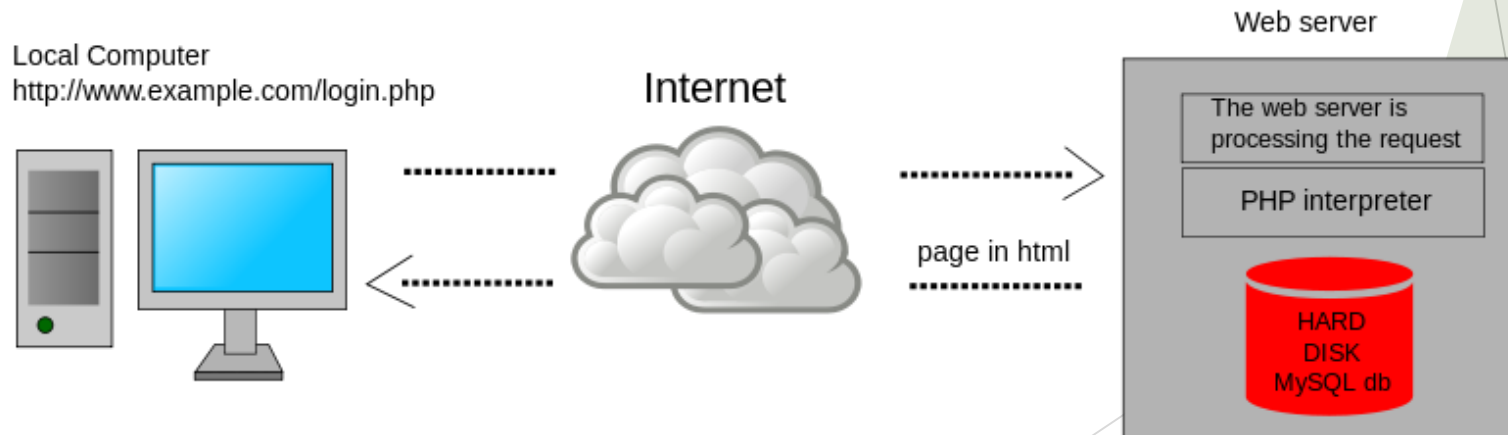


A statikus weblapok

# Statikus és dinamikus weblapok

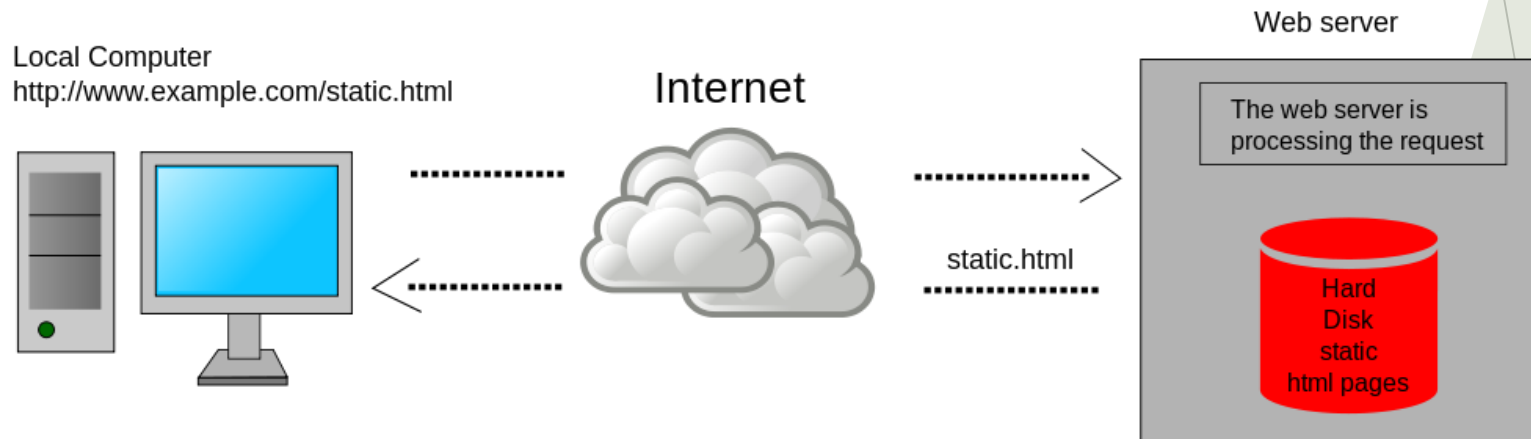
A honlapok lehetnek statikusak és dinamikusak.

