

A Web jelölőnyelvei

Jeszenszky Péter

2024. október 16.

A Web jelölőnyelvei

- HTML
- SVG
- MathML
- Markdown (külön tárgyaljuk)

- “A HTML a Web elsődleges leíró nyelve.”
- “[...] egy szemantikai szintű leíró nyelv és a kapcsolódó szemantikai szintű alkalmazásprogramozási interfészek a Weben elérhető oldalak készítéséhez, melyek a statikus dokumentumoktól a dinamikus alkalmazásokig terjednek.”
 - Lásd: [HTML Living Standard](#)

A HTML születése

- Tim Berners-Lee (TBL) találta fel és alkotta meg a HTML-t.
- Az első nyilvános webhely: <http://info.cern.ch/> (indulás: 1991. augusztus 6.)
 - Lásd: [Restoring the first website](#)
- Az első nyilvános műszaki dokumentáció a HTML-ről:
<https://info.cern.ch/hypertext/WWW/MarkUp/Tags.html>

Főbb HTML verziók (1)

- HTML 4.01 (hatálytalanítva)
- XHTML 1.0 (hatálytalanítva)
- XHTML 1.1 (hatálytalanítva)
- HTML5 (az aktuális verzió)

Főbb HTML verziók (2)

Verzió statisztikák:

- W3Techs: [Usage statistics and market share of HTML for websites](#)
HTML5 is used by 96.3% of all the websites who use HTML.

HTML 4.01 (1)

[HTML 4.01 Specification](#) (W3C ajánlás, 1999. december 24.; hatálytalanítva: 2018. március 27.)

- Az utolsó SGML-alapú HTML verzió.
- Média típus: `text/html`

HTML 4.01 (2)

Dokumentumtípus-deklarációk:

- **Strict:**

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN"  
  "http://www.w3.org/TR/html4/strict.dtd">
```

- **Transitional:**

```
<!DOCTYPE HTML PUBLIC  
  "-//W3C//DTD HTML 4.01 Transitional//EN"  
  "http://www.w3.org/TR/html4/loose.dtd">
```

- **Frameset:**

```
<!DOCTYPE HTML PUBLIC  
  "-//W3C//DTD HTML 4.01 Frameset//EN"  
  "http://www.w3.org/TR/html4/frameset.dtd">
```

- Az XML alkalmazásként definiált HTML szigorúbb szabályokat ír elő a dokumentumok számára, így azok feldolgozása egyszerűbb.
- Különösen lényeges ez a hagyományos asztali gépekhez képest korlátozott lehetőségekkel bíró eszközökénél (például mobil eszközöknél).
- Az XHTML illetve annak modularizációja lehetővé teszi az XHTML kombinálását más XML alkalmazásokkal.
 - Például MathML és SVG beágyazás XHTML dokumentumokba – ezek a dokumentumok a továbbiakban azonban már nem XHTML dokumentumok.

XHTML 1.0 (1)

XHTML 1.0 The Extensible HyperText Markup Language (Second Edition) – A Reformulation of HTML 4 in XML 1.0 (W3C ajánlás, 2000. január 26.; hatálytalanítva: 2018. március 27.)

- A HTML 4 újrafogalmazása XML alkalmazásként.
- Média típus: `application/xhtml+xml`

XHTML 1.0 (2)

Dokumentumtípus-deklarációk:

- **Strict:**

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"  
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
```

- **Transitional:**

```
<!DOCTYPE html PUBLIC  
"-//W3C//DTD XHTML 1.0 Transitional//EN"  
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
```

- **Frameset:**

```
<!DOCTYPE html PUBLIC  
"-//W3C//DTD XHTML 1.0 Frameset//EN"  
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-frameset.dtd">
```

XHTML 1.1 (1)

XHTML Modularization 1.1 – Second Edition (W3C ajánlás, 2010. július 29.; hatálytalanítva: 2018. március 27.)

- A modularizáció lehetővé teszi az XHTML résznyelveinek definiálását és az XHTML kiterjesztését.
- Megvalósítható a DTD vagy XML Schema felhasználásával is.
- Szabványos modulok egy készletét biztosítja.
 - Például: *Frames* (frame, frameset, noframes elemek), *Hypertext* (a elem), *Text* (div, h1, p, ... elemek), ...
- Több modul kombinálása révén úgynevezett hibrid dokumentumtípusok létrehozását teszi lehetővé.

XHTML 1.1 (2)

XHTML Basic 1.1 – Second Edition (W3C ajánlás, 2010. november 23.; hatálytalanítva: 2018. március 27.)

- Az XHTML egy olyan részhalmaza, amely számos különböző eszköz számára alkalmas (mobiltelefonok, PDA-k, elektronikus könyvolvasók, tv-készülékek, ...).
- A dokumentumtípus-definíció megvalósítása modulok segítségével az *XHTML Modularization* ajánlásban foglaltak szerint.
- Dokumentumtípus-deklaráció:

```
<!DOCTYPE html PUBLIC  
  "-//W3C//DTD XHTML Basic 1.1//EN"  
  "http://www.w3.org/TR/xhtml-basic/xhtml-basic11.dtd">
```

- Média típus: `application/xhtml+xml`

XHTML 2.0

XHTML 2.0 (W3C munkacsoport feljegyzés, 2010. december 16.)

