

INFO1 – Tartalom közlése a Weben: HTML, CSS

Kovács Kristóf, Pálovics Róbert, Wettl Ferenc

2016-10-04

1 Bevezetés

- Jelölőnyelv

2 WWW

- A kezdetek
- A web működése vázlatosan
- Sütik

3 HTML

- Tartalom és forma a HTML-ben
- További nyelvi elemek

4 CSS

Amit megtanulunk

- A jelölőnyelv fogalma, a tartalom és a vizuális megjelenítés szétválasztása
- A web működésének alapjai
- A web jelölőnyelve: HTML alapok + CSS alapok
- A tanultak alapján mindenki elkészítheti saját honlapját!
- Alapismeretek forrása: a [W3C: World Wide Web Consortium \(www.w3.org\)](http://www.w3.org) oldala, valamint a www.w3schools.com/html/ és a www.w3schools.com/css/ kurzus.
- Nagyobb munka előtt érdemes tanulmányozni a programozói stílusra vonatkozó ajánlásokat (pl. [Google HTML CSS guide](#))
- A kész fájlt érdemes „validálni”, azaz a kód helyességét ellenőriztetni pl. a W3C-vel: [HTML-t itt](#), [CSS-t itt](#).

1 Bevezetés

- Jelölőnyelv

2 WWW

- A kezdetek
- A web működése vázlatosan
- Sütik

3 HTML

- Tartalom és forma a HTML-ben
- További nyelvi elemek

4 CSS

Jelölőnyelv

- **Jelölőnyelv (markup language)** feliratokkal, jegyzetekkel lát el egy dokumentumot, úgy hogy az szintaktikailag elkülönül a dokumentum szövegétől.
- Előzmények a számítógép előtti időkből: az írógéppel írt szöveg jelölése a nyomda számára, a kefelenyomat korrektúrajelei, az első elektronikus nyomdagépek nyelve, . . .
- Szedési információkat jelölő nyelvek pl. a troff (AT&T Unix-ra írt dokumentumkezelője), $\text{T}_{\text{E}}\text{X}$, $\text{L}_{\text{A}}\text{T}_{\text{E}}\text{X}$ (matematikai tartalmú dokumentumokra), strukturális jelölést ad az XML (általános célú leíró nyelv, speciális célú leíró nyelvek létrehozására).

- Legegyszerűbb példák a **könnyű leíró (lightweight markup) nyelvek**: ember által könnyen írható, olvasható, más jelölő nyelvekre programmal könnyen konvertálható nyelvek. Pl. az intézeti [wiki](#) (vagy a Wikipédia) nyelve ilyen:

= Cím 1 =

== Cím 2 ==

Bekezdés, benne ''dőlt'', '''félkövér''' szöveg.

1 Bevezetés

- Jelölőnyelv

2 WWW

- A kezdetek
- A web működése vázlatosan
- Sütik

3 HTML

- Tartalom és forma a HTML-ben
- További nyelvi elemek

4 CSS

1 Bevezetés

- Jelölőnyelv

2 WWW

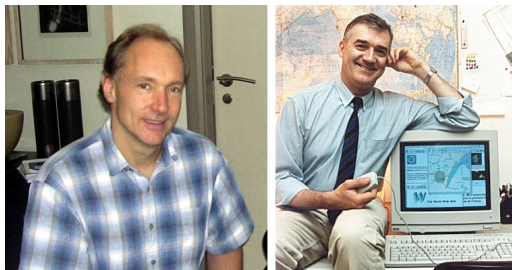
- A kezdetek
- A web működése vázlatosan
- Sütik

3 HTML

- Tartalom és forma a HTML-ben
- További nyelvi elemek

4 CSS

World Wide Web



Ábra: Sir Tim Berners-Lee (bal) és Robert Cailliau (jobb)

Sir Tim Berners-Lee és Robert Cailliau 1990-ben a CERN-ben tervezték meg a WWW-t, hogy egy olyan hálózatot hozzanak létre amelyben bárki hozzáférhet adott információkhoz.

