

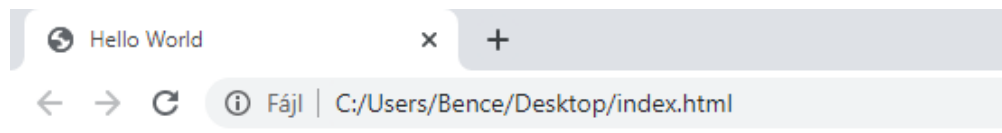
Szkriptnyelvek - 7. gyakorlat

Cservenák Bence

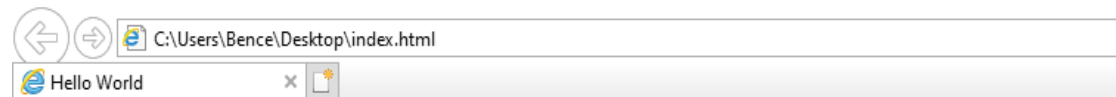


HTML5 és CSS recap

- **HTML:** weboldalak és webes alkalmazások készítésére szolgáló leíró nyelv
- **CSS:** weboldalak tartalmának formázására szolgáló stílusleíró nyelv
- A kód értelmezéséhez és megjelenítéséhez **böngészőt** használunk
 - ez viszont böngésző- és böngészőverziófüggő lehet!



Árvíztűrő tükörfúrógép



ÄrvÄztL±rL‘ tÄLkÄ¶rfÄsrÄlgÄ©p



- HTML alapfogalmak

- **tagek:** < és > közé írt, speciális jelentésű HTML objektumok

- ▶ **páros tagek:** nyitótag + zárótag

```
<p>Ide jön valami szöveg...</p>
```

- ▶ **páratlan (=üres) tagek:** csak nyitótag

```
Első sor <br/>Második sor
```

- ▶ **tagek egymásba ágyazása:**

```
<div><p><span>Hello World!</span></p></div>
```

- **attribútumok:** tagekhez tartozó speciális tulajdonságok

- ▶ speciális értékek tartoznak hozzájuk, a nyitótag neve után írjuk őket

```
<a href="https://www.google.com" class="link">Google</a>
```

- Az alap HTML dokumentum

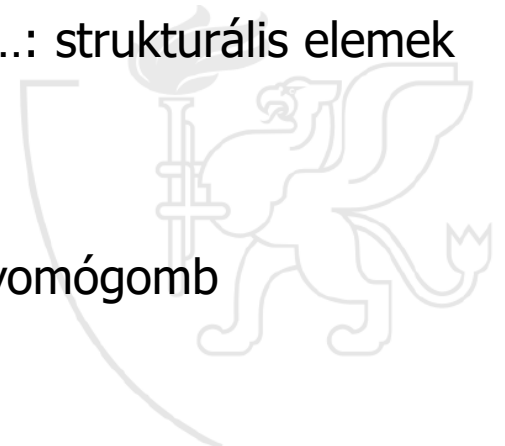
```
<!DOCTYPE html>
<html>
<head>
  <title>Böngészőfülön megjelenő cím...</title>
  <meta charset="UTF-8"/>
</head>
<body>

</body>
</html>
```

- `<head>...</head>`: metainformációk
- `<body>...</body>`: a megjelenítendő tartalom

- Néhány fontosabb HTML elem

- `<!-- ez egy komment -->`: megjegyzés
- `<h1>...</h1>`, `<h2>...</h2>`, ..., `<h6>...</h6>`: címsorok
- `<p>...</p>`: bekezdés
- `
`: sortörés
- `Google`: hivatkozás (link)
- ``: kép
- `<div>...</div>`, `...`, `<section>...</section>`, ...: strukturális elemek
- `...`, `...`: listák
- `<table>...</table>`: táblázat
- `<button type="button">Click Me!</button>`: egyszerű nyomógomb
- `id` attribútum: egyedi azonosító → egy adott elem azonosítása
- `class` attribútum: osztály → általában több elem csoportosítására használjuk



- CSS beágyazása

- **inline CSS:** tagek `style` attribútumaként megadva

```
<p style="color: blue;">I'm blue da ba dee da ba die...</p>
```

- **internal CSS:** a `<head>`-ben `<style>` és `</style>` között megadva

```
<head>
  <style>
    p { color: blue; }
  </style>
</head>
```

- **external CSS:** a `<head>`-ben, `<link/>` segítségével hivatkozva (külső stíluslap)

```
<head>
  <link rel="stylesheet" href="main.css"/>
</head>
```

