



Bevezetés a webes szoftverfejlesztésbe

HTML



Automatizálási és
Alkalmazott
Informatikai Tanszék

Gincsei Gábor
gincsei@aut.bme.hu

Egy kis történelem



Tim Berners-Lee

1991	HTML megjelenése
1995	HTML 2.0
1997	HTML 3.2
1999	HTML 4.01
2000	XHTML 1.0
2002 - 2009	XHTML 2.0
2012	HTML5

HTML 4.01 után XHTML irányába fordult a világ.

Az XHTML 2.0 sokkal szigorúbb. A nem szabványos weboldalakat nem szabad megjeleníteni. 2009-re rájöttek, hogy ez zsákutca.

HTML5 szabvány sokkal megengedőbb és visszafelé is kompatibilis maradt.

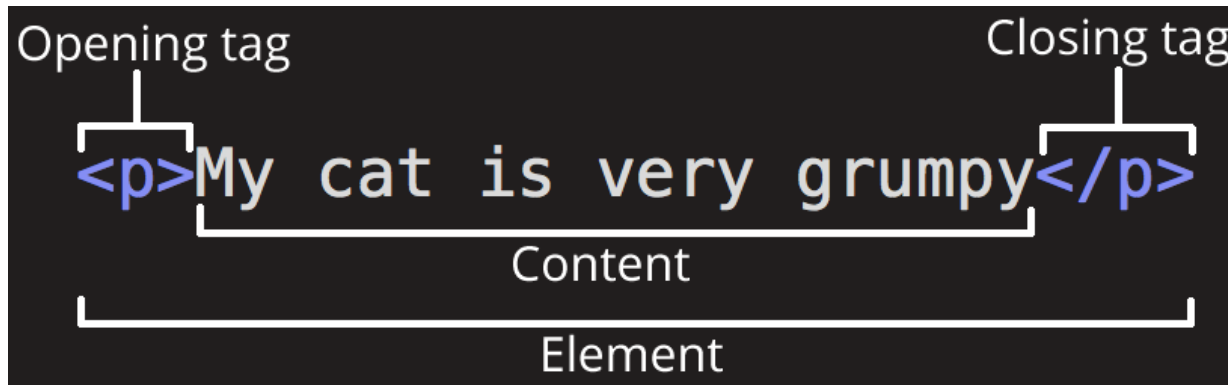
HTML5-öt 2012-ben publikálták, de 2004 óta fejlesztik.

Mi a HTML?

- Hypertext Markup Language
- Jelölőnyelv ami leírja a böngészőnek, hogyan épül fel a weboldal struktúrája.
- HTML elemekből épül fel,
 - > Ez határozza meg hogyan jelenjen meg vagy viselkedjen az adott rész.
 - > Pl: <button>,

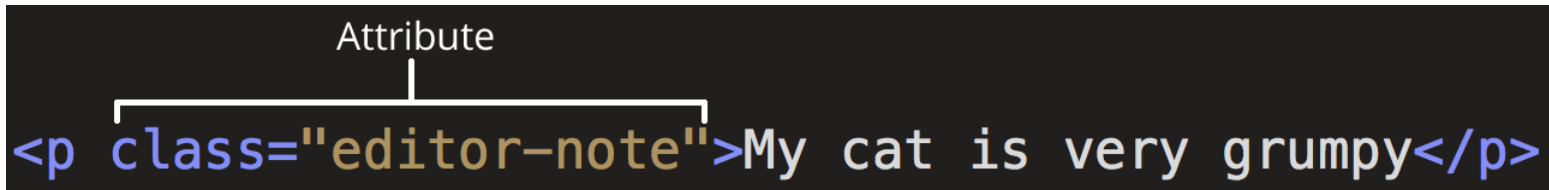
Hogyan épül fel egy HTML elem?

- Nyitó tag: Az elem neve < és > között.
 - > A tag nem kis-nagybetű érzékeny, de ajánlott kisbetűkkel írni.
- Lezáró tag: azonos a nyitó taggel, de </-el kezdődik.
- Tartalom: A két tag közötti rész, ami jelenleg szöveg.
 - > Ha a tag nem tartalmazhat semmit (üres tag), akkor lezáró tagje sincs.
 - > Pl: vagy
 DE <script src="..."></script>



Attribútumok

- Extra információt adnak az elemhez
 - > Pl.: Elem egyedi azonosítója, neve, CSS osztályok...
- A nyitó tagbe kell megadni, szóközzel elválasztva egymástól.
- Az attribútum neve után = jel következik
 - > De a bool értékű attribútumoknál elhagyható az érték
 - > Pl.: disabled="true", disabled="disabled" vagy disabled
- Az attribútum értékét idézőjelek közé tesszük
 - > Használhatjuk a " vagy a ' idézőjelet is.



```
<p class="editor-note">My cat is very grumpy</p>
```