World Wide Web

- Még 1990 végén elkészítették az első böngészőt, majd 1991-ben lett ténylegesen publikus a WWW az interneten.
- Hogy ez effektíven működhessen bevezették az URL-t (Uniform Resource Locator), a HTML-t (HyperText Markup Language) és a HTTP-t (HyperText Transfer Protocol).
- A CERN 1993-ban bejelentette, hogy a Web ingyenes lesz mindenkinek, majd később ebben az évben elkészült az első igazán elterjedt böngésző, a Mosaic.

1 Bevezetés

- Jelölőnyelv

2 WWW

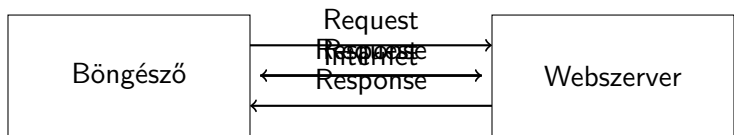
- A kezdetek
- A web működése vázlatosan
- Sütik

3 HTML

- Tartalom és forma a HTML-ben
- További nyelvi elemek

4 CSS

Böngésző és webszerver



A böngészőt összeköti az internet egy webszerverrel.

Küld a böngésző egy kérést (request) a webszervernek, hogy melyik adott oldalt szeretné lekérdezni, meg még küld egyéb dolgokat is mellette.

Kap erre egy választ (response), ami tartalmazza a HTML-t amit majd a böngésző megjelenít a felhasználónak.

Így folytatódik a böngészés, akárhányszor új oldalra akar lépni a felhasználó a böngésző küld egy kérést, majd erre válaszol a webszerver.

1 Bevezetés

- Jelölőnyelv

2 WWW

- A kezdetek
- A web működése vázlatosan
- Sütik

3 HTML

- Tartalom és forma a HTML-ben
- További nyelvi elemek

4 CSS

Sütik (cookies)

- Egy webszerver másodpercenként több ezer kérést kaphat, ez egy statikus oldalnál nem jelent gondot, de mi a helyzet például egy közösségi oldallal?
- A felhasználó be tud jelentkezni és utána bejelentkezve is marad, de honnan tudja a webszerver, hogy ki kicsoda?
- Egy webszerver válaszként nem csak HTML kódot küldhet, többek közt sütiket is. (fortune cookie)
- A süti egy ideiglenes élettartalmú minimális szövegfájl, mely egy adott weboldalhoz tartozik, melyet a webszerver küld és a felhasználó számítógépére a kereső menti egy erre kijelölt könyvtárba.
- Amikor kérést küld a böngésző egy weboldalnak, elküldi az oldalhoz tartozó sütiket is, innen tudja a webszerver, hogy „ki” küldte a kérést.

Sütik tartalma

- *Név:* A süti neve
- *Érték:* Az adat amit tárol
- *Tulajdonságok:*
 - *Lejárati idő:* mikor kell a böngészőnek törölnie a sütit
 - *Domain:* melyik weboldalhoz tartozik a süti
 - stb.

A böngésző ezek közül csak a süti nevét és értékét küldi el a kérésben, a többi a böngészőre tartozik.



Name:	PREF
Content:	ID=1111111111111111
Domain:	.google.hu
Path:	/
Send For:	Any type of connection
Expires:	31 December, 2015 17:02:22

Sütik biztonsága

- A süti nem tartalmazhatja egyszerűen a felhasználónevet, hisz ekkor nagyon könnyen hamisítható lenne.
- A süti nem vírus, nincs benne futtatható program.
- Tipikusan egy hosszú kulcsot (karakterorozat) tartalmaz, melyet a webszerver is tárol és összekapcsolja a felhasználóval.
- Így a felhasználó nem tudja, hogy másokhoz milyen kulcs tartozik, a webszerver viszont tudja, hogy melyik kulcs melyik felhasználóé.
- Példa: a keresők nagy része a keresési eredményeit személyre szabja és bejelentkezés nélkül is odafigyel a felhasználó preferenciáira.