- ▶ A statikus weblapok tartalma és forrás kódja állandó. A statikus weblapok létrehozására alkalmas a HTML (jelölőnyelv) és JavaScript (szkript nyelv) és minden kliens oldalon futó webes nyelv.
- ▶ A dinamikus weblapok pl. honlapmotor segítségével generálódnak a szerveren lekérési paraméterek és jelenlegi állapot alapján. A kéréseket (Request) a kliens küldi a szerverre és így létrejön pl. az interneten keresztül a kommunikáció.



# A statikus weblap

- ▶ Fontos különbség, hogy a kliens (pl. webböngésző program) a szerveren levő előre megírt forráskódot értelmezi és megjeleníti, tehát a html fájl manuálisan akár egy szövegszerkesztővel is könnyen átírható a szerveren.



# A HTML nyelv

- ▶ A HyperText Markup Language egy jelölőnyelv ami mára már internetes szabvánnyá vált.
- ▶ Forráskódja .html/.htm kiterjesztésű fájlban található
- ▶ Strukturális elemek, kulcsszavak („tag”-ek) segítségével definiáljuk a szöveg megjelenési formáját
- ▶ Könnyen értelmezhető és könnyen átírható, mindössze a kulcsszavak ismeretére és megfelelő alkalmazására van szükség. Minden nyelvnek a dokumentációjában az interneten könnyen megtalálhatjuk a kulcsszavak leírását és használatát  
Referencia: <http://www.w3schools.com/tags/default.asp>
- ▶ Mára egyre jobban elterjedt a HTML5 használata (videó/zene lejátszást támogat!). Ezt már minden modern böngésző képes értelmezni

## Érdekesség:

*A HTML nyelv Markup (jelölő) nyelv és nem a hagyományos értelemben vett programozási nyelv*

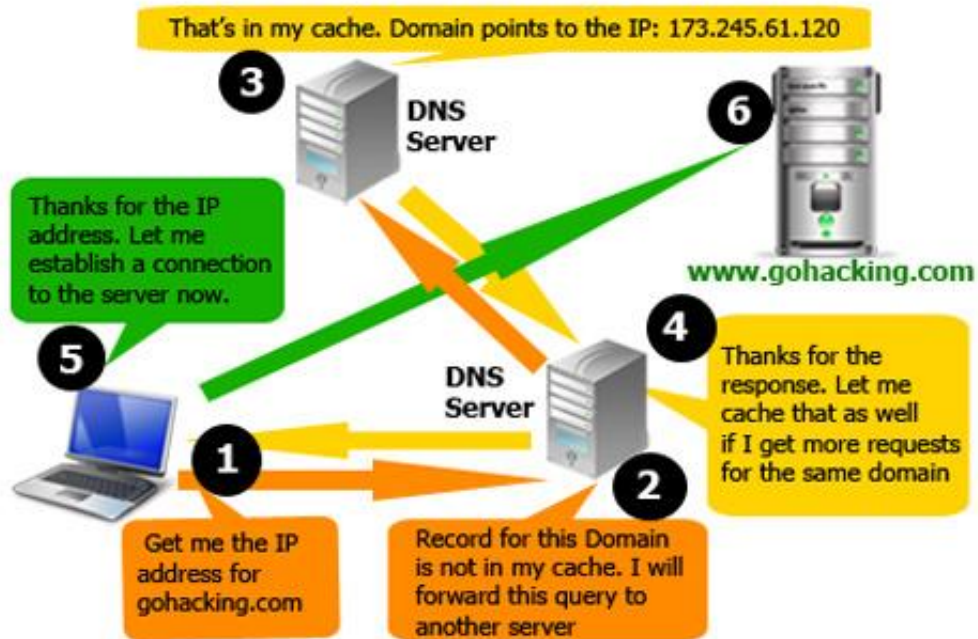
<http://stackoverflow.com/questions/145176/is-html-considered-a-programming-language>