- CSS kommentek: `/* ez egy komment CSS-ben */`
- Utasítások szintaxisa: `selector { property1: value1; property2: value2; ... }`
 - `selector`: a formázandó objektum
 - `property`: a formázandó tulajdonság
 - `value`: az adott tulajdonsághoz tartozó, beállítani kívánt érték

```
body {  
    margin: 0;  
    padding: 0;  
    background-color: #f0f0f0;  
    font-family: Arial, sans-serif;  
}  
  
#main-content .centered {  
    text-align: center;  
}
```



- Alapvető CSS szelektorok

- `div`: kijelöli az összes `<div>` elemet (element selector)
- `.foo`: kijelöli az összes `class="foo"` attribútummal rendelkező elemet (class selector)
- `#foo`: kijelöli az `id="foo"` attribútummal rendelkező elemet (id selector)
- `div, p`: kijelöli az összes `<div>` és `<p>` elemet (group selector)
- `div p`: kijelöli az összes olyan `<p>` elemet, amely egy `<div>`-en belül van
- `div>p`: kijelöli az összes olyan `<p>` elemet, amelynek közvetlen szülője egy `<div>`
- ...



- A JavaScript egy platformfüggetlen szkriptnyelv
- Eredeti célja: weboldalak interaktívvá tétele
- Kliens- és szerveroldalon egyaránt használható
 - kliensoldalon a böngészőben lévő JavaScript Engine felel a futtatásért
 - szerveroldalon futtatva nincs szükségünk böngészőre (pl. Node.js)
- Teljes integráció a HTML-lel és CSS-sel
 - egy HTML dokumentumba ágyazhatunk be JavaScript nyelven írt kódokat



- JavaScript kódot a `<script>` taggel ágyazhatunk be HTML-be
 - tetszőleges mennyiségű `<script>` tag lehet egy HTML fájlban
 - általában a `<head>`-be, illetve a `<body>` végére írjuk ezeket
 - a tag `src` attribútumával külső JavaScript fájl is beágyazható

```
<!DOCTYPE html>
<html>
  <head>
    <title>Hello JavaScript</title>
    <meta charset="UTF-8"/>
    <script src="main.js"></script>
  </head>
  <body>
    <script>
      // ide jön a JavaScript kód...
    </script>
  </body>
</html>
```



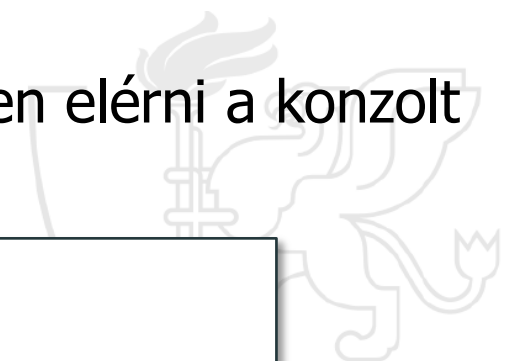
- JavaScriptben is lehetőségünk van **kommentek** (megjegyzések) írására

```
// egysoros komment  
  
/*  
    több soros  
    komment  
*/
```



- `alert()` : felugró ablak
- `document.write()` : kiíratás a HTML dokumentumba
 - ha azután használjuk, miután a HTML dokumentum betöltődött, minden meglévő HTML tartalmat töröl
- `console.log()` : kiíratás a konzolra
 - mi ezt fogjuk használni :)
 - a böngészőknél általában az F12-vel tudjuk egyszerűen elérni a konzolt

```
alert("Hello Alert Box!");  
document.write("Hello HTML Document!");  
console.log("Hello Console!");
```



- `boolean`: logikai adattípus
 - `true` vagy `false` értéket vehet fel
- `number`: szám adattípus
 - egész vagy lebegőpontos szám
- `string`: szöveges adattípus
 - megadás: aposztrófok (`' ... '`) vagy idézőjelek (`" ... "`) között
 - összefűzés: `+` operátorral
- `undefined`: definiálatlan érték
 - pl. amikor egy változónak még nem adtunk értéket



- JavaScriptben többféle módon hozhatunk létre változókat

```
var a = 10;           // ezt ne használjuk!  
let b = 20;           // használjuk helyette inkább ezt!  
  
console.log("a értéke: " + a + ", b értéke: " + b);  
  
const KEDVENC_TARGY = "Szkriptnyelvek"    // konstans (állandó)  
// KEDVENC_TARGY = "Prog2"                // hiba!  
  
console.log("Kedvenc tárgyam: " + KEDVENC_TARGY);
```

```
a értéke: 10, b értéke: 20  
Kedvenc tárgyam: Szkriptnyelvek
```

