

Informatika 1

Internet, HTML, CSS

Wetl Ferenc és Kovács Kristóf
prezentációjának felhasználásával

Budapesti Műszaki és Gazdaságtudományi Egyetem

2018-10-04

Az internetre kötött gépek azonosítására szolgáló cím az **IP cím** (IP address), ami

- IPv4 szabvány: `nnn.nnn.nnn.nnn` alakú (32 bites, 4 db 8-bites szám decimális alakban) – 2015 nyarán kifogyott
- IPv6 szabvány: `xxxx:xxxx:xxxx:xxxx:xxxx:xxxx:xxxx:xxxx` alakú (128 bit, 8 db 16 bites hexadecimálisan ábrázolt szám)

Az internetre kötött gépek azonosítására szolgáló cím az **IP cím** (IP address), ami

- IPv4 szabvány: `nnn.nnn.nnn.nnn` alakú (32 bites, 4 db 8-bites szám decimális alakban) – 2015 nyarán kifogyott
- IPv6 szabvány: `xxxx:xxxx:xxxx:xxxx:xxxx:xxxx:xxxx:xxxx` alakú (128 bit, 8 db 16 bites hexadecimálisan ábrázolt szám)

típus	cím	honnan tudom meg?
IPv4	152.66.83.241	http://miazipcimem.hu/
IPv6	2001:738:2001:2010:891b:efb:2b36:5447	http://whatismyipaddress.com/
lokális hálózat	192.168.xxx.xxx	ifconfig (WIN ipconfig)
lokális ipV6	fe80::	Reserved IP addresses

Az internetre kötött gépek azonosítására szolgáló cím az **IP cím** (IP address), ami

- IPv4 szabvány: `nnn.nnn.nnn.nnn` alakú (32 bites, 4 db 8-bites szám decimális alakban) – 2015 nyarán kifogyott
- IPv6 szabvány: `xxxx:xxxx:xxxx:xxxx:xxxx:xxxx:xxxx:xxxx` alakú (128 bit, 8 db 16 bites hexadecimálisan ábrázolt szám)

típus	cím	honnan tudom meg?
IPv4	152.66.83.241	http://miazipcimem.hu/
IPv6	2001:738:2001:2010:891b:efb:2b36:5447	http://whatismyipaddress.com/
lokális hálózat	192.168.xxx.xxx	ifconfig (WIN ipconfig)
lokális IPv6	fe80::	Reserved IP addresses

Ezeket a címeket decentralizált internet hatóságok osztják ki. Adott címtartományt kapnak a kontinensek, akik eldöntik, hogy milyen címtartományokat kapnak az országok és így tovább.

- Egy internetcímen is folyhat párhuzamos kommunikáció a **port**okon keresztül

- Egy internetcímen is folyhat párhuzamos kommunikáció a **port**okon keresztül
- Port azonosító: 16 bit-es előjel nélküli szám (0-65535-ig)

- Egy internetcímen is folyhat párhuzamos kommunikáció a **port**okon keresztül
- Port azonosító: 16 bit-es előjel nélküli szám (0-65535-ig)
- Ezekre van pár konvenció, pl a honlapokat a 80, 8000 vagy 8080-as porton szoktuk nézegetni.

A ping

- A **ping** egy rendszerprogram (utility), mely eldönti, hogy egy adatcsomag hibátlanul eljut-e a megadott IP címre.

A ping

- A **ping** egy rendszerprogram (utility), mely eldönti, hogy egy adatcsomag hibátlanul eljut-e a megadott IP címre.
- Ha a ping parancs után nem IP cím áll, hanem egy név, a **DNS (Domain Name System)** szolgáltatással megtudja, hogy a szerver nevéhez (host name) milyen IP-cím tartozik, majd egy PING üzenetet küld a címre.

A ping

- A **ping** egy rendszerprogram (utility), mely eldönti, hogy egy adatcsomag hibátlanul eljut-e a megadott IP címre.
- Ha a ping parancs után nem IP cím áll, hanem egy név, a **DNS (Domain Name System)** szolgáltatással megtudja, hogy a szerver nevéhez (host name) milyen IP-cím tartozik, majd egy PING üzenetet küld a címre.
- PING jelentése "Send a packet to a computer and wait for its return (Packet INternet Groper)" (groper – molesztáló)

A ping

- A **ping** egy rendszerprogram (utility), mely eldönti, hogy egy adatcsomag hibátlanul eljut-e a megadott IP címre.
- Ha a ping parancs után nem IP cím áll, hanem egy név, a **DNS (Domain Name System)** szolgáltatással megtudja, hogy a szerver nevéhez (host name) milyen IP-cím tartozik, majd egy PING üzenetet küld a címre.
- PING jelentése "Send a packet to a computer and wait for its return (Packet INternet Groper)" (groper – molesztáló)
- A domain neveket internet hatóságok osztják ki, náluk lehet beregisztrálni.