Leggyakoribb attribútumok

- **id:** Egyedi azonosítója egy elemnek
 - > Később ez alapján tudjuk majd kiválasztani az elemet
 - > Ha több elemnek is azonos az id-ja az oldalon, nem kapunk hibát, de a jQuery hibásan fog működni.
- **title:** Az elem címe, általában tooltipként jelenik meg
- **class:** A megadott CSS osztály ráteszi az elemre
 - > Ehhez a <head>-ben linkelni kell a CSS fájlt is.
- **style:** CSS szabályt lehet közvetlenül az elemre tenni
 - > Ha lehet kerüljük az inline style-okat → class.

Block és inline elemek

- **Blokk** szintű elemek

- > Mindig új sorban jelennek meg.
- > Blokk elemet csak blokk szintűbe lehet beágyazni.
- > Pl.: paragrafus, lista, navigációs menü, lábléc

- **Inline** elemek

- > A böngésző ugyanabban a sorban az előtte lévő elem mögött jeleníti meg.
- > Blokk szintű elem tartalmának egy része
- > Pl.: paragrafuson belüli hivatkozások, vastag betűs szöveg

Miért van szükség a HTML5-re?

- Fő célja, hogy a webes alkalmazásokhoz ne legyen szükség pluginek használatára.
 - > Adobe Flash,
 - > Microsoft Silverlight,
 - > JavaFX
- Maradjon kompatibilis a korábbi verziókkal.
- Ügyeljünk rá, hogy bár a böngészők támogatják, de nem feltétlenül mindent és nem mindent ugyanúgy.
 - > Egyes funkciók a különböző böngészőkben eltérhetnek.
 - Pl.: dátum választó, validáció