# A Markup nyelvek

- ▶ A HTML a jelölőnyelvek családjába tartozik
- ▶ Valamilyen szabály szerint kulcsszavakkal jelöljük meg a forráskódban a megjelenítendő szöveg stílusát és megjelenési formáját
- ▶ Általában nem túl szigorúak a szabályok (pl. kulcsszavakat nem muszáj kisbetűvel írni), de szabvány szabályozhatja, hogy a kulcsszavakat hogy kell megjelölni, lezárni
- ▶ Néhány ilyen nyelv:
  - ▶ XML (.xml) - általános jelölő nyelv, tetszőleges kulcsszavakkal
  - ▶ HTML (.html) - elnéző szabályok
  - ▶ XHTML (.xhtml) - szigorúbb szabályok, így bővíthető (eXtensible)
  - ▶ Markdown (.md)
  - ▶ TeX (.tex) - matematikai stílusformázás

# Honlapcímek feloldása

- ▶ Mi történik amikor beírunk egy honlapcímet?
- ▶ Minden internetre kötött szervernek és gépnek van egy egyéni azonosítója, IP címe (IPv4 hamarosan IPv6!) amit az internetszolgáltató oszt ki
- ▶ Amikor beírunk egy szöveges honlapcímet, akkor a DNS (Domain Name System) szerverek segítségével a szöveges címhez IP cím rendelődik és így a kliensünk megtalálja a szerveret az interneten és létrejön a kommunikáció



Érdekesség:  
Hamarosan elfognak az IPv4 címek, jön az IPv6!  
[https://hu.wikipedia.org/wiki/Az\\_IPv4-c%C3%ADmek\\_elfogy%C3%A1sa](https://hu.wikipedia.org/wiki/Az_IPv4-c%C3%ADmek_elfogy%C3%A1sa)

# A szerver honlapcíme

- ▶ Amikor létrejött a kapcsolat szerver és kliens között, akkor a webszerver (pl. WAMP, IIS) konfigurációja alapján a kezdő oldal jelenik meg (ha tartománynevet (domain name) írtunk be: pl: hu.wikipedia.org), ez általában az index.html
- ▶ Ha URL címet írtunk be, akkor egyből a szerver egy aloldalát nyithatjuk meg
- ▶ (pl: `http://www.domainnevem.hu/utvonal/aloldal.htm`)
- ▶ A címbe:
  - ▶ `http` - a kommunikáció (protokoll) típusa
  - ▶ `www.domainnevem.hu` - a szerver tartományneve (domain name) amihez rendelhetünk egy IP-t
  - ▶ `utvonal/aloldal.htm` - a fájlhoz vezető elérési út a célgépen belül.

# A HTML nyelv szintaxisa

- ▶ Konvenció szerint a kezdő oldal index.html nevet kapja, az aloldalak aloldal.htm neveket
- ▶ Első html fájlunk:

```
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title>Helló Világ</title>
</head>
<body>

<h1>Az első rovatom</h1>
<p>Az első paragrafusom.</p>

</body>
</html>
```

- ▶ Minden kulcsszó („tag”) <> között van, a tag-ek érvényességi határa egyértelműen látszódik <tag>-al kezdődik és </tag>-al záródik
- ▶ A kulcsszavakat kis és nagybetűvel is írhatjuk (megengedő markup) pl: head=HEAD, de fontos megjegyezni, hogy nemcsak a honlapunk hanem a forráskódunk is igényesen kell kinézzen, azért hogy a saját kódunkat később is átlássuk és azért, hogy mások is megérthessék, felhasználhassák és továbbfejleszthessék bármilyen kódunkat. Törekedni kell a jó egységes stílusra, minél több kommenttel érdemes ellátni a kódot és figyelni kell a minél jobb (újra) felhasználhatóságára. Tehát pl. jó konvenció: minden tag kisbetű legyen