- A **ping** egy rendszerprogram (utility), mely eldönti, hogy egy adatcsomag hibátlanul eljut-e a megadott IP címre.
- Ha a ping parancs után nem IP cím áll, hanem egy név, a **DNS (Domain Name System)** szolgáltatással megtudja, hogy a szerver nevéhez (host name) milyen IP-cím tartozik, majd egy PING üzenetet küld a címre.
- PING jelentése "Send a packet to a computer and wait for its return (Packet INternet Groper)" (groper – molesztáló)
- A domain neveket internet hatóságok osztják ki, náluk lehet beregisztrálni.

A ping

- A **ping** egy rendszerprogram (utility), mely eldönti, hogy egy adatcsomag hibátlanul eljut-e a megadott IP címre.
- Ha a ping parancs után nem IP cím áll, hanem egy név, a **DNS (Domain Name System)** szolgáltatással megtudja, hogy a szerver nevéhez (host name) milyen IP-cím tartozik, majd egy PING üzenetet küld a címre.
- PING jelentése "Send a packet to a computer and wait for its return (Packet INternet Groper)" (groper – molesztáló)
- A domain neveket internet hatóságok osztják ki, náluk lehet beregisztrálni.

```
C:\Users\Tofi>ping bme.hu

Pinging bme.hu [152.66.115.203] with 32 bytes of data:
Reply from 152.66.115.203: bytes=32 time=66ms TTL=52
Reply from 152.66.115.203: bytes=32 time=69ms TTL=52
Reply from 152.66.115.203: bytes=32 time=73ms TTL=52
Reply from 152.66.115.203: bytes=32 time=62ms TTL=52

Ping statistics for 152.66.115.203:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 62ms, Maximum = 73ms, Average = 67ms
```



- **Jelölőnyelv (markup language)** feliratokkal, jegyzetekkel lát el egy dokumentumot, úgy hogy az szintaktikailag elkülönül a dokumentum szövegétől.

- **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,...

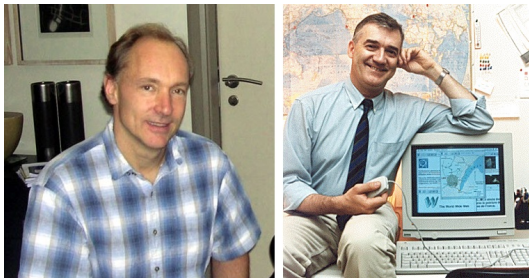
- **Jelölőnyelv (markup language)** feliratokkal, jegyzetekkel lát el egy dokumentumot, úgy hogy az szintaktikailag elkülönül a dokumentum szövegétől.
- Előzmények a számítógép előtti időkből: az írógéppel írt szöveg jelölése a nyomda számára, a kefelenyomat korrektúrajelei, az első elektronikus nyomdagépek 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)** vagy **markdown** 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.



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.

- Még 1990 végén elkészítették az első böngészőt, majd 1991-ben lett ténylegesen publikus a WWW az interneten.

- Még 1990 végén elkészítették az első böngészőt, majd 1991-ben lett ténylegesen publikus a WWW az interneten.
- Hogy ez effektíven működhessen bevezették az

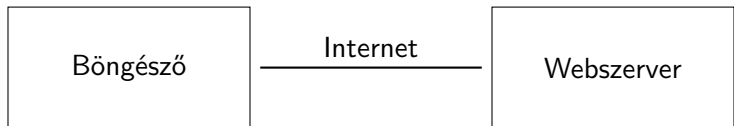
- Még 1990 végén elkészítették az első böngészőt, majd 1991-ben lett ténylegesen publikus a WWW az interneten.
- Hogy ez effektíven működhessen bevezették az
 - URL-t (Uniform Resource Locator), ami az elérési címeket egységesíti,

- 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), ami az elérési címeket egységesíti,
 - és a HTTP-t (HyperText Transfer Protocol), ami a gépek (böngészés közbeni) kommunikációját szabványosítja,

- 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), ami az elérési címeket egységesíti,
 - és a HTTP-t (HyperText Transfer Protocol), ami a gépek (böngészés közbeni) kommunikációját szabványosítja,
 - és a HTML-t (HyperText Markup Language), ami az oldalak tartalmának formátumát írja le.

