Kiemelések Nagy Gusztáv tankönyveiből:

**6.1. Bevezetés**

**Mi a CSS?**

A CSS a *Cascading Style Sheets* rövidítése

A stílusok a HTML megjelenítési elemei és attribútumai helyett használhatók, azoknál

jóval több lehetőséget biztosítva

A stílusok meghatározzák, hogy hogyan jelenjenek meg vizuálisan a HTML elemek

A stíluslapok segítségével könnyen szét lehet választani az oldal tartalmát annak kinézetétől

(a dizájntól)

A stílusokat általában külön állományban tároljuk (.*css* kiterjesztéssel)

Külső stíluslapokkal gyorsítani tudjuk a munkavégzést

Több stílus is hatással lehet egy elem megjelenésére

a *World Wide Web Consortium*40 (W3C), egy non-profit, szabványokat alkotó

szervezet oldotta meg. A HTML 4.0-ás verziójával és a vele párhuzamosan fejlesztett

CSS segítségével létrejött egy jól használható eszközpáros a webfejlesztők részére.

Mára minden jelentősebb böngésző támogatja a CSS-t, bár a támogatottság mértékében

vannak eltérések.

**Lépcsős elrendezés**

Melyik stílus fog érvényesülni, ha több stílust is definiálunk ugyanahhoz a HTML elemhez?

A következő négy beállítás érvényesül egyre nagyobb prioritással (tehát ütközés esetén a

későbbi felülírja az előzőt).

1. a böngésző alapbeállítása

2. külső stíluslap

3. *head* elemben definiált stílus

4. soron belüli stílus

Tehát a soron belüli stílus a legmagasabb prioritású, tehát felülír minden alacsonyabb

szintű formázást.

A nyelvtan három elemet különböztet meg: kiválasztó, tulajdonság és érték:

kiválasztó {tulajdonság: érték}

A kiválasztó legegyszerűbb esetben egy HTML tag, a tulajdonság azt határozza meg,

hogy milyen jellemzőt akarunk módosítani, míg az érték a változást határozza meg. A tulajdonságot

és az értéket egy kettősponttal kell egymástól elválasztani, és a kettőt együtt

kapcsos zárójelbe tenni.

body {color: black}

Ha az érték több szóból áll, idézőjelbe kell tenni:

p {font-family: "sans serif"}

**7.1. Bevezetés a JavaScript nyelvbe**

Mi a JavaScript?

Szkriptnyelv (tehát értelmezett, a böngésző értelmezi)

Interaktivitást (működést) ad a HTML oldalhoz

HTML oldalba építhető, vagy attól jobban elszeparálható kód

Események kezelésére alkalmazható

A neve ellenére nincs szoros kapcsolatba a Java nyelvvel

C++ és Java szintaxisra alapoz

A szabvány leírását az ECMAScript specifikáció tartalmazza

**Kód beillesztése a HTML oldalba**

A HTML *script* tagja használható arra, hogy az oldalhoz JavaScript kódot kapcsoljunk. A

*script* tag tartalma közvetlenül nem jelenik meg a böngésző ablakában, hanem a böngésző

fogja azt futtatni. Például a következő kód a HTML aktuális helyére a *Hello Világ!*

szöveget jeleníti meg:

<body>

<script type="text/javascript">

document.write("Hello Világ!")

</script>

</body>

**Megjegyzés**: Ha megnézzük a böngészőben az oldalunk forrását, akkor látható, hogy ott az eredeti, letöltött

verzió látszik, és nem a JavaScript által „módosított”.

Jól látszik a példán, hogy JavaScriptben elfelejthetjük a sorokat lezáró pontosvesszőket.

A *document.write* módszerét a mai korszerű megközelítésben nem illik alkalmazni.

Nagyon egyszerű kódok kivételével nem szokás48 a JavaScript kódot a *body* tagba tenni.

Összetettebb működést inkább önálló függvények formájában szoktunk elhelyezni a

*head* tagba, vagy még inkább külső JavaScript állományba. Nézzük meg ezek módjait:

48 A könnyebb áttekinthetőség érdekében a demonstrációs példákban, vagy éppen a fejlesztés közben ezt a

szabályt időnként áthágjuk.

7.JavaScript 119. oldal

<head>

<script type="text/javascript">

function message() {

alert("This alert box was called with the onload event")

}

</script>

</head>

<body onload="message()">

...

</body>

A példán látszik, hogy a *body* tag betöltődésekor (*onload* esemény) fog lefutni a *message*

nevű függvény, ami egy figyelmeztető dialógusablakot dob fel. Ha a *message* függvény

kódját egy *message.js* állományban helyeztük volna, akkor a head részbe a következő

kód lenne szükséges:

<script type="text/javascript" src="message.js"></script>

**7.1.1 Változók**

A változók olyan tárolók, amelyekbe adatokat helyezhetünk. A változó értékét meg tudjuk

változtatni a szkript futása során. A változóra a nevével tudunk hivatkozni, és az értékét

bármikor lekérdezni vagy módosítani.

A változó nevek (mint ahogy az egész JavaScript nyelv) kis-nagybetű érzékenyek, és betűvel

vagy aláhúzás karakterrel kezdődnek.

**Változó deklaráció**

A *var* kulcsszó segítségével a következő módon tudunk változót létrehozni:

var strname = "Hege"

Létre lehet hozni változót *var* nélkül is:

strname = "Hege"

Ahogy a példákban is látszik, egyből értéket is adtunk a változóknak.