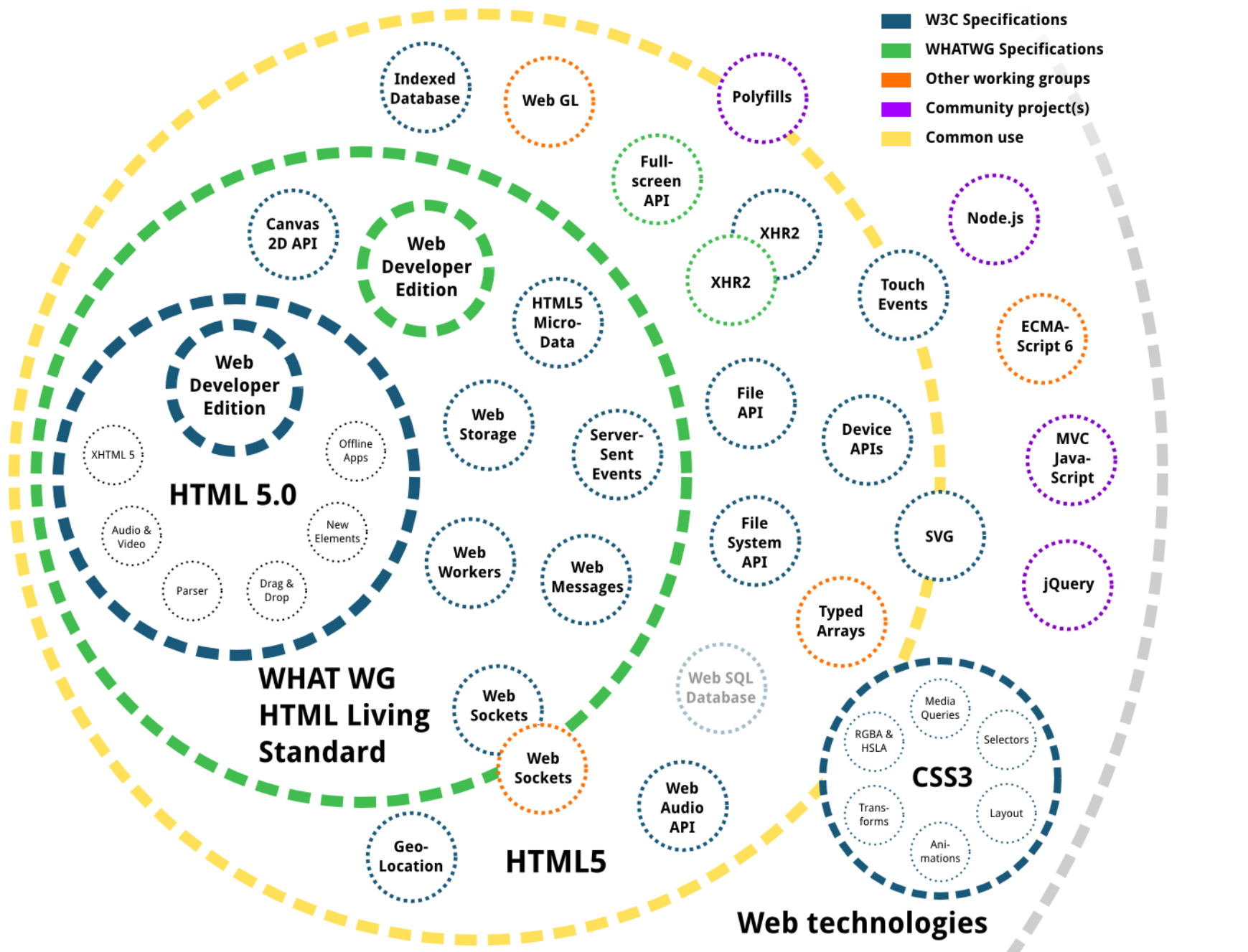
Böngésző támogatás

- Nincs 100%-ban „HTML5 kompatibilis” böngésző
 - > Némelyik ezt támogatja belőle, némelyik azt...
- Sokat ront a helyzeten, hogy a HTML5 mozgó célpont: bekerülnek, kikerülnek, változnak dolgok, még nem számít véglegesnek.
- Megoldás: külön-külön ellenőrizzük, hogy a támogatni szánt böngészőink mit tudnak, és mit nem, és aszerint járjunk el.
 - > Nem véletlen, hogy egy webes projekt első és legfontosabb kérdése, hogy mik a minimális böngészőverziók, amiket megcéloz...

Feature detection

- SOHA ne kérdezzük le a böngésző nevét vagy verzióját egy feature meglétének "eldöntésére"
- Helyette közvetlenül a feature meglétét kérdezzük le Modernizr-rel, vagy pár sor kóddal.

```
try {  
    return 'localStorage' in window  
        && window['localStorage'] !== null;  
} catch(e) {  
    return false;  
}
```



HTML5-ös újdonságok

- Új HTML tagek, amik az oldal struktúráját jobban leírják
- Forms 2.0
 - > Új beviteli mezők (új attribútumok az input taghez)
 - > Kliens oldali validáció (Javascript nélkül)
- Új API-k
 - > Storage (local és session): Adattárolás a böngészőben.
 - > Web sockets: Duplex kommunikáció.
 - > Canvas: 2D-s rajzolási lehetőség JavaScriptből.
 - > Audio/Video: Zene/videó lejátszása külső komponens nélkül.
 - > Web worker: Külön szálon futó JS támogatása.
 - > Geolokáció: Fizikai hely megosztásának lehetősége.

Helló világ HTML-ben

<> HelloVilag.html x

```
1  <!DOCTYPE html>
2  <html>
3      <head>
4          <title>Oldal címe</title>
5      </head>
6      <body>
7          Helló világ!
8      </body>
9  </html>
```

<!DOCTYPE> és <html>

<!DOCTYPE html>

- A doctype mondja meg a böngészőnek, hogy milyen típusú és verziójú HTML-t használ az oldal.

<html lang="hu">

- A <html> az oldal gyökéreleme, ezen belül található minden.
 - > Itt adhatjuk meg, hogy az oldal milyen nyelven készült, ami SEO szempontból fontos.

<head>

- A head szekcióban adhatunk meg az oldalra vonatkozó meta adatokat.
 - > Karakter kódolás, oldal címe, cachelés megadása...
- Itt hivatkozhatunk CSS fájlokra
 - > JavaScript fájlokat a body lezáró tag előtt érdemes.

<head>

```
<meta charset="utf-8">
```

```
<title>Első HTML5 oldalam</title>
```

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

```
<script src="script.js"></script>
```

