamint a címkéket a felsorolások elemeinek a lezárására, vagy a </p> címkét a bekezdések végén.

Az XHTML állomány nem tartalmazhat átlapolást. Ha egy szövegrészt egy szakaszon félkövéren, utána fékövéren és dőlten, majd simán csak dőlten akarunk megjeleníteni, akkor a HTML szabvány szerint alkalmazhatjuk a következő megoldást:

normálfélkövérfélkövér és dőltdőltismét normál

Az XHTML-ben az átlapolás tiltása miatt csak a következő megoldások helyesek:

normálfélkövérfélkövér és dőltdőltismét normál

normálfélkövérfélkövér és dőltdőltismét normál

A HTML nyelv a címkék és az attribútumok kulcsszavaiban érzéketlen a kis- és nagybetűk közötti különbségre. Egyaránt használhatjuk például a <html>, a <HTML> vagy akár a <Html> alakot is. Az XHTML-ben csak csupa kis betűből álló kulcsszavakat lehet használni, mivel az XML érzékeny a kis- és nagybetűk eltérésére.

Az attribútumoknak mindig rendelkezniük kell egy egyenlőségjel után megadandó értékkel. Ezt az értéket mindig idézőjelek között kell megadni, akkor is ha nem szöveg a típusa.

2.8. 6.2.8 HTML5

A HTML5 a korábbi HTML szabványoknak egy jelentősen átgondolt és számos új lehetőséggel kibővített változata. Célja többek között, hogy egy weboldal a hozzá tartozó JavaScript programokkal együtt, minden további alkalmazás igénybevétele nélkül képes legyen olyan szolgáltatások megvalósítására, amelyeket jelenleg csak az Adobe által készített Flash vagy a Microsofthoz kötődő Silverlight bővítmények segítségével tudnak megoldani.

A HTML5 lehetőséget ad videók weboldalba ágyazására, JavaScriptből egyszerűen kezelhető grafikus felületek (canvas) alkalmazására, és több olyan a felhasználói felület interaktivitását fokozó lehetőségeket tartalmaz, mint például a fogd és vidd (drag and drop) műveletek támogatása.

A HTML5-ben végleges eltávolításra kerülnek azok az elemek, amelyeknek a használatát már a legutóbbi (HTML 4.01) verzióban sem ajánlották.

2.9. 6.2.9 Hírcsatornák

A különféle webes tartalmakat (portálok, blogok, stb.) rendszeresen olvasó felhasználó számára fontos, hogy kedvenc oldalait folyamatosan követni tudja, ha valahol egy új cikket vagy bejegyzést publikálnak, arról minél hamarabb értesüljön. Mindezt úgy szeretnénk megoldani, hogy az érintett weblapokat ne kelljen rendszeresen felkeresni. Különösen fontos ez, ha sok olyan oldalt követnénk figyelemmel, ahol csak viszonylag ritkán jelennek meg új tartalmak.

A web egyes pontjain megjelenő új tartalmak címének és szöveges kivonatának összegzésére és továbbítására többféle formátumot is kidolgoztak. A legelterjedtebbek az Atom és az RSS különféle verziói, az eltérő formátumok mindegyike XML alapú.

A hírcsatornákban közvetített tartalom a számítógépen futó (pl.: BlogBridge, FeedReader, NewsFire) vagy webes felületű (pl.: Google Reader, Netvibes) hírolvasók segítségével tekinthető meg. Léteznek alkalmazások a

hírcsatornák tartalmának egyesítésére és szűrésére is. Sok mobiltelefon is rendelkezik hírcsatorna olvasó alkalmazással.

A böngészők a hírcsatornák olvasását többféle módon is támogathatják. Megjeleníthetik a hírcsatorna tartalmát a böngészőablakban, vagy készíthetnek aktív könyvjelzőt a hírcsatorna bejegyzéseiből. Sok levelezőprogram szintén képes kezelni hírcsatornákat, híreiket mint beérkezett küldeményeket jelenítve meg.

A térbeli vonatkozással rendelkező hírek gyűjtésére kidolgozták a GeoRSS szabványt. Az RSS vagy akár Atom alapú üzenetek a hírhez kapcsolódó geometriai információt a szintén XML alapú GML (Geography Markup Language) segítségével írják le.

Amennyiben a hírcsatorna által közvetített tartalmak hangállományok (.mp3 vagy .ogg formátumú fájlok), akkor podcastról beszélünk. A podcastok használatát a személyi számítógépeken futtatható alkalmazások mellett a fejlettebb zenelejátszók is támogathatják.