- Egy új nyelv, mely nem volt kompatibilis visszafelé a korábbi HTML és XHTML nyelvekkel.
- A W3C úgy döntött, hogy nem folytatják tovább a fejlesztést, így nem lesz XHTML 2.0 ajánlás.
- A HTML5 vette át eredetileg kitűzött szerepét.
- Lásd: [Frequently Asked Questions \(FAQ\) about the future of XHTML](#)

HTML5: történet

Lásd: [HTML Living Standard – Introduction – History](#)

HTML5: fejlesztés (1)

- Eredetileg a WHATWG fejlesztette ki a HTML5 specifikációt.
- A W3C 2007-ben kapcsolódott be a HTML5 fejlesztésébe.
- Lásd:
 - Ian Hickson. [The WHATWG Blog – W3C restarts HTML effort.](#) 7 March 2007.
 - [W3C Relaunches HTML Activity.](#) 7 March 2007.

HTML5: fejlesztés (2)

2012 júliusa és 2019 júniusa között a WHATWG és a W3C is külön specifikációt fejlesztett, melynek során eltérő fejlesztési modellt követtek.

- A W3C a HTML 5.0 specifikáció ajánlasként való kiadása után a következő verzióon dolgozott (HTML 5.1).
- A WHATWG specifikációja soha nem lesz lezárt, folyamatosan fejlesztik (“élő szabvány”).

HTML5: fejlesztés (3)

- 2019 júniusáig a W3C-n belül a Web Platform Munkacsoport fejlesztette a HTML nyelvhez kapcsolódó specifikációkat.
- A W3C specifikációi a WHATWG specifikációján alapultak.
 - A W3C bizonyos részeket külön dokumentumokba emelt ki.

HTML5: fejlesztés (4)

- A két szervezet 2019. május 28-án aláírt egy megállapodást arról, hogy együttműködnek a HTML és DOM specifikációk egyetlen verziójának kifejlesztésén.
 - Lásd: Jeff Jaffe. [W3C and WHATWG to work together to advance the open Web platform](#). 28 May 2019.
- Együttműködési megállapodás:
 - A HTML-t és a DOM-ot elsősorban a WHATWG fejleszti.
 - A W3C ajánlasként szándékozik jóváhagyni és kiadni a WHATWG specifikációkat.
 - Lásd: [Memorandum of Understanding Between W3C and WHATWG](#). May 28, 2019.
- A továbbiakban a W3C-n belül a [HTML Munkacsoport](#) felelős a HTML fejlesztéséért.

HTML5 szabvány

- **WHATWG:**
 - HTML Living Standard
 - HTML: The Living Standard – Edition for Web Developers
 - Nem tartalmazza a csupán a böngészőgyártók számára szóló információkat.
- **W3C:** a <https://www.w3.org/TR/html> URI jelenleg átirányít a WHATWG specifikációra.

HTML elemek (1)

- Az elemeknek, attribútumoknak és attribútumértékeknek meghatározott jelentése (szemantikája) van.
 - Például az `ol` elem egy rendezett listát ábrázol, a `lang` attribútum pedig a tartalom nyelvét ábrázolja.
- A szerzők számára tilos az elemek, attribútumok és attribútumértékek a megfelelő rendeltetésüktől eltérő jelentésbeli céllal történő használata.

HTML elemek (2)

- A HTML előző verzióiban rendelkezésre álló prezentációs lehetőségek többsége többé nem megengedett.
- A prezentációs jelölők problémái:
 - A prezentációs elemek használata rontja a hozzáférhetőséget.
 - Magasabb karbantartási költségek.
 - Nagyobb dokumentumméret.

HTML elemek (3)

Csak a `style` attribútum és a `style` elem maradt meg, mint prezentációs jelölési lehetőség.

HTML elemek (4)

Média-függetlenként lettek újrafogalmazva a következő, korábban prezentációs elemek:

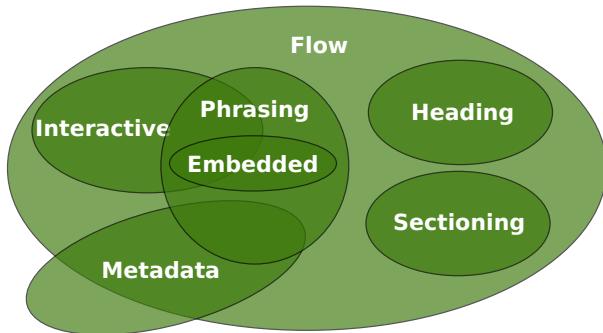
Elem	HTML 4.01, XHTML	
	1.0	HTML5
b	Félkövér betű	Kulcsszavak
<i>i</i>	Kurzív betű	Hangnembeli változás
hr	Vízszintes választóvonal	Témaváltás
<small>small</small>	Kisebbs betűméret	Lapszéli megjegyzés
s	Áthúzás	Pontatlan szöveg
u	Aláhúzás	Artikulálatlan jelölés (például hibásan írt szöveg)

HTML elemek (5)

- A specifikációban definiált minden egyes elemnek van egy tartalommodellje (az elem szükséges tartalmának egy leírása).
- Egy HTML elem tartalma meg kell, hogy feleljen a tartalommodelljében leírt követelményeknek.

HTML elemek (6)

Minden egyes HTML elem nulla vagy több kategóriába sorolható az alábbiak közül:



Ábra 1: Forrás: <https://html.spec.whatwg.org/images/content-venn.svg>

HTML elemek (7)

Idegen elemek (*foreign elements*): A MathML és SVG névterekbe tartozó elemek.