</head>

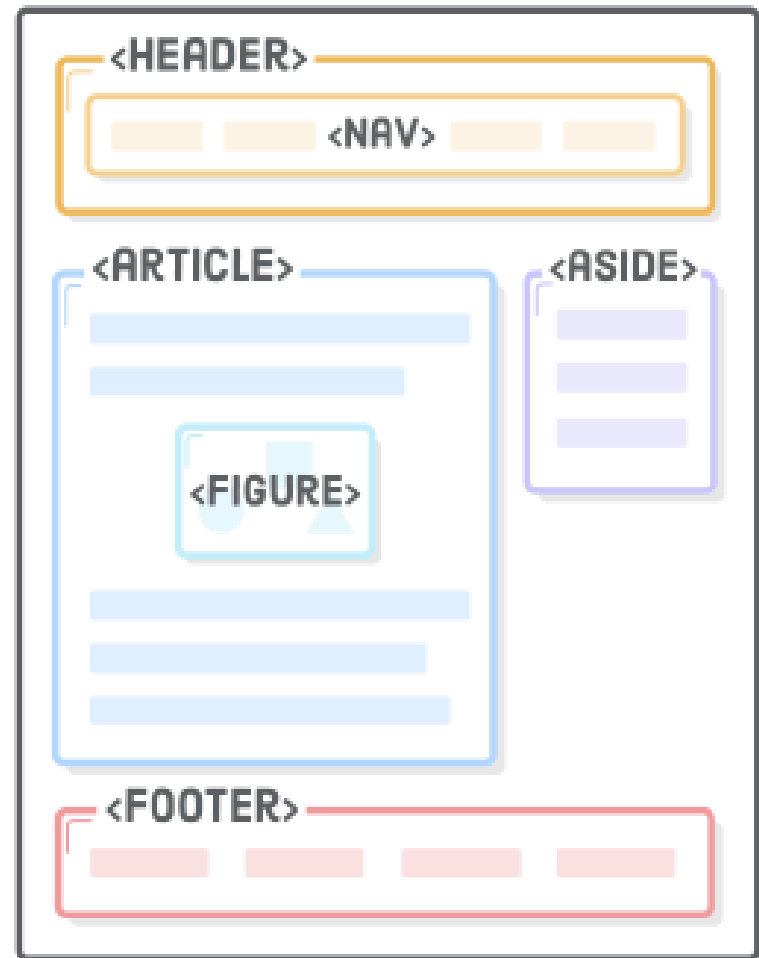
Metaadatokkal bővített Helló Világ

<> HelloVilagMeta.html x

```
1  <!DOCTYPE html>
2  <html lang="hu">
3  <head>
4      <meta charset="utf-8">
5      <meta name="viewport" content="width=device-width, initial-scale=1">
6      <title>Első HTML5 oldal</title>
7      <link rel="stylesheet" href="style.css">
8  </head>
9  <body>
10     <p>Helló világ!</p>
11     <!-- TODO: Oldal kódja -->
12
13     <script src="script.js"></script>
14 </body>
15 </html>
```

Oldalváz

- **<header>**
 - > Az oldal vagy egy section fejléce.
- **<nav>**
 - > Navigációs lineket fogja össze
- **<aside>**
 - > Oldaljegyzet, ami a mellette lévő tartalomhoz nem kapcsolódik.
- **<section>**
 - > Az oldalt logikai egységre bontására.
- **<article>**
 - > Egy önállóan értelmezhető része az oldalnak.
 - > Pl.: cikk, hozzászólás...
- **<footer>**
 - > Az oldal, vagy akár egy article lábléce.

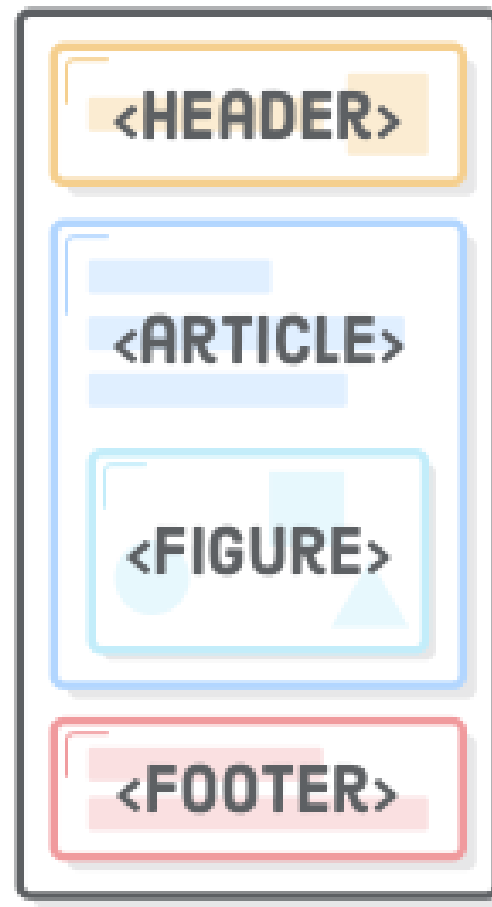


DIV vs szemantikus HTML5?