3. 6.3 Webszerverek működésének alapelvei

3.1. 6.3.1 A http és a https protokollok

A http az alkalmazási réteg egy protokollja, amit a webes tartalmak (HTML oldalak és az azokhoz kapcsolódó egyéb adatok) letöltésére dolgoztak ki. A http kérésekkel mindig egy erőforráshoz férünk hozzá, amit az URI (Unified Resource Indentifier, egységes erőforrás azonosító) segítségével határozunk meg. Amennyiben az erőforrás meghatározása hely szerint történik, akkor URL-ről (Unified Resource Locator) beszélünk.

Az URL tartalmazza a protokoll megnevezését, a gép nevét vagy IP címét, opcionálisan a hálózati port számát, és az erőforrásnak a gépen belüli elhelyezkedését. Csatlakozhat még az URL-hez kérdőjel karaktert követően egy paraméterlista valamint kettős kereszt karaktert követően a hiperszöveg egy pontjának (pl. egy fejezet kezdetének) az azonosítója.

Többféle kéréssel fordulhatunk a http protokoll alapján egy erőforráshoz, a leggyakoribb a GET és a POST metódus alkalmazása. Ezekkel lekérhetjük az erőforrás tartalmát vagy adatokat küldhetünk az erőforrásnak.

A http mellett létezik még egy vele teljesen azonos feladatokat ellátó, de az adatforgalmat egy titkosított csatornán keresztül intéző protokoll, a https. Általában olyan helyeken használják, ahol kiemelten fontos a biztonság (pl. banki tranzakciók webes felületen történő intézése), mert a továbbított adatokat a hálózat közbenső elemein így nem tudják lehallgatni vagy manipulálni. A teljes biztonsághoz fontos, hogy az ilyen módon elért hely hitelesített kulccsal rendelkezzen, hogy közbeékelődéses támadásokra se legyen esély.

A böngészőprogramok általában a protokoll megnevezésén túl is jelzik valamilyen módon, hogy a https protokollt használjuk, és hogy ez hitelesített kulccsal történik-e. A Firefox például sárga háttérszínnel jeleníti meg a címet, ami előtt pedig egy zöld színű mező szolgáltat információkat a kulcs hitelességéről. Gyakran jelölik bezáródó lakat szimbólummal, ha https protokollt használunk a http helyett.

3.2. 6.3.2 Statikus és dinamikus webtartalmak

Az erőforrásnak az URL-ben meghatározott számítógépen belüli elhelyezkedése általában egy a webszerver fájlrendszerén található állományának feleltethető meg. Statikus tartalom esetében a GET kérés hatására ennek az állománynak a tartalmát fogja elküldeni szerver a kliens részére.

Dinamikus tartalmak esetében a szerveren elhelyezkedő állományban egy program található. Amikor a kliens le akarja kérdezni az erőforrást, a szerver futtatja a programot, majd a program által előállított adathalmazt fogja elküldeni a kliens részére.

A programnak paramétereket adhatunk át az URL-ben. Az erőforrás elérési útvonala után, egy kérdőjelet követően tudjuk megadni a paraméterlistát kulcsszó=érték alakú elemeit egymástól „&” karakterrel elválasztva. Az alap karakterkészletben nem található vagy speciális funkciót betöltő karaktereket a „%” karaktert követően megadott hexadecimális kódjukkal tudjuk elérni. Az ilyen módon megadott adatokhoz a szerveren futó program hozzáfér, azokat felhasználja a kimenet előállítása során.

A szerveren futó program hozzáférhet még az úgynevezett sütik (cookies) tartalmához is. A süti egy tetszőleges tartalmú, általában szöveges adatcsomag, amit a szerver a kért adatokkal együtt küld a kliens részére, az pedig a kérések során visszaküldi a szervernek. Elsősorban azért van a sütikre szükség, mert a http egy állapotmentes protokoll, így munkamenetek kialakítására csak ilyen kerülőúton nyílik lehetőség.

A munkamenetek biztosíthatják például, hogy egy portálra történő felhasználói bejelentkezés (felhasználónév és jelszó megadása) után tetszőleges ideig az adott felhasználó nevében tudjuk használni az adott portált. A sütik általában addig érvényesek, ameddig a böngészőprogram fut, de a szerver kérheti hosszabb időre történő tárolásukat is. (Erre a böngésző figyelmeztetni szokott.)