# Felépítése

## Egy HTML állomány három fő részre bontható:

- ▶ HTML 4.01-nél: A Dokumentum Típus Definíció az állomány legelején, ami a használni kívánt DTD-t adja meg (böngészőnek segített a megjelenítésben), pl:
  - ▶ `<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01//EN" "http://www.w3.org/TR/html4/strict.dtd">`
  - ▶ Ez elavult, HTML 5-ben már így néz ki:  
`<!DOCTYPE html>`
- ▶ Az egész HTML `<html>` taggal kezdődik és `</html>`-el zárul
- ▶ a HTML fejléc `<head>`, ami technikai és dokumentációs adatokat tartalmaz, amelyeket az internet böngésző nem jelenít meg, tehát átlag felhasználó ezeket nem látja és
- ▶ a HTML törzs `<body>`, amely a megjelenítendő információkat tartalmazza.
- ▶ Tag-ek tartalmazhatnak attribútumokat is, ezek név=„érték” párosok pl: (itt az attribútum neve: href)  
`<a href=" https://hu.wikipedia.org">Ez egy link</a>`

# A HTML fejléc

▶ `<head>...</head>` között van

▶ Tag-ek:

▶ `<title>` - Oldal címét adja meg

▶ `<style>` - Beírt (inline) CSS kód helye

▶ `<link>` - Elérési útvonal a szerveren külső fájlhoz (pl .css fájlhoz)  
Attribútumok: `rel="stylesheet" href="mystyle.css"`

▶ `<meta>` - Egyéb információk az oldalról, készítőről...

Keresőmotoroknak: `<meta name="keywords" content="HTML,alapot">`

Honlap leírása: `<meta name="description" content="HTML alapok">`

.html fájl kódolása: `<meta charset="UTF-8">`

Készítő: `<meta name="author" content="Valaki Vagyok"> ...`

Referencia: [http://www.w3schools.com/tags/tag\\_meta.asp](http://www.w3schools.com/tags/tag_meta.asp)

▶ `<script>` - Beírt JavaScript helye

▶ `<base>` - Aloldalak elérése

`<base href="http://www.honlapom.com/kepek/" target="_blank">`

Attribútum: új lapon nyit meg mindent: `target="_blank"`

# A HTML törzs

- ▶ `<body>...</body>` között van
- ▶ Tag-ek:
  - ▶ `<h1>...<h6>` - Fontossági és nagysági sorrendben a rovatok tag-jei
  - ▶ `<p>` - Paragrafus
  - ▶ `<hr>` - vízszintes vonal, `<br>` - sortörés  
Figyelem, ezeknek a tag-eknek nincs bezáró párjuk (XHTML-ben önmagát be kell zárni: `<hr />` és `<br />`)
  - ▶ Attribútumok:  
`style="text-align:center;"` - CSS stílus alkalmazása (figyelem lezáró pontosvessző)  
`background-color, color, font-family, font-size, text-align`
  - ▶ `<pre>` - Ahogy bekerül a szöveg ebbe a részbe úgy fog megjelenni  
Ugyanis a kódban az egymás mellett levő több szóközt (whitespace) egy szóközzel helyettesít, ez a tag ezt megakadályozza

# A HTML formázás

- ▶ Formázási tagek:
  - ▶ `<b>` vagy `<strong>` - **Vastagítás**
  - ▶ `<i>` vagy `<em>` - *Döntés*
  - ▶ `<small>` - Kisebb szöveg
  - ▶ `<sup>` - Kitevő
  - ▶ `<sub>` - index
  - ▶ `<ins>` - aláhúzás
  - ▶ `<del>` - áthúzás
  - ▶ `<mark>` - **kihúzás**
  - ▶ `<abbr>` - Rövidítés magyarázata
- ▶ Kommentek: `<!--` Kommentek ide, több sorból is állhat `-->`
- ▶ HTML linkek:  
`<a href="http://www.w3schools.com/tags/default.asp/" target="_blank">Referencia</a>`
- ▶ Képek: ``  
alt: Szöveg megjelenítése, ha nem találja a képet