Csak div-ek

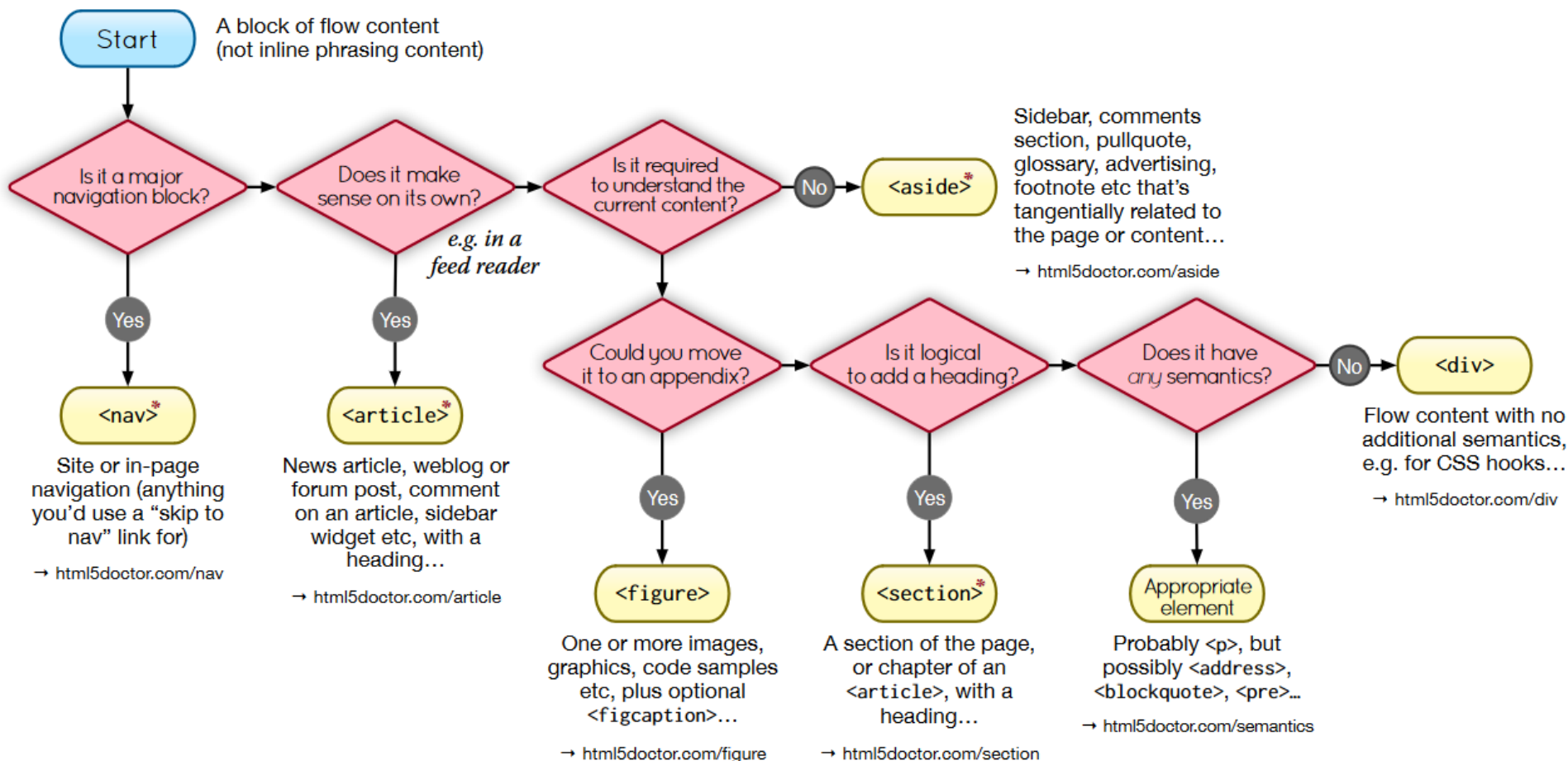


HTML 5 szemantikus



Miért jobb a HTML5-ös oldalváz?

- Az egyes elemek jelentéssel bírnak
 - > A `<div id=„...”` -nak nincs semmi extra jelentése
 - > A böngészők, keresőmotorok, felolvasó szoftverek értelmezhetik → értelmezhető az oldal felépítése.
 - > Több szemantikai információ.
- A Google a headerben megadott értéken fontosabbnak tartja mint ami a footerben van.
- A `<nav>`-ban megadott elemekről tudjuk, hogy hivatkozások. Egy felolvasó szoftvernek ez sokat segíthet.
- Megadja az oldal struktúráját.