Az dinamikus tartalmak általában HTML kódok, de nincs semmi akadálya annak, hogy bármilyen más típusú adatot is ilyen módon állítsunk elő. Az eredmény lehet például egy képfájl, ami egy a paraméterek segítségével meghatározott elhelyezkedésű terület térképét tartalmazza.

3.3. 6.3.3 Programozási lehetőségek

A szerveren futó, a statikus webtartalmakat előállító program sokféle lehet. Az egyik lehetőség, hogy egy bináris programot használunk, amit a webszerverrel való kommunikálásra alkalmazott felület után CGI (Common Gateway Interface, általános átjáró felület) programnak szokás nevezni. A bináris programot sokféle programozási nyelven, sokféle fejlesztőeszköz segítségével elő lehet állítani.

Lefordított programok mellett interpreteres megoldásokat is használhatunk. Ilyenkor a webszerver egy értelmező segítségével dolgozza fel a valamilyen magasabb szintű nyelvben, sok esetben egy a statikus részeket tartalmazó HTML kódba beágyazva elhelyezett programot.

Népszerűek az ilyen feladatokra a kifejezetten erre a célra kidolgozott PHP és ASP nyelvek, de gyakran készülnek dinamikus oldalak Perl, Python, Ruby vagy Java nyelven is.

A webszerveren futó programok gyakran használnak az adatok tárolására valamilyen SQL alapú adatbázis-kezelő programot. Ezek a programok futhatnak a webszerverrel azonos számítógépen vagy bármilyen más helyen, amit a webszerveren futó programok elérnek.

3.4. 6.3.4 Portálmotorok

A webszerveren található dinamikus tartalmak általában egy összetett rendszert alkotnak. Különféle programok állítják elő egy portál különféle felületeit a kapott paraméterek és egyéb változó adatok alapján.

Hogy ne kelljen valamennyi alkalommal teljesen a kezdetektől indulva elkészíteni vagy akár csak egyesével összeválogatni és testre szabni a szükséges programokat amikor egy dinamikus webtartalmakra épülő portált akarunk kialakítani, előre összeállított és a megfelelő, a telepítést és a beállításokat is elvégző programokat is tartalmazó csomagokat, úgynevezett portálmotorokat készítenek.

A portálmotok legnépesebb családját a tartalomkezelő rendszerek alkotják, mint például Joomla! vagy a Drupal. Ezek egyszerűen felhasználhatóak egy szokásos webportál gyors kialakítására, valamint általában számos olyan kiegészítő modullal rendelkeznek, amelyek segítségével sokféle egyéb funkcióval bővíthetőek a szolgáltatásaik.

Vannak olyan portálmotorok is, amelyeket valamilyen konkrétabb feladatra fejlesztenek. Ilyenek például az e-Learning rendszerek alapját képező oktatási portálmotorok, mint például BlackBoard vagy az eGEO és a vGEO hátterében is működő Moodle. Léteznek még portálmotorok többek között webes felületű csoportmunka (pl. a phpGroupWare vagy az eGroupWare) vagy wiki oldalak szerkesztésének (pl. a Wikipedia hátterében is működő MediaWiki) támogatására is.

A portálmotorok sokszor többféle webszervert és többféle adatbázis-kezelő szervert is támogatnak. A portál megjelenése általában rugalmasan testreszabható.

4. 6.4 Kliens oldali programok

4.1. 6.4.1 Javascript

A JavaScriptek egy a weblapokba beágyazott, a böngésző által a kliens oldalon végrehajtott programok. Nem keverendők össze sem a Java programozási nyelvvel és az ahhoz kapcsolódó eszközökkel, sem pedig az azon alapuló, a weblapokba ágyazható Java appletekkel!

Első változata 1995-ben, LiveScript néven, a Netscape Navigator böngészőben jelent meg. 1996-ban a Microsoft Internet Explorer 3.0 is átvette JScript néven. Az egységesség érdekében később az ECMA (European Computer Manufacturers Association) ECMA Script néven szabványosította. A napjaink böngészőiben JavaScript néven implementált megoldás többé-kevésbé ezt a szabványt követi. A név arra utal, hogy szintaktikájában közel van a Java-hoz, ami elsősorban annak tudható be, hogy mind a Java, mind pedig a JavaScript létrehozásakor a C és a C++ nyelvek szintaktikáját vették alapul.