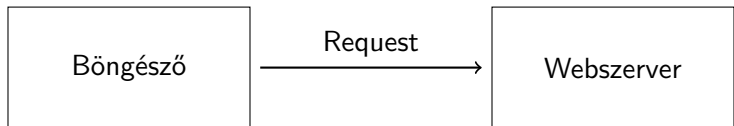
- Még 1990 végén elkészítették az első böngészőt, majd 1991-ben lett ténylegesen publikus a WWW az interneten.
- Hogy ez effektíven működhessen bevezették az
 - URL-t (Uniform Resource Locator), ami az elérési címeket egységesíti,
 - és a HTTP-t (HyperText Transfer Protocol), ami a gépek (böngészés közbeni)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

Böngésző és webszerver



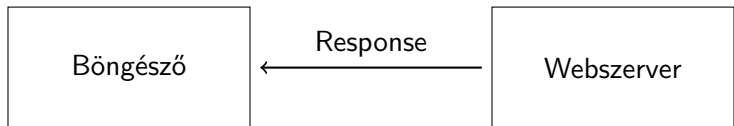
A böngészőt összeköti az internet egy webszerverrel.

Böngésző és webszerver



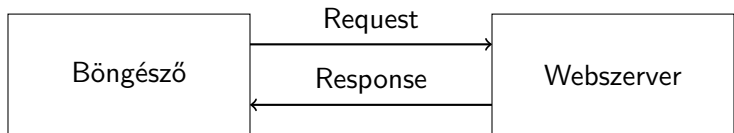
Küld a böngésző egy kérést (request) a webszervernek, hogy melyik adott oldalt szeretné lekérdezni, meg még küld egyéb dolgokat is mellette.

Böngésző és webszerver



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

Böngésző és webszerver



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

Sütik (cookies)

- Egy webservert másodpercenként több ezer kérést kaphat, ez egy statikus oldalnál nem jelent gondot, de mi a helyzet például egy közösségi oldallal?

Sütik (cookies)

- Egy webserver 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 webserver, hogy ki kicsoda?

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)

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.

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
```

- A süti nem tartalmazhatja egyszerűen a felhasználónevet, hisz ekkor nagyon könnyen hamisítható lenne.

- A süti nem tartalmazhatja egyszerűen a felhasználónevet, hisz ekkor nagyon könnyen hamisítható lenne.
- A süti nem vírus, nincs benne futtatható program.

- 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 (karakter sorozatot) tartalmaz, melyet a webszerver is tárol és összekapcsolja a felhasználóval.

- 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 (karakter sorozatot) 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.

- 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 (karakter sorozatot) 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.

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

- Egy speciális XML (eXtensible Markup Language), ami címkékből (tag-ek) áll.

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

- Egy speciális XML (eXtensible Markup Language), ami címkékből (tag-ek) áll.
- Szövegszerkesztőben szerkeszthető, programmal könnyen generálható.

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

- Egy speciális XML (eXtensible Markup Language), ami címkékből (tag-ek) áll.
- Szövegszerkesztőben szerkeszthető, programmal könnyen generálható.
- A HTML-kód *elemén* a kód egy címkével megjelölt részét értjük.

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

- Egy speciális XML (eXtensible Markup Language), ami 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.

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

- Egy speciális XML (eXtensible Markup Language), ami 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.

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

- Egy speciális XML (eXtensible Markup Language), ami 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 paraméterei/attribútumai. Ezek formája `attribútum_név="tulajdonság"` alakú, például
Itt az `órarendem`.

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

- Egy speciális XML (eXtensible Markup Language), ami 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 (HTML 5 szabvány)

```
<!DOCTYPE html>                <!-- oldal típusa -->
<html lang="hu">
<head>                            <!-- címkék, meta-adatok ... -->
<meta charset="utf-8" />        <!-- karakter kódolás -->
</head>
<body>