* Sectioning content element

These four elements (and their headings) are used by HTML5's outlining algorithm to make the document's outline
→ html5doctor.com/outline

2011-07-22 v1.5
For more information:
www.html5doctor.com/semantics

Oldalak közti navigáció (hyperlink)

- Az oldalak közötti navigációra az <a> tag

```
<a href="http://www.aut.bme.hu">AUT portál</a>
```

- Link megnyitása új tabon

```
<a href="http://www.aut.bme.hu"  
  target="_blank">AUT portál</a>
```

- Levelező program megnyitása a kliensen

```
<a href="mailto:xy@example.com?Subject=Hello">  
  Levél küldése!</a>
```

DEMO

Oldalváz felépítése

HTML5 szemantikus elemek használata

<https://gsnedders.html5.org/outliner/>



Automatizálási és
Alkalmazott
Informatikai Tanszék

Az oldalunk felépítése

Heading-level outline

- <h1> Mobil- és webes szoftverek - MyBlog**
- <h2> --- Navigáció ---**
- <h2> Itt a .NET Core 2.0, és vele új Entity Framework Core és Visual Studio is**
- <h2> Korábbi blog bejegyzések**
 - <h3> .NET és a WebAssembly**
 - <h3> Monospaced fontok programozáshoz**
 - <h3> URL nyomtatásban :)**
 - <h3> Feliratkozás a hírlevélre**

Structural outline

- Mobil- és webes szoftverek - MyBlog**
 - Navigáció ---**
 - Itt a .NET Core 2.0, és vele új Entity Framework Core és Visual Studio is**
 - [section element with no heading]
 - [section element with no heading]
 - [section element with no heading]
 - Korábbi blog bejegyzések**
 - .NET és a WebAssembly**
 - Monospaced fontok programozáshoz**
 - URL nyomtatásban :)**
 - Feliratkozás a hírlevélre**

HTML oldalon hibák / figyelmeztetések

Warning Section lacks heading. Consider using `h2` - `h6` elements to [add identifying headings to all sections](#).

[From line 44, column 9; to line 44, column 27](#)

```
<section id="meta">
```

Warning Section lacks heading. Consider using `h2` - `h6` elements to [add identifying headings to all sections](#).

[From line 53, column 9; to line 53, column 35](#)

```
<section id="introduction">
```

Warning Section lacks heading. Consider using `h2` - `h6` elements to [add identifying headings to all sections](#).

[From line 68, column 9; to line 68, column 30](#)

```
<section id="content">
```

Warning The `date` input type is not supported in all browsers. Please be sure to test, and consider using a polyfill.

[From line 130, column 17; to line 130, column 70](#)

```
<input id="txtBirthDate" name="birthDate" type="date">
```

HTML űrlap <form>

- A felhasználótól származó adatok továbbítása a szerver felé.
 - > beviteli mezők (input)
 - > gombok (button / <input type="submit">
 - > csoportba foglalhatjuk őket (fieldset / legend)

```
<form action="action_page.php" method="get">  
  <label for="name">Név:</label>  
  <input type="text" id="name"  
        name="name" value="Gincsei Gábor">  
  
  <br>  
  <input type="submit" value="Küldés">  
</form>
```

<input> tag típusok

- Egyszerű beviteli mező
 - > text / password / number (A többsoros más tag: <textarea>)
- Választós mezők
 - > radio / checkbox
- Gombok
 - > button / submit / reset (használható a <button> tag is)
- Fájl
 - > File
- Dátum típusok
 - > date / datetime/ datetime-local / month / time / week
- Egyéb gyakori típusok
 - > email / range / search / tel / url / color

Beviteli mező előtti címke: <label>

- Cél: a megjelenített szöveg és a beviteli mező összetartozzon.
 - > A címkére kattintva a fókuszt a beviteli mezőre ugrik
- Egyszerűen a beviteli mezőt a label-be ágyazzuk.

<label>Név:

<input type="text" id="fullName">

</label>

- A for attribútummal megadjuk, hogy melyik inputhoz tartozik.
 - > JS-ből a htmlFor-al kérdezhető le, hogy melyik beviteli mezőhöz tartozik a címke (for értéke)

<label for="fullName">Név:</label>

<input type="text" id="fullName">

Fontosabb input attribútumok

- Helyőrző szöveg megadása

```
<input type="text" placeholder="keresés">
```

- Csak olvasható input

```
<input type="text" readonly>
```

- Letiltott elem

```
<input type="text" disabled>
```

- Korábbi értékeket ne kínálja fel

```
<input type="text" autocomplete="off">
```

- Automatikusan kapjon fókuszt

```
<input type="text" autofocus>
```

Legördülő lista: <select>

- Az option tagekkel adhatók meg az elemek
- Az optgrouppal tudjuk őket csoportosítani.