HTML elemek (8)

A HTML5-ben a jobb tagoláshoz bevezetett elemek:

- `article`
- `aside`
- `figure`
- `footer`
- `header`
- `hgroup`
- `nav`
- `section`

HTML elemek (9)

A HTML5-ben bevezetett további új elemek:

- audio
- canvas
- data
- dialog
- math
- meter
- picture
- progress
- summary/details
- time
- video
- ...

HTML elemek (10)

Az összes HTML elem:

- [HTML Standard – Index – Elements](#)
- [MDN Web Docs – HTML Elements Reference](#)

HTML elemek (11)

Példa: <https://html-basics.surge.sh/hu>

Szemantikus HTML (1)

Példa:

```
<h1>This is a top-level heading</h1>  
<span style="font-size: 2em; margin: 0.67em 0;">  
  This is not a top-level heading,  
  although it looks like that  
</span>
```

Forrás: [Semantics in HTML](#)

Szemantikus HTML (2)

Előnyök:

- Jó a keresőoptimalizálás (*search engine optimization* – SEO) számára
- Javítja a karbantarthatóságot
- Akadálymentesség

Lásd: [HTML: A good basis for accessibility](#)

Szemantikus HTML (3)

Kapcsolódó fogalom: **akadálymentesség** (*web accessibility*)

- Az akadálymentesség azt jelenti, hogy a webhelyeket, eszközöket és technológiákat úgy tervezik és fejlesztik, hogy azokat fogyatékkal élő embertársaink is használhassák.
- Lásd: <https://www.w3.org/mission/accessibility/>

Globális HTML attribútumok

Valamennyi HTML elemhez megadható attribútumok:

- `class`
- `dir`
- `id`
- `lang`
- `style`
- `title`
- `xml:lang` (csak az XML szintaxisban ajánlott használni)
- Egyéni adat attribútumok (*custom data attributes*)
- ...

Lásd:

https://developer.mozilla.org/en-US/docs/Web/HTML/Global_attributes

class attribútum

- A szerzők az elemek kiterjesztéséhez használhatják a class attribútumot, gyakorlatilag saját elemeket létrehozva úgy, hogy közben a legalkalmasabb létező HTML elemet használják.
- A HTML elemeken megadott class attribútum értéke olyan tokenek egy szóközzel elválasztott listája, melyek azokat a különféle osztályokat ábrázolják, melyekhez az elem tartozik.
- Példák:

```
<p class="author">...</p>  
<p class="note">...</p>  
<p class="warning">...</p>  
<p class="note important">...</p>
```

Egyéni adat attribútumok (1)

- Egy egyéni adat attribútum egy olyan attribútum, melynek neve a data- karakterlánccal kezdődik, melyet legalább egy karakter követ a kötőjel után.
- Olyan egyéni adatok, állapot, annotációk és más hasonlóak az oldal vagy alkalmazás számára történő privát tárolására szolgálnak, melyekhez nincsenek megfelelőbb elemek vagy attribútumok.
- Minden HTML elemhez tetszőleges számú egyéni adat attribútum adható meg tetszőleges értékkel.

Egyéni adat attribútumok (2)

- Példa:

```
<form>
  <p>
    <label for="house">Ház:</label>
    <select name="house" id="house" required>
      <option value="">--Válassz egy lehetőséget--</option>
      <option value="gryffindor"
        data-color-primary="red"
        data-color-secondary="gold">Griffendél</option>
      <!-- ... --->
    </select>
  </p>
</form>
```

Egyéni adat attribútumok (3)

- Példa:

```
document.querySelectorAll('a').forEach(function (element) {
  element.addEventListener('mouseenter', function (event) {
    const a = event.currentTarget;
    const timer = setTimeout(function () {
      window.location.href = a.href;
    }, 3000);
    // Időzítő tárolása a data-timer attribútumban:
    a.dataset.timer = timer;
  });
  element.addEventListener('mouseleave', function (event) {
    // Időzítő lekérése a data-timer attribútumból:
    const timer = event.currentTarget.dataset.timer;
    clearTimeout(timer);
  });
});
```

HTML szintaxisok (1)

A HTML egy dokumentumok leírására szolgáló absztrakt nyelvet határoz meg, az ezt az absztrakt nyelvet használó erőforrások átviteléhez pedig két konkrét szintaxist.

HTML szintaxisok (2)

- **HTML szintaxis:**

- Bár nagyon hasonlít az SGML-hez és az XML-hez, egy külön nyelv saját feldolgozási szabályokkal.
- Kompatibilis a legtöbb ősi böngészővel.
- Fájlkiterjesztés: `.html`, `.htm`
- Média típus: `text/html`

- **XML szintaxis:**

- Az XML 1.0 és a *Namespaces in XML 1.0* szabványokon alapuló szintaxis.
- Nem határoz meg az további szintaktikai követelményeket az XML-hez előírtakon túl.
- XHTML szintaxisnak is nevezik.
- Fájlkiterjesztés: `.xhtml`, `.xht`
- Média típus: `application/xhtml+xml`

HTML: a HTML szintaxis (1)

Példa:

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="utf-8">
    <title>Sample Page</title>
    <link rel="stylesheet" href="style.css">
  </head>
  <body>
    <p>Hello, World!</p>
  </body>
</html>
```

HTML: a HTML szintaxis (2)

Kötelező a dokumentumtípus-deklaráció.

HTML: a HTML szintaxis (3)

- Speciális karakterek:
 - Elem szövege nem tartalmazhat '<' karaktert vagy félreérthető '&' karaktert.
 - Attribútumérték nem tartalmazhat félreérthető '&' karaktert.
- Félreérthető '&' karakter:
 - Egy olyan '&' karakter, melyet egy vagy több ASCII alfanumerikus karakter és egy ';' karakter követ, melyek nem felelnek meg a szabvány által definiált nevesített karakterhivatkozások egyikének sem (például `&nosuchchar;`).
 - Lásd: [Named character references](#)

HTML: a HTML szintaxis (4)

Az elem- és attribútumnevek kisbetű-nagybetű érzéketlenek.

- Az elemek és attribútumok neveinek megadásakor (még az idegen elemeknél is) tetszőlegesen keverhetők a kis- és nagybetűk.
- Ugyanabban a nyitó címkében soha nem fordulhat elő két vagy több olyan attribútum, melyek nevei kisbetű-nagybetű érzéketlen hasonlítás esetén megegyeznek.

HTML: a HTML szintaxis (5)

Nem idézett attribútumérték szintaxis:

- Ha egy, az üres karakterlánctól különböző attribútumérték nem tartalmaz egyetlen literális *whitespace* karaktert sem, akkor megadható az attribútumérték-határolók elhagyásával.
- Ekvivalensek például az alábbiak:

```
<input value="yes">  
<input value=yes>
```

HTML: a HTML szintaxis (6)

Logikai attribútumok:

- Sok attribútum logikai.
- Egy logikai attribútum jelenléte egy elemen az igaz értéket ábrázolja, hiánya pedig a hamis értéket.
- Ha megjelenik az attribútum, akkor értéke az üres karakterlánc kell, hogy legyen, vagy egy olyan érték, mely kisbetű-nagybetű érzéketlen hasonlítás esetén megegyezik az attribútum nevével.
- Ekvivalensek például az alábbiak:

```
<input type=checkbox checked name=agree disabled>  
<input type=checkbox checked=checked name=agree  
  disabled=disabled>  
<input type='checkbox' checked='' name="agree"  
  disabled="">
```

HTML: a HTML szintaxis (7)

Üres (*void*) elemek:

- Csak nyitó címkéjük van, tilos hozzájuk záró címke megadása.
- Például: `br`, `img`, `input`, `link`, `meta`, ...

HTML: a HTML szintaxis (8)

Az idegen elemeknek vagy egy nyitó és egy záró címkéjük van, vagy egy önlezáróként jelölt nyitó címkéjük, mely esetben nem lehet záró címkéjük.

- Ekvivalensek például az alábbi SVG elemek:

```
<circle cx="50" cy="50" r="50"></circle>  
<circle cx="50" cy="50" r="50"/>
```

HTML: a HTML szintaxis (9)

Opcionális címkék:

- Bizonyos elemek nyitó és záró címkéi elhagyhatók.
 - Egy elem nyitó címkéjének elhagyása az itt tárgyalt esetekben nem azt jelenti, hogy az elem nem nincs ott (feltételezeten, de ott van)!
 - Például egy HTML dokumentumnak mindig van egy gyökéreleme, még akkor is, ha a `<html>` karakterlánc egyáltalán nem jelenik meg benne.
- Lásd: [Optional tags](#)

HTML: a HTML szintaxis (10)

Opcionális címkék (folytatás):

- Elhagyható egy `li` elem záró címkéje, ha az elemet közvetlenül egy másik `li` elem követi vagy nincs több tartalom a szülő elemben.
 - Ekvivalensek például az alábbiak:

- ```

 Apple
 Banana
 Cherry

```

- ```
<ul>
  <li>Apple
  <li>Banana
  <li>Cherry
</ul>
```

HTML: a HTML szintaxis (11)

Opcionális címkék (folytatás):

- Elhagyható a `html` elem nyitó címkéje, ha elsőként nem egy megjegyzést tartalmaz.
- Elhagyható a `html` elem záró címkéje, ha nem egy megjegyzés követi közvetlenül.
- ...

HTML: a HTML szintaxis (12)

Opcionális címkék (folytatás):

- Ha nem lényegesek az elemek közötti whitespace karakterek, ekvivalensek például az alábbiak:

- ```
<!DOCTYPE html>
<html>
 <head>
 <title>Sample Page</title>
 </head>
 <body>
 <p>Hello, World!</p>
 </body>
</html>
```

- ```
<!DOCTYPE html>
<title>Sample Page</title>
<p>Hello, World!</p>
```

HTML: a HTML szintaxis (13)

- Nem támogatottak a névtér-deklarációk, még idegen elemekhez sem.
- CDATA-szakaszok csak idegen tartalomban (MathML vagy SVG) használhatók.

HTML: az XML szintaxis

Példa:

```
<?xml version="1.0" encoding="UTF-8"?>
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en">
  <head>
    <title>Sample Page</title>
    <link rel="stylesheet" href="style.css"/>
  </head>
  <body>
    <p>Hello, World!</p>
  </body>
</html>
```

Nincs HTML séma

- DTD-k és XML sémák nem képesek kifejezni a HTML által támasztott valamennyi megfelelési követelményt.
- Lásd például az egyéni `data-*` attribútumokat.
 - Lásd: [Embedding custom non-visible data with the data-* attributes](#)

HTML dokumentumtípus-deklaráció

- A HTML szintaxisban a `<!DOCTYPE html>` dokumentumtípus-deklaráció szükséges, melynek célja mindössze annak biztosítása, hogy a dokumentum megjelenítése a szabványos módban történjen.
 - A fenti rövid dokumentumtípus-deklaráció előállítására nem képes szoftverek használhatják helyette a `<!DOCTYPE html SYSTEM "about:legacy-compat">` dokumentumtípus-deklarációt.
 - Lásd: [The DOCTYPE](#)
- Az XML szintaxisban tetszőleges dokumentumtípus-deklaráció használható, megadása nem is kötelező.
 - Az `application/xhtml+xml` média típussal továbbított dokumentumok megjelenítése mindig a szabványos módban történik.
 - Lásd: [Writing documents in the XML syntax](#)
- Lásd még: <https://www.w3.org/TR/html5-diff/#doctype>

DOM (1)

- Egy DOM fa egy dokumentum memóriabeli ábrázolása.
- Vonatkozó szabvány (WHATWG): [DOM Living Standard](#)

DOM (2)

- A DOM egy alkalmazásprogramozási interfész (API) dokumentumok (főleg HTML és XML dokumentumok) eléréséhez és manipulálásához.
- A DOM a **Dokumentum Objektum Modell** (***Document Object Model***) jelenti.
- Minden dokumentumot egy fa ábrázol, mely csomópontokból áll.
 - A csomópontok főbb típusai: Document, DocumentType, Element, Text, ProcessingInstruction, Comment

DOM (3)

Példa:

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <title>Sample Page</title>
  </head>
  <body>
    <p>Hello, World!</p>
    <!-- This is a comment -->
  </body>
</html>
```

```
{ DOCTYPE: html
  html lang="en"
  | head
  |   | #text: " "
  |   | title
  |   |   | #text: "Sample Page"
  |   | #text: " "
  | #text: " "
  | body
  |   | #text: " "
  |   | p
  |   |   | #text: "Hello, World!"
  |   | #text: " "
  |   | #comment: "This is a comment"
  |   | #text: " "
```

DOM (4)

- Minden egyes csomópontot egy API-val rendelkező objektum ábrázol, így tehát manipulálható.
- A DOM interfészek Web IDL-ben kerülnek leírásra.

DOM (5)

- A Web IDL egy interfészleíró nyelv, mely böngészőkben implementálható interfészek leírására szolgál.
- Az aktuális szabvány (WHATWG): [Web IDL Living Standard](#)
- Példa a használatra: [Node interfész](#)

DOM (6)

- A HTML specifikáció a HTML elemek ábrázolásához a DOM interfészeket kiterjesztő további interfészeket határoz meg.
- Példa:
 - meta elem
 - img elem

DOM (7)

- A DOM nem csupán egy API, a HTML implementációk megfelelési kritériumai is DOM műveletekkel vannak meghatározva.
- Példa: [The Navigator object](#)
- A specifikációk jórészt a DOM fák segítségével vannak megfogalmazva, nem pedig a jelölőkével.

DOM (8)

- Egy DOM fa szkriptekből manipulálható az oldalon.
- Példa:

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <title>DOM Example</title>
  </head>
  <body>
    <p>User agent: <span id="ua"></span></p>
    <script>
      document.getElementById('ua').innerHTML = navigator.userAgent;
    </script>
  </body>
</html>
```

DOM (9)

Eszközök:

- [Live DOM Viewer](#)

További információk:

- [MDN Web Docs – Document Object Model \(DOM\)](#)
- [The Modern JavaScript Tutorial – DOM tree](#)

A DOM használata (1)

```
// CSS tulajdonságok értékének beállítása:  
document.body.style.backgroundColor = 'aliceblue';  
document.body.style.margin = '2rem';  
document.getElementById('copyright').style.fontStyle = 'italic';  
  
// CSS tulajdonság számított értékének lekérése:  
const fontSize = window.getComputedStyle(document.body).fontSize;  
console.log(`Computed font size of body: ${fontSize}`);  
  
// Elem osztályainak manipulálása:  
console.log(document.body.className);  
document.getElementById('copyright').classList.add('important');
```

A DOM használata (2)

Elem elrejtése és megjelenítése:

```
<aside>
  <div id="help">
    <!-- ... -->
  </div>
  <button onclick="toggleHelp()">Hide/Show Help</button>
</aside>
<script>
  function toggleHelp() {
    const element = document.getElementById('help');
    element.hidden = !element.hidden;
  }
</script>
```

Az onevent attribútumok használata nem ajánlott!

A DOM használata (3)

Elem elrejtése és megjelenítése:

```
<aside>
  <div id="help">
    <!-- ... -->
  </div>
  <button id="toggle-help-btn">Hide/Show Help</button>
</aside>
<script>
  function toggleHelp(event) {
    const element = document.getElementById('help');
    element.hidden = !element.hidden;
  }
  document.getElementById('toggle-help-btn')
    .addEventListener('click', toggleHelp);
</script>
```

A DOM használata (4)

Elemek dinamikus létrehozása:

```
<div id="container"></div>
<button id="add-item-btn">Add Item</button>
<script>
  function addItem(event) {
    const container = document.getElementById('container');
    const item = document.createElement('div');
    item.className = 'item';
    container.appendChild(item);
  }
  document.getElementById('add-item-btn')
    .addEventListener('click', addItem);
</script>
```

A DOM használata (5)

Elemek dinamikus törlése:

```
<div id="container"></div>
<button id="remove-children-btn">Remove Children</button>
<script>
  function removeChildren(event) {
    const container = document.getElementById('container');
    while (container.firstChild) {
      container.removeChild(container.firstChild);
    }
  }
  document.getElementById('remove-children-btn')
    .addEventListener('click', removeChildren);
</script>
```

HTML: DOM, HTML és XML szintaxis

A DOM, a HTML szintaxis és az XML szintaxis közül nem mind képes ugyanazt a tartalmat ábrázolni.

	HTML	XML	DOM
Névterek	nem	igen	igen
noscript	igen	nem	nem
--> karakterlánc megjegyzésben	nem	nem	igen

HTML API-k

Lásd: [MDN Web Docs – Web APIs](#)

- [Console API](#)
- [Fetch API](#)
- [Web Storage](#)
- ...

Böngészők megjelenítési módjai (1)

A renderelő motorok az alábbi módokban jelenítenek meg HTML dokumentumokat:

- Kompatibilitási mód (*quirks mode*)
- Szabványos mód (*standards mode*)
- Majdnem szabványos mód (*almost standards mode*)

Böngészők megjelenítési módjai (2)

- A megjelenítési módok létezésének történeti oka, hogy a korai webszabványok nem voltak kompatibilisek az akkoriban létező böngészők viselkedésével.
- A böngészők egy új megjelenítési módot vezettek be, hogy megfeleljenek a webszabványoknak, miközben továbbra is képesek a létező tartalmak helyes megjelenítésére.
- Így eltérő módon történik az aktuális szabványoknak megfelelő modern és a régmódi weboldalak megjelenítése.

Böngészők megjelenítési módjai (3)

- **Kompatibilitási mód (*quirks mode*):** régi böngészők viselkedésének utánzása (emulálása) az aktuális webszabványokat sértő módon régi weboldalak megjelenítéséhez.
 - *Quirk*: jelentése “furcsa viselkedés”.
- **Szabványos mód (*standards mode*):** a weboldalak megjelenítése az aktuális webszabványoknak megfelelően.
- **Majdnem szabványos mód (*almost standards mode*):** néhány böngészőmotor rendelkezik egy harmadik üzemmóddal is, mely bizonyos magasságok meghatározásában tér el a szabványos módtól, mely például képek táblázatcellákban való elhelyezését érinti.

Böngészők megjelenítési módjai (4)

- Korábban a böngészőmotorok némileg eltérő kompatibilitási módokat használtak, azonban a WHATWG a [Quirks Mode](#) specifikációban szabványosította a kompatibilitási módot.
- A specifikáció nem sorol fel a szabványos viselkedéstől való minden eltérést, melyek jelenleg a böngészőkben léteznek, számos ilyen más WHATWG specifikációk részleteznek.

Böngészők megjelenítési módjai (5)

- A WHATWG **DOM** specifikációja átnevezi a szabványos és a majdnem szabványos módot:
 - *Standards mode* helyett *no-quirks mode*
 - *Almost standards mode* helyett *limited-quirks mode*

Böngészők megjelenítési módjai (6)

Megjelenítési mód választása egy HTML dokumentumhoz:

- A `text/html` média típussal továbbított dokumentumokhoz a dokumentumtípus-deklaráció határozza meg a megjelenítési módot.
- Az `application/xhtml+xml` média típussal továbbított dokumentumok mindig a szabványos módban kerülnek megjelenítésre.
 - Ilyenkor a dokumentum elemzése egy olyan XML elemzővel történik, mely a dokumentum jólformáltságát is ellenőrzi.

Böngészők megjelenítési módjai (7)

A használt megjelenítési mód meghatározása:

- A Document DOM interfész `compatMode` attribútuma a "BackCompat" karakterláncot adja vissza, ha a dokumentum megjelenítési módja a kompatibilitási mód, egyébként pedig a "CSS1Compat" karakterláncot.
 - Lásd: [DOM – Living Standard – Interface Document](#)
- A böngésző felhasználói felületén:
 - Firefox: [Oldal adatai](#) (az Eszközök menüből elérhető)

Böngészők megjelenítési módjai (8)

További hasznos címek:

- [MDN Web Docs – Quirks Mode and Standards Mode](#)
- [Henri Sivonen. Activating Browser Modes with Doctype.](#)

Hibakezelés (1)

- A HTML korábbi verzióitól eltérően az aktuális specifikáció az érvényes dokumentumok feldolgozása mellett bizonyos szintű részletességgel az is előírja, hogy hogyan kell feldolgozni az érvénytelen dokumentumokat.
- A webböngészők valójában nagyon elnézőek a hibákkal szemben, automatikusan kijavítják a nem érvényes tartalmat.
 - Azonban ez elsősorban a HTML szintaxisú dokumentumokra vonatkozik!

Hibakezelés (2)

Példa:

```
<html>
  <foo>
    <p bar class=>Does it <b>work,<i> or</b> not</i>?
  <
  </body>
</html>
```

Vizsgáljuk meg a dokumentumból felépülő DOM fát a böngészőben a webfejlesztő eszközökkel vagy a [Live DOM Viewer](#) révén!

Hibakezelés (3)

Megfelelési ellenőrző vagy más néven érvényesítő (*conformance checker*, *validator*):

- Egy olyan szoftvereszköz, mely ellenőrzi, hogy egy HTML dokumentum teljesíti-e a HTML specifikáció megfelelési követelményeit.
- Példa:
 - Nu Html Checker (platform: Java; licenc: *MIT License*)
<https://validator.github.io/validator/>
<https://github.com/validator/validator>
 - Webes interfész: <https://validator.w3.org/nu/>

Hibakezelés (4)

Lásd:

- [HTML Living Standard – Parsing HTML documents](#)
 - [An introduction to error handling and strange cases in the parser](#)
- Paul Irish, Tali Garsiel. [How browsers work](#).
 - [Browsers' error tolerance](#)

Böngésző támogatás

- A modern böngészőmotorok támogatják.
- Lásd:
 - *HTML5test – How well does your browser support HTML5?* (inaktív)
<https://html5test.com/>
<https://github.com/WebPlatformTest/HTML5test>
 - Nem hivatalos frissített verzió: <https://html5test.co/>
<https://github.com/niutech/html5test>
 - [Can I use... Support tables for HTML5, CSS3, etc](#)

Böngésző fejlesztőeszközök

- Chromium, Google Chrome, Opera: [Chrome DevTools](#)
- Firefox: [Firefox DevTools User Docs](#)
- Safari: [Safari Developer Tools](#)
- Chromium-alapú Edge: [Microsoft Edge DevTools documentation](#)

Böngészők fejlesztői kiadásai

- Google Chrome: [Google Chrome for developers](#)
- Firefox: [Firefox Browser Developer Edition](#)

HTML szerkesztők (1)

Szabad és nyílt forrású szoftverek:

- Visual Studio Code (platform: Linux, macOS, Windows; licenc: *MIT License*) <https://code.visualstudio.com/>
<https://github.com/Microsoft/vscode>
 - Lásd: <https://code.visualstudio.com/docs/languages/html>
 - Ajánlott kiterjesztések:
 - Live Preview
<https://marketplace.visualstudio.com/items?itemName=ms-vscode.live-server> <https://github.com/microsoft/vscode-livepreview>

HTML szerkesztők (2)

Emmet (programozási nyelv: JavaScript; licenc: *MIT License*)

<https://emmet.io/> <https://github.com/emmetio/emmet>

- Szövegszerkesztő bővítmények HTML és CSS kód írásának gyorsításához.
- Számos szövegszerkesztőhöz rendelkezésre áll, mint például: Eclipse, NetBeans, Notepad++, Visual Studio Code, IntelliJ IDEA, ...
 - Lásd: <https://emmet.io/download/>
- Dokumentáció: <https://docs.emmet.io/>
 - Testreszabás: <https://docs.emmet.io/customization/>
- Lásd még: [Emmet in Visual Studio Code](#)

HTML szerkesztők (3)

Példák Emmet rövidítésekre:

- `ul>li*3`

kifejtése:

```
<ul>
  <li></li>
  <li></li>
  <li></li>
</ul>
```

- `section.chapter>h1{Introduction}+p`

kifejtése:

```
<section class="chapter">
  <h1>Introduction</h1>
  <p></p>
</section>
```

HTML szerkesztők (4)

Példák Emmet rövidítésekre (folytatás):

```
ul>li*5>a[href=#chapter$]{Chapter $}
```

kifejtése:

```
<ul>
  <li><a href="#chapter1">Chapter 1</a></li>
  <li><a href="#chapter2">Chapter 2</a></li>
  <li><a href="#chapter3">Chapter 3</a></li>
  <li><a href="#chapter4">Chapter 4</a></li>
  <li><a href="#chapter5">Chapter 5</a></li>
</ul>
```

HTML: Egyéb szoftverek (1)

Szabad és nyílt forrású szoftverek:

- HTML5 Boilerplate (platform: böngésző; licenc: *MIT License*)
<https://html5boilerplate.com/>
<https://github.com/h5bp/html5-boilerplate>
- http-server (platform: Node.js; licenc: *MIT License*)
<https://www.npmjs.com/package/http-server>
<https://github.com/http-party/http-server>
- Tidy (platform: Linux, macOS, Windows; licenc: *Tidy License*)
<https://www.html-tidy.org/> <https://github.com/htacg/tidy-html5>

HTML: Egyéb szoftverek (2)

http-server: a simple static HTTP server

- Használat:

```
$ npx http-server -o index.html
```

```
$ npm install -g http-server
```

```
$ http-server -o index.html
```

- A dokumentum itt elérhető: <http://localhost:8080/index.html>.

HTML: online szolgáltatások

Surge: ingyenes statikus HTML publikálás a parancssorból.

- Webhely: <https://surge.sh/>
- Dokumentáció: <https://surge.sh/help/>
- Telepítés és használat:

```
$ npm install --global surge  
$ surge --domain example.surge.sh
```

- A webhely itt elérhető: <https://example.surge.sh/>.

SVG (1)

- Nyelv két-dimenziós vektorgrafika XML-ben történő leírásához.
- Az interaktív grafikát és animációt is támogatja.
- Fejlesztője a [W3C SVG Working Group](#).

SVG (2)

- Az aktuális szabvány: [Scalable Vector Graphics \(SVG\) 1.1 \(Second Edition\)](#) (W3C ajánlás, 2011. augusztus 16.)
- A szabvány következő verziója: [Scalable Vector Graphics \(SVG\) 2](#) (előzetes W3C javaslattev, 2018. október 4.)
- Az SVG egy profilja mobil eszközökre: [Scalable Vector Graphics \(SVG\) Tiny 1.2 Specification](#) (W3C ajánlás, 2008. december 22.)

SVG (3)

- SVG tartalom beágyazható más dokumentumokba.
 - A HTML támogatja a közvetlen beágyazást (lásd az `svg` elemet).
- Séma (DTD):
<https://www.w3.org/Graphics/SVG/1.1/DTD/svg11.dtd>
- Fájlkiterjesztés: `.svg`
- Média típus: `image/svg+xml`

SVG (4)

Böngésző támogatás:

- Az összes modern böngésző natív módon támogatja.
- Lásd: <https://caniuse.com/svg>

SVG (5)

Szabad és nyílt forrású szoftverek: szerkesztők

- Inkscape (platform: Linux, macOS, Windows; licenc: GPLv2)
<https://inkscape.org/> <https://gitlab.com/inkscape/inkscape>
- macSVG (platform: macOS; licenc: *MIT License*)
<https://macsvg.org/> <https://github.com/dsward2/macSVG>
- SVG-Edit (programozási nyelv: JavaScript; licenc: *MIT License*)
<https://github.com/SVG-Edit/svgedit>
- tldraw (programozási nyelv: TypeScript; licenc: *Apache License 2.0*)
<https://www.tldraw.com/> <https://github.com/tldraw/tldraw>

SVG (6)

Szabad és nyílt forrású szoftverek: könyvtárak

- Apache Batik (programozási nyelv: Java; licenc: *Apache License 2.0*)
<https://xmlgraphics.apache.org/batik/>
- Frappe Charts (programozási nyelv: JavaScript; licenc: *MIT License*)
<https://frappe.io/charts> <https://github.com/frappe/charts>
- Rough.js (programozási nyelv: TypeScript; licenc: *MIT License*)
<https://roughjs.com/> <https://github.com/rough-stuff/rough>
- Snap.svg (programozási nyelv: JavaScript; licenc: *Apache License 2.0*) <http://snapsvg.io/>
<https://github.com/adobe-webplatform/Snap.svg/>
- SVG.js (programozási nyelv: JavaScript; licenc: *MIT License*)
<https://svgjs.dev/> <https://github.com/svgdotjs/svg.js>

SVG (7)

Mintapéldák:

- Bootstrap Icons (licenc: *MIT License*)
<https://icons.getbootstrap.com/> <https://github.com/twbs/icons/>
- Feather – Simply beautiful open source icons (licenc: *MIT License*)
<https://feathericons.com/> <https://github.com/feathericons/feather>
- Inkscape Gallery <https://inkscape.org/gallery/>
- Super Tiny Icons (licenc: *MIT License*)
<https://github.com/edent/SuperTinyIcons>
- Tabler Icons (licenc: *MIT License*) <https://tabler.io/icons>
<https://github.com/tabler/tabler-icons>

SVG (8)

További hasznos címek:

- [MDN Web Docs – SVG](#)
 - [SVG Tutorial](#)

MathML (1)

- Nyelv matematikai jelölések XML-ben történő leírásához.
- Célja lehetővé tenni a weben matematika “felszolgálását”, fogadását és feldolgozását, miként a HTML lehetővé tette ezt a funkcionalitást szöveghez.
- Számos célra felhasználható: például matematikai tartalom megjelenítése a Weben, matematikai formulák exportálása komputer algebra rendszerekből, ...
- Fejlesztője a [W3C Math Working Group](#)

MathML (2)

- Az aktuális szabvány: **Mathematical Markup Language (MathML) Version 3.0 2nd Edition** (W3C ajánlás, 2014. április 10.)
 - ISO szabványként is ratifikálták: **ISO/IEC 40314:2016: Information technology – Mathematical Markup Language (MathML) Version 3.0 2nd Edition**
- A szabvány következő, jelenleg fejlesztés alatt álló verziója a MathML 4: <https://www.w3.org/TR/mathml4/>

MathML (3)

- Leírható segítségével matematikai kifejezések megjelenítése és jelentése is.
 - Külön elemkészletek szolgálnak a két eltérő célra (prezentációs elemek, tartalmi elemek).
- MathML tartalom beágyazható más dokumentumokba.
 - A HTML támogatja a közvetlen beágyazást (lásd a `math` elemet).

MathML (4)

- Séma (RELAX NG):
 - <https://www.w3.org/Math/RelaxNG/mathml3/mathml3.rng>
 - <http://www.w3.org/Math/RelaxNG/mathml3/mathml3.rnc>
- Fájlkiterjesztés: `.mml`
- Média típus: `application/mathml+xml`,
`application/mathml-presentation+xml`,
`application/mathml-content+xml`

MathML (5)

Böngésző támogatás:

- **Blink** (Chromium, Google Chrome, Opera, Chromium-alpú Edge): támogatás a 2023 januárjában megjelent 109-es Chromium verzió óta van
 - Lásd: <https://www.igalia.com/2023/01/10/Igalia-Brings-MathML-Back-to-Chromium.html>
- **Gecko** (Firefox): támogatott
- **WebKit** (Safari): támogatott

Lásd: [Can I use... Support tables for HTML5, CSS3, etc](#)

MathML (6)

- Sok további alkalmazás támogatja, például a Maple, Wolfram Mathematica, LibreOffice (MathML importálás), ...
- Ezek egy gyűjteményét lásd itt:
https://www.w3.org/wiki/Math_Tools

MathML (7)

Szabad és nyílt forrású szoftverek:

- MathJax (programozási nyelv: JavaScript; licenc: *Apache License 2.0*)
<https://www.mathjax.org/> <https://github.com/mathjax/MathJax>
 - JavaScript-ben írt megjelenítő motor.

MathML (8)

Mintapéldák:

- <https://developer.mozilla.org/en-US/docs/Web/MathML/Examples>
- W3C MathML Test Suite

MathML (9)

További hasznos címek:

- [MDN Web Docs – MathML](#)
- [Planet MathML \(W3C\)](#)
- [MathML in Web Browsers](#)