...                                <!-- az oldal "teste" -->

</body>
</html>
```

Speciális karakterek

karakter	html-ben	(angol) megnevezés
<	<	less-than
>	>	greater-than
"	"	quotation mark
&	&	ampersand
	 	non-breaking space
©	©	copyright
®	®	registered trademark
™	™	trademark

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)

A tartalom és a vizuális megjelenítés

- Néhány nem tartalmi (vizuális) címke:

A tartalom és a vizuális megjelenítés

- Néhány nem tartalmi (vizuális) címke:
 - `<i>Kurzív (dőlt)</i>` (italic)

A tartalom és a vizuális megjelenítés

- Néhány nem tartalmi (vizuális) címke:
 - `<i>Kurzív (dőlt)</i>` (italic)
 - `Félkövér` (bold)

A tartalom és a vizuális megjelenítés

- Néhány nem tartalmi (vizuális) címke:
 - `<i>Kurzív (dőlt)</i>` (italic)
 - `Félkövér` (bold)
 - `_{Alsó index}` (subscript)

A tartalom és a vizuális megjelenítés

- Néhány nem tartalmi (vizuális) címke:
 - `<i>Kurzív (dőlt)</i>` (italic)
 - `Félkövér` (bold)
 - `_{Alsó index}` (subscript)
 - `^{Felső index}` (superscript)

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)

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)

- 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)
 - `
` (line brake)

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)
 - `
` (line brake)
- A tartalom szervezéséről

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)
 - `
` (line brake)
- A tartalom szervezéséről
 - A `div` és a `span` a szemantikai összetartozást jelölik.

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)
 - `
` (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 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)
 - `
` (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.

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)
 - `
` (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.
- A tartalom és a külalak elválasztása

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)
 - `
` (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.
- A tartalom és a külalak elválasztása
 - A `<p "stilus attribútumok ...">tartalom</p>`

- CSS: Cascading Style Sheets (cascading: több helyen lehet a stílus megadva, lépcsőzetesen a legáltalánosabb szabálytól haladva speciális felé dönt a megjelenítésről.)

- 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

- 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

- 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

- 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

- 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,

- 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/

- 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

- A CSS a következő alapvető szintaxist követi:
kiválasztó {tulajdonság: érték; tulajdonság2: érték2;}

- A CSS a következő alapvető szintaxist követi:
kiválasztó {tulajdonság: érték; tulajdonság2: érték2;}
- Pl:
p {color: red;}

- 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;  
}
```

- 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;}

- 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. */

- Inline (szövegközi):

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

- Inline (szövegeközi):

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

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

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

- Inline (szövegeközi):

```
<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">
```

- Inline (szövegeközi):

```
<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.

- Inline (szövegeközi):

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

- Inline (szövegekzi):

```
<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.
- A html kódban csoportosíthatjuk a *címkéinket* (tageinket). Egyedi nevet adhatunk nekik, vagy egy közös osztályba rakhatjuk őket.
 - Ezekre így hivatkozhatunk a CSS-ben:
(tag)(.class)(#id){ ... }

- Linkeknél vannak még speciális kijelölő elemek, ezekkel mondhatjuk meg, hogy hogyan nézzen ki egy link

- Linkeknél vannak még speciális kijelölő elemek, ezekkel mondhatjuk meg, hogy hogyan nézzen ki egy link
 - alaphelyzetben (`link`),

- Linkeknél vannak még speciális kijelölő elemek, ezekkel mondhatjuk meg, hogy hogyan nézzen ki egy link
 - alaphelyzetben (`link`),
 - ha már látogatott oldalra mutat (`visited`),

- 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`),

- 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`).