# Szöveg csoportosítása

- ▶ A szövegeket különböző szempontok szerint lehet csoportosítani
  - ▶ Az azonos stílusú bekezdések egyforma class névbe kerüljenek  
`<div class="csoport">szöveg</div>`
  - ▶ Az általános class tulajdonságnál több formázást igénylő szövegek kapjanak saját ID-t amivel majd lehet rájuk hivatkozni  
`<div class="csoport" id="különleges">szöveg</div>`  
Így öröklí a class stílus tulajdonságait, de emellett még több formázás is érvényes lehet rá
  - ▶ Az azonos stílusú szövegrészek egyforma span class névbe kerüljenek (nem kezd új sort!)  
`<p>ez egy <span class="kijelöl">fontos</span> rész</p>`

# A HTML formázás CSS

- ▶ A CSS (Cascading Style Sheets) a HTML elemek formázására, stílusára, megjelenítésére használatos nyelv
- ▶ 3 féleképpen lehet html-hez rendelni:
  - ▶ Inline attribútumként (`style="csstag:érték;"`)
  - ▶ A `<head>` fejlécen belül `<style>...</style>` tag-ek között
  - ▶ Külső fájlból (ez a legpraktikusabb, html kódtól függetlenül is módosíthatjuk a stílus leírásokat bármikor és más html-ben is újrahasznosítható, felhasználható)
- ▶ CSS-ben a különböző tag-ekre, classokra, ID-kre hivatkozunk így pl. az összes egyforma class-ú tag egyforma alaptulajdonságokat kapnak, de ezeken belül a egyéni ID-vel rendelkezők több tulajdonsággal is bírhatnak és így felül is írhatunk meglévő tulajdonságokat
- ▶ A `<head>` fejlécben `<link rel="stylesheet" href="stilus.css">` segítségével elérjük a külső css kódot
- ▶ Szintaxis: (a tagnév elhagyása ajánlott)  
tagnév vagy tagnév#ID vagy tagnév.class {  
    tulajdonság: érték;  
}
- ▶ <http://www.w3schools.com/cssref/default.asp>  
[http://www.w3schools.com/cssref/css\\_selectors.asp](http://www.w3schools.com/cssref/css_selectors.asp)

# CSS doboz modell, layout

- ▶ [http://www.w3schools.com/css/css\\_boxmodel.asp](http://www.w3schools.com/css/css_boxmodel.asp)
- ▶ A különböző összetartozó szövegeket/elemeket csoportosíthatjuk és vagy más objektumhoz képest relatív, vagy az oldalhoz képest abszolút módon elhelyezhetünk
- ▶ A <div> tag segítségével csoportokat definiálhatunk
- ▶ position tulajdonságot kell állítani.
- ▶ relative, absolute, static, fixed (képernyőre rögzített)



# CSS3

- ▶ Elképesztő dolgokat lehet csinálni a CSS3-ban, amiket a modern böngészők támogatnak
  - ▶ Selectors
  - ▶ Box Model
  - ▶ Backgrounds and Borders
  - ▶ Image Values and Replaced Content
  - ▶ Text Effects
  - ▶ 2D/3D Transformations
  - ▶ Animations
  - ▶ Multiple Column Layout
  - ▶ User Interface
- ▶ [http://www.w3schools.com/css/css3\\_intro.asp](http://www.w3schools.com/css/css3_intro.asp)



# HTML oldaltervezés

- ▶ A HTML/CSS eszközökkel a kezünkben képesek vagyunk szinte bármilyen kinézetű oldalt elkészíteni, csak a képzeletünk és az időnk szab határt
- ▶ Jó hozzáállás a honlap készítésnél az, hogy először átgondoljuk mit szeretnénk csinálni, kb. találjuk ki mi lesz a célja a honlapnak, hogy nézzen ki (mi legyen a fő stílus témája...)
- ▶ A HTML oldalunk vázát érdemes elsősre megcsinálni, majd a részletekbe menő (gyakorlati, komplex) problémákat részfeladatokra bontással érdemes megoldani