<select>

```
<optgroup label="Swedish Cars">
```

```
  <option value="volvo">Volvo</option>
```

```
  <option value="saab">Saab</option>
```

```
</optgroup>
```

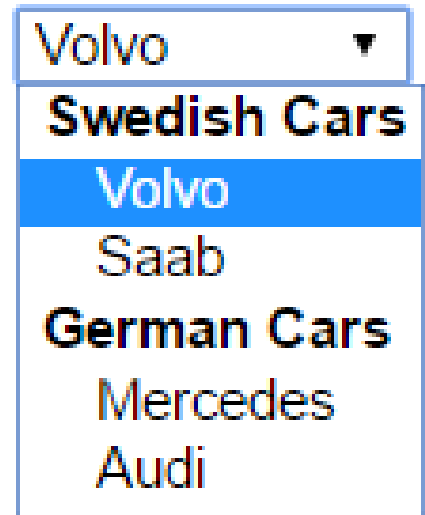
```
<optgroup label="German Cars">
```

```
  <option value="mercedes">Mercedes</option>
```

```
  <option value="audi">Audi</option>
```

```
</optgroup>
```

```
</select>
```



Volvo ▼
Swedish Cars
Volvo
Saab
German Cars
Mercedes
Audi

Szűrhető lista: <datalist>

```
<input list="browsers">
```

```
<datalist id="browsers">
```

```
  <option value="Internet Explorer">
```

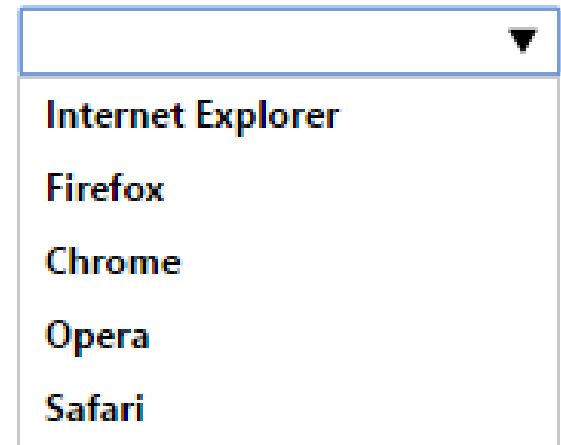
```
  <option value="Firefox">
```

```
  <option value="Chrome">
```

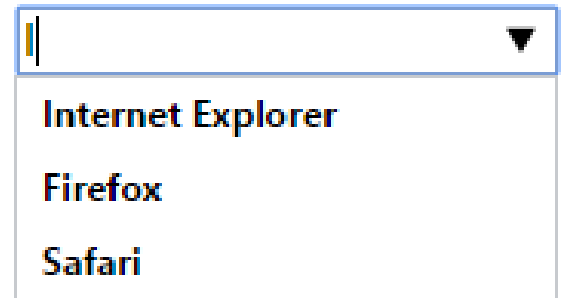
```
  <option value="Opera">
```

```
  <option value="Safari">
```

```
</datalist>
```



A screenshot of a web browser's dropdown menu. The menu is open, showing a list of browser names: Internet Explorer, Firefox, Chrome, Opera, and Safari. The menu is styled with a light blue border and a white background.



A screenshot of a web browser's dropdown menu. The menu is open, showing a list of browser names: Internet Explorer, Firefox, and Safari. The menu is styled with a light blue border and a white background.

Fájl feltöltés <input type="file">

- Olyan input amivel fájlt lehet kiválasztani feltöltéshez.

```
<input type="file">
```

- Formon (enctype) / inputon a formenctype megadása

```
<input formenctype="multipart/form-data">
```

```
<form ... enctype="multipart/form-data">
```

- Több fájl feltöltésére multiple

```
<input type="file" multiple>
```

- Kiválasztott fájlok átadása JS-nek

```
<input type="file" multiple  
onchange="handleFile(this.files)">
```

> https://developer.mozilla.org/en/docs/Using_files_from_web_applications

Validáció célja

- A felhasználónál érjük el, hogy kitöltse az űrlapot
 - > Kötelező mezők definiálása
 - > Adat típusok definiálása (szám / email ...)
 - > Tetszőleges formátum reguláris kifejezéssel
- Mindezt HTML-ből JavaScript nélkül
- Ha egy fomat nem kell validálni: **novalidate**
- Csak submit-kor fut a validáció