- 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; }
```


- HTML-ben az alábbiak azonosak:

- HTML-ben az alábbiak azonosak:
 - `<p>bekezdés szó</p>`

- HTML-ben az alábbiak azonosak:
 - `<p>bekezdés szó</p>`
 - `<p>
bekezdés szó
</p>`

- HTML-ben az alábbiak azonosak:
 - `<p>bekezdés szó</p>`
 - `<p>`
bekezdés `szó`
`</p>`
 - az utóbbi a preferált

A kód kinézete

- HTML-ben az alábbiak azonosak:
 - `<p>bekezdés szó</p>`
 - `<p>`
bekezdés `szó`
`</p>`
 - az utóbbi a preferált
- CSS-ben az alábbiak azonosak:

A kód kinézete

- HTML-ben az alábbiak azonosak:
 - `<p>bekezdés szó</p>`
 - `<p>`
bekezdés `szó`
`</p>`
 - az utóbbi a preferált
- CSS-ben az alábbiak azonosak:
 - `p{color:black;background-color:white;}`

- HTML-ben az alábbiak azonosak:
 - `<p>bekezdés szó</p>`
 - `<p>`
bekezdés `szó`
`</p>`
 - az utóbbi a preferált
- CSS-ben az alábbiak azonosak:
 - `p{color:black;background-color:white;}`
 - `p {`
color: black;
background-color: white;
}

- HTML-ben az alábbiak azonosak:
 - `<p>bekezdés szó</p>`
 - `<p>`
bekezdés `szó`
`</p>`
 - az utóbbi a preferált
- CSS-ben az alábbiak azonosak:
 - `p{color:black;background-color:white;}`
 - `p {`
color: black;
background-color: white;
}
 - az utóbbi a preferált

- HTML-ben az alábbiak azonosak:
 - `<p>bekezdés szó</p>`
 - `<p>`
bekezdés `szó`
`</p>`
 - az utóbbi a preferált
- CSS-ben az alábbiak azonosak:
 - `p{color:black;background-color:white;}`
 - `p {`
color: black;
background-color: white;
}
 - az utóbbi a preferált
- ezt hívjuk **kódkonvenció**nak, minden nyelvnek megvannak a szabályai és konvenciói.

- HTML-ben az alábbiak azonosak:
 - `<p>bekezdés szó</p>`
 - `<p>`
bekezdés `szó`
`</p>`
 - az utóbbi a preferált
- CSS-ben az alábbiak azonosak:
 - `p{color:black;background-color:white;}`
 - `p {`
color: black;
background-color: white;
}
 - az utóbbi a preferált
- ezt hívjuk **kódkonvenció**nak, minden nyelvnek megvannak a szabályai és konvenciói.
- Átláthatóbb az ember számára.

- HTML-ben az alábbiak azonosak:
 - `<p>bekezdés szó</p>`
 - `<p>`
bekezdés `szó`
`</p>`
 - az utóbbi a preferált
- CSS-ben az alábbiak azonosak:
 - `p{color:black;background-color:white;}`
 - `p {`
color: black;
background-color: white;
}
 - az utóbbi a preferált
- ezt hívjuk **kódkonvenció**nak, minden nyelvnek megvannak a szabályai és konvenciói.
- Átláthatóbb az ember számára.
- Az **identálás** az, amikor az egymásba ágyazott elemeket beljebb kezdjük.

- Milyen szolgáltatást nyújt a DNS?

- Milyen szolgáltatást nyújt a DNS?
- Mit jelentenek: URL, HTTP, HTML?

- Milyen szolgáltatást nyújt a DNS?
- Mit jelentenek: URL, HTTP, HTML?
- Mi a jelölőnyelv?

- Milyen szolgáltatást nyújt a DNS?
- Mit jelentenek: URL, HTTP, HTML?
- Mi a jelölőnyelv?
- Mik a sütik és mire valók?

- Milyen szolgáltatást nyújt a DNS?
- Mit jelentenek: URL, HTTP, HTML?
- Mi a jelölőnyelv?
- Mik a sütik és mire valók?
- Melyik címkével (tag-gel) tudunk képet behelyezni az oldalra?

- Milyen szolgáltatást nyújt a DNS?
- Mit jelentenek: URL, HTTP, HTML?
- Mi a jelölőnyelv?
- Mik a sütik és mire valók?
- Melyik címkével (tag-gel) tudunk képet behelyezni az oldalra?
- Melyik címkével tudunk linket behelyezni az oldalra?

- Milyen szolgáltatást nyújt a DNS?
- Mit jelentenek: URL, HTTP, HTML?
- Mi a jelölőnyelv?
- Mik a sütik és 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?

- Milyen szolgáltatást nyújt a DNS?
- Mit jelentenek: URL, HTTP, HTML?
- Mi a jelölőnyelv?
- Mik a sütik és 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!

- Milyen szolgáltatást nyújt a DNS?
- Mit jelentenek: URL, HTTP, HTML?
- Mi a jelölőnyelv?
- Mik a süтик és 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 szolgáltatást nyújt a DNS?
- Mit jelentenek: URL, HTTP, HTML?
- Mi a jelölőnyelv?
- Mik a süтик és 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!

- Milyen szolgáltatást nyújt a DNS?
- Mit jelentenek: URL, HTTP, HTML?
- Mi a jelölőnyelv?
- Mik a süтик és 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!