A JavaScript végrehajtása az azt tartalmazó HTML kódot megjelenítő böngésző feladata. A JavaScript program tetszőlegesen tudja módosítani az oldal tartalmát, újabb oldalakat nyithat meg vagy felugró ablakokban kérdéseket és üzeneteket intézhet a felhasználóhoz, de ezen kívül máshoz nem férhet hozzá, hiszen az nagyon komoly biztonsági kockázatokat jelentene. Gondoljunk csak bele, milyen lenne, ha akármelyik általunk meglátogatott weboldal üzemeltetői hozzáférhetnének a számítógépünkön tárolt fájlokhoz az oldalban elhelyezett JavaScriptek által.

Bár az előbb említett okokból a JavaScriptek futtatása egy elszigetelt környezetben történik, használata még így is kockázatokkal és kellemetlenségekkel járhat. Nincs akadálya annak, hogy egy oldal a benne futó JavaScript segítségével idegesítően sok felugró ablakot nyisson, de akár a böngésző egy hibáját kihasználva megkísérelhet olyan dolgokhoz is hozzáférni, amelyekhez nem lehetne. A JavaScriptek futtatását ezért sok esetben korlátozzák, teljesen letiltják vagy csak meghatározott oldalak esetében engedélyezik.

Nem minden esetben lehetséges az oldalban elhelyezett JavaScriptek végrehajtása, a biztonsági okokból történő tiltás mellett elképzelhető még, hogy a HTML kódot megjelenítő eszköz nem is képes erre. Az oldalakat ezért célszerű úgy megtervezni, hogy JavaScript nélkül is teljes körűen használhatóak legyenek, vagy legalább csak azok a dolgok ne legyenek elérhetőek, amelyeket más módon nem lehetne megoldani.

4.2. 6.4.2 AJAX

A JavaScript alkalmazásában rejlő egyik lehetőség az oldalak használhatóságának javítása, a felhasználói interakciók válaszidejének csökkentése. Ha például szeretnénk egy menüpontra kattintva kibontani az almenü elemeit, azt JavaScript nélkül csak úgy tudjuk megoldani, hogy a böngésző a menüponthoz rendelt link segítségével újból lekéri a szerverről a teljes oldalt a kibontott menüponttal. Mivel egy JavaScript módosíthatja az oldal tartalmát, a menüpont kibontását is azonnal, a teljes oldal ismételt letöltése és az ahhoz kapcsolódó válaszidők kivárása nélkül meg tudja oldani.

Elképzelhető, hogy a kívánt változtatáshoz szükségünk van bizonyos, a kliens által még nem lekért adatokra (például egy többszálú hozzászólásokat tartalmazó oldalnál egy szál kifejtéséhez az azt alkotó hozzászólások adataira), vagy hogy a változtatást meghatározó adatokat szükséges valamilyen módon a szerverrel is közölni (például a hozzászólásunk szövegét, hogy az másoknál is megjelenhessen). Az ilyen feladatok megoldására a JavaScript programok képesek az XMLHttpRequest (XHR) felületen keresztül XML tartalmú üzeneteket küldeni és fogadni a http vagy https protokollokon keresztül. Az így felépülő technológiát AJAX-nak (Asynchronous JavaScript and XML) szokták nevezni.

A weboldalak felületének kialakításakor az AJAX technológia használata napjainkban általánosnak mondható.

Az egységesség érdekében javaslom öszefoglaló és ellenőrző kérdések fejezetek elkészítését. Ugyanakkor javaslom a dokumentum rövidítését, mert túl sok általános leírást tartalmaz a téma fontosságához képest. A dokumentumból hiányoznak a példák is, amelyek fontosak a téma megértéséhez. Sok helyen hiányzik az értelmzés, ami fontos lenne a témakör megértéséhez pl. mi az RSS, miért és mire használjuk.

Irodalomjegyzék

http://www.w3.org/html/

http://www.w3.org/standards/webdesign/htmlcss

http://www.w3.org/Style/CSS/

http://www.w3.org/standards/webdesign/script

http://www.w3.org/standards/webdesign/graphics

http://www.w3.org/standards/webdesign/audiovideo

http://www.w3.org/TR/xhtml2/

http://dev.w3.org/html5/spec/spec.html

http://www.ecma-international.org/publications/standards/Ecma-262.htm

http://www.w3.org/TR/XMLHttpRequest/