Validációs attribútumok

- Kötelezően kitöltendő mező

```
<input type="text" required>
```

- Validáció reguláris kifejezéssel

```
<input type="text" name="code"  
      pattern="[A-Za-z]{2}"  
      title="Két betűs országkód">
```

- Min/max hossz (karakterszám)

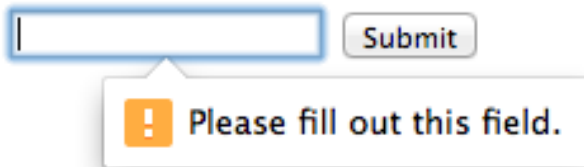
```
<input type="text" maxlength="20">
```

- Min/max érték és a lépésköz megadása

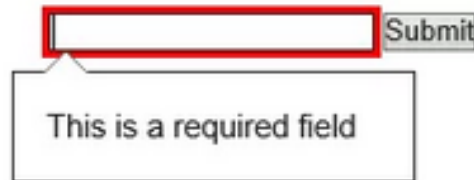
```
<input type="number" min="1" max="10" step="1">
```

Hogyan jelenik meg a böngészőben?

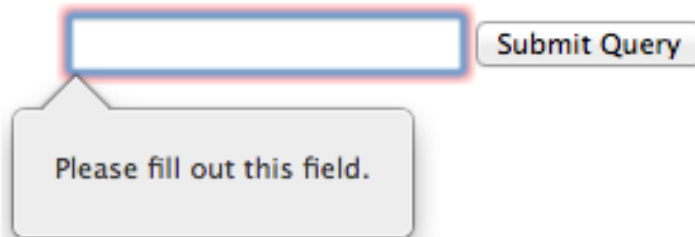
Chrome 30



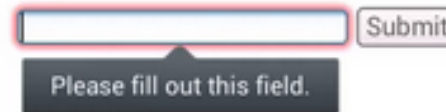
Internet Explorer 10



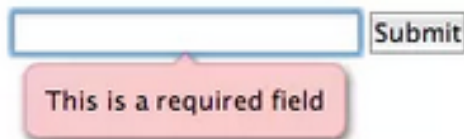
Firefox 23



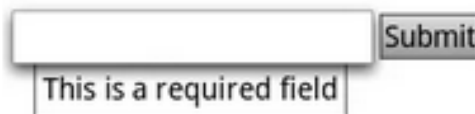
Firefox for Android



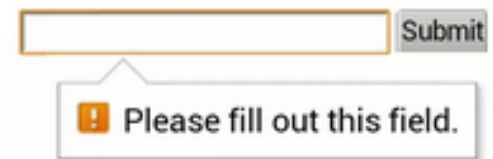
Opera 12



Opera Mobile



Chrome for Android



Validációs API

- Validációt JS kódból is kiválthatjuk
 - > **willValidate**: Fogja-e validálni a mezőt az űrlap elküldésekor.
 - > **validity**: Egy `ValidityState` objectet ad vissza, amiben minden típusú validációhoz megkajuk, hogy érvényes vagy nem.
 - Pl: `customError`, `tooLong`, `valueMissing`, `valid`.
 - > **validationMessage**: Hibaüzenet amit a böngésző megjelenít ha az adott elem nem érvényes.
 - > **checkValidity()**: Validálja az adott mezőt, vagy ha az űrlapon hívjuk, akkor a teljes űrlapot validálja.
 - > **setCustomValidity()**: A validációs hibaüzenetet lehet vele lecserélni.

DEMO

Űrlapkezelés és HTML validáció



Automatizálási és
Alkalmazott
Informatikai Tanszék

Zene lejátszása HTML-ből

- Támogatott formátumok

- > mp3
- > wav (IE kivételével)
- > Ogg (IE és Safari kivételével)

Element					
<video>	4.0	9.0	3.5	4.0	10.5

<audio controls>

<source src="horse.ogg" type="audio/ogg">

<source src="horse.mp3" type="audio/mpeg">

A böngésző nem támogatja az audio taget.

</audio>

Video lejátszása HTML-ből

- Támogatott formátumok

- > mp4

- > webM (Chrome, FF, Opera)

- > Ogg (Chrome, FF, Opera)

Element					
<video>	4.0	9.0	3.5	4.0	10.5

```
<video width="320" height="240" controls autoplay>
```

```
<source src="movie.mp4" type="video/mp4">
```

```
<source src="movie.ogg" type="video/ogg">
```

A böngésző nem támogatja a video lejátszását.

```
</video>
```