1 Bevezetés

- Jelölőnyelv

2 WWW

- A kezdetek
- A web működése vázlatosan
- Sütik

3 HTML

- Tartalom és forma a HTML-ben
- További nyelvi elemek

4 CSS

HTML jelölő nyelv – címke (tag), attribútum (attribute)

- Jelölő (markup) nyelv, címkékből (tag-ek) áll.
- Szövegszerkesztőben szerkeszthető, programmal könnyen generálható.
- A HTML-kód elemén a kód egy címkével megjelölt részét értjük.
- Ennek formája `<címke_neve>tartalom</címke_neve>`, ha valamilyen tartalomra vonatkozik és `<címke_neve>` vagy `<címke_neve />`, ha önmagában áll.
- Például `szöveg` jelöli, hogy a **szöveg** kiemelt fontosságú, míg `
` vagy `
` egy sortörést eredményez.
- A címkéknek (tag-eknek) lehetnek paramétereik/attribútumai. Ezek formája `attribútum_név="tulajdonság"` alakú, például
Itt az `órarendem`.
- Megjegyzés a kódban: `<!-- akármi van itt nem látszik -->`

Példa oldal

```
<!DOCTYPE html>
<html lang="hu">
<meta charset="utf-8" />
<body>

<h1>Cím</h1>

<h2>Alcím</h2>

<p>Egy bekezdés.</p>

</body>
</html>
```

Cím

Alcím

Egy bekezdés.

A fontosabb címkék

- `<h1>Cím</h1>` (heading)
- `<h6>Al...alcím</h6>` (heading)
- `<p>Bekezdés</p>` (paragraph)
- `Kiemel` (emphasize)
- `Erősen kiemel` (strong)
- `<a>Horgony (link)` (anchor)
- `Számozatlan lista` (unordered list)
- `Számozott lista` (ordered list)
- `Listaelem` (list item)
- `<blockquote>Idézet</blockquote>` (blockquote)
- `<div>Rész</div>` (division)
- `Sorközi csoportosítás, arasznyi` (span)
- `` (image)

1 Bevezetés

- Jelölőnyelv

2 WWW

- A kezdetek
- A web működése vázlatosan
- Sütik

3 HTML

- Tartalom és forma a HTML-ben
- További nyelvi elemek

4 CSS

A tartalom és a vizuális megjelenítés

- Néhány nem tartalmi (vizuális) címke:
 - `<i>Kurzív (dőlt)</i>` (italic)
 - `Félkövér` (bold)
 - `_{Alsó index}` (subscript)
 - `^{Felső index}` (superscript)
 - `<pre>Előformázott</pre>` (preformatted)
 - `<hr />` (horizontal rule – thematic change in the content)
 - `
` (line brake)
- A tartalom szervezéséről
 - A `div` és a `span` a szemantikai összetartozást jelölik.
 - A `div` nagyobb szövegrészt zár magába, felette lehet akár még egy `div`, mely `div`-eket zár össze egy részbe.
 - A `span` soron belül végzi ezt.

1 Bevezetés

- Jelölőnyelv

2 WWW

- A kezdetek
- A web működése vázlatosan
- Sütik

3 HTML

- Tartalom és forma a HTML-ben
- További nyelvi elemek

4 CSS

Attribútumok

- title cím egy HTML-elemhez:

```
<p title="Ez nagyon fontos!">Itt van a lényeg.</p>
```

- href az a címke mellett a link címét adja meg:

```
HTML <a href="http://www.w3schools.com">kurzus</a>.
```

- src a hivatkozás forrása, alt az alternatív megjelenés (pl. felolvasáskor vagy ha nem találja a képfájlt), width és height pixelben értendő:

```

```

- style="tulajdonság:érték" az elem stílusát adja meg:

```
<body style="background-color:lightgrey">
```

```
<p style="color:red">Pirossal szedett bekezdés.</p>
```

```
<p style="font-family:courier">Courier bekezdés.</p>
```

```
<h1 style="text-align:center">Cím középen</h1>
```



Speciális karakterek

karakter	html-ben	(angol) megnevezés
<	<	less-than
>	>	greater-than
&	&	ampersand
	 	non-breaking space

- 1 Bevezetés
 - Jelölőnyelv
- 2 WWW
 - A kezdetek
 - A web működése vázlatosan
 - Sütik
- 3 HTML
 - Tartalom és forma a HTML-ben
 - További nyelvi elemek
- 4 CSS

CSS

- CSS: Cascading Style Sheets (cascading: több helyen lehet a stílus megadva, lépcsőzetesen a legáltalánosabb szabálytól haladva speciális felé dönt a megjelenítésről.)
- Cél: a tartalom és a megjelenítés szétválasztása
- Fizikai szétválasztás lehetősége: stílus .css-fájl(ok)ban
- HTML elemeinek formázására szolgál
- Előnyei: újrafelhasználhatóság és könnyű módosíthatóság
- Forrás tanuláshoz: www.w3schools.com/css,
- Validálás: jigsaw.w3.org/css-validator/
- Csak szórakoztatásul a lehetőségekről, nem követendő: [CSS Zen Garden](#)

CSS szintaxis

- A CSS a következő alapvető szintaxist követi:

```
kiválasztó {tulajdonság: érték; tulajdonság2: érték2;}
```

- Pl:

```
p {color: red;}
```

vagy több deklaráció áttekinthetően szedve:

```
p {  
  color:#f00;  
  background: white;  
}
```

- Több elem stílusa egyszerre megadható:

```
h1, h2 {color: red;}
```

- Megjegyzés (comment):

```
/* Ide bármit írhatok, akár több sorban is. */
```



CSS kód elhelyezése

- Inline (szövegközi):

```
<p style=".....CSS kódok.....">bekezdés szövege</p>
```

- Beágyazott, ez a html *head* részébe megy:

```
<style>  
    CSS formázás kódolása  
</style>
```

- Külső stílusfile (ez a legelterjedtebb), szintén a *head*-be:

```
<link rel="stylesheet" href=".....css">
```

- Legerősebb az inline, majd a beágyazott és végül a külső.

Csoportosítás

A html kódban csoportosíthatjuk a *címkéinket* (tageinket). Egyedi nevet adhatunk nekik, vagy egy közös osztályba rakhatjuk őket.

id weboldalankét egy id csak egy elemhez tartozhat, és egy elemnek csak egy id-ja lehet, így ez egyfajta elnevezésként képzelhető el

```
<div id="hirek">.....</p>
```

class több elemnek lehet ugyanaz az osztálya, és egy elemnek több osztálya is lehet

```
<p class="megjegyzes">.....</p>
```

```
<p class="megjegyzes">.....</p>
```

```
...
```

```
<div class="megjegyzes">.....</div>
```

Kiválasztók

- elemkiválasztó: mindig egy html jelölőelemre vonatkozik
`p {color: red;}`
- id kiválasztó: az adott id-vel jelölt elemre vonatkozik
`#hirek {background: white;}`
- osztálykiválasztó (class kiválasztó): az adott osztályba tartozó elemekre vonatkozik
`.megjegyzes {font-size: small;}`
- kombinálhatjuk az elem- és osztálykiválasztót, pl:
`p.megjegyzes {color: blue;}`
ekkor azok a p-k lesznek kékek, amik a megjegyzés osztályba tartoznak

Link megjelenítései

- Linkeknél vannak még speciális kijelölő elemek, ezekkel mondhatjuk meg, hogy hogyan nézzen ki egy link
 - alaphelyzetben (`link`),
 - ha már látogatott oldalra mutat (`visited`),
 - ha épp fölé visszük az egeret (`hover`),
 - amikor épp rákattintunk (`active`).

- Például (LoVe-HAte sorrendben):

```
a:link {  
  color: green;  
  text-decoration: underline;  
}  
a:visited { color: purple; }  
a:hover {  
  color: blue;  
  text-decoration: none;  
}  
a:active { color:red; }
```


Kérdések

- Mi a jelölőnyelv?
- Mik a sütik és főként mire valók?
- Melyik címkével (tag-gel) tudunk képet behelyezni az oldalra?
- Melyik címkével tudunk linket behelyezni az oldalra?
- Melyik attribútummal adhatunk meg inline CSS formázást?
- Soroljunk fel néhány, a HTML-ben speciális karaktert!
- Hogyan választható szét a tartalom és a forma (megjelenítés) a HTML-ben?
- Milyen állapotai vannak egy linknek? Adjunk egy példát, ahol CSS kód segítségével mindegyiken változtatunk valamit!
- Mi a különbség a class és az id között? Példával mutassuk be melyiket hogyan használjuk!