- A JavaScript (a Pythonhoz hasonlóan) **dinamikusan típusos nyelv**

```
let val;                                // val típusa: undefined
console.log(val);

val = 42;                                // val típusa: number
console.log(val);

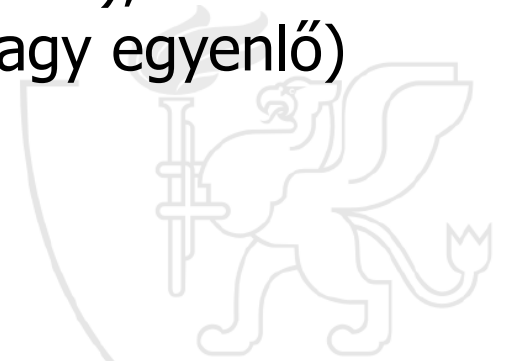
val = "Béla";                            // val típusa: string
console.log(val);
```

```
undefined
42
Béla
```

- **Aritmetikai operátorok:** + (összeadás), - (kivonás), * (szorzás), / (valós osztás), % (maradékos osztás), ** (hatványozás)
- **Hozzárendelő operátorok:** =, +=, -=, *=, /= (ugyanúgy, mint C-ben)
- **Inkrementáló, dekrementáló operátorok:** ++, -- (ugyanúgy, mint C-ben)
- **Összehasonlító operátorok:** == (egyenlő), === (azonos érték és típus), != (nem egyenlő), !== (eltérő érték vagy típus), < (kisebb mint), <= (kisebb vagy egyenlő), > (nagyobb mint), >= (nagyobb vagy egyenlő)

```
42 == "42";           // true
42 === "42";          // false
```

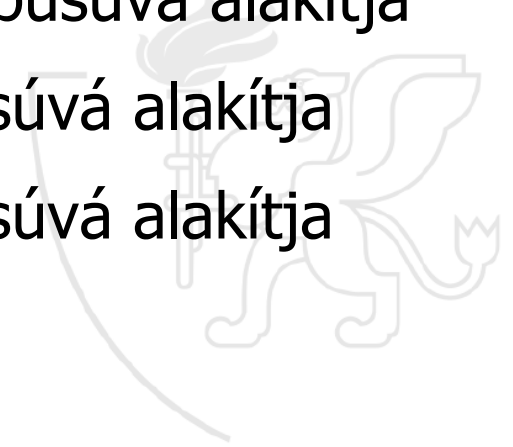
- **Logikai operátorok:** && (logikai ÉS), || (logikai VAGY), ! (logikai NEM)



- JavaScriptben a típusok közötti konverzió automatikus

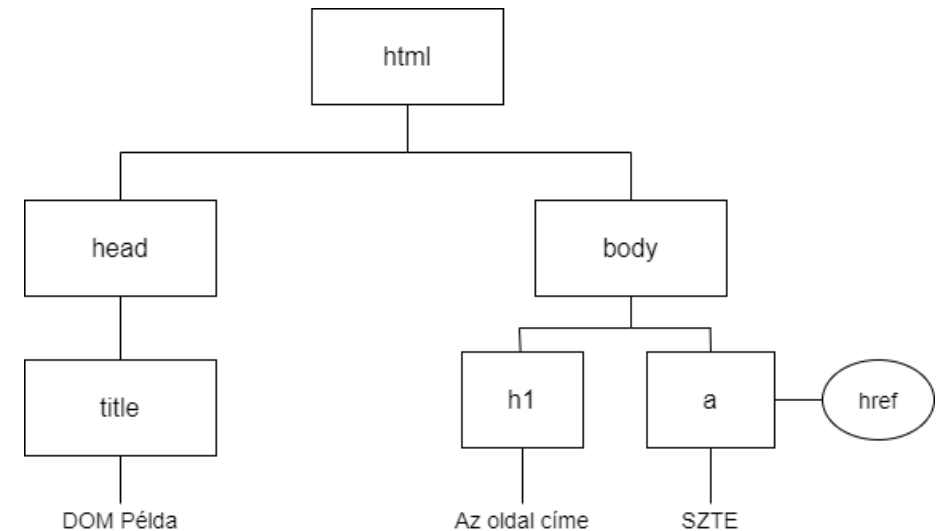
```
"2" + 5;           // "2" + "5" = "25"  
"2" * 5;           // 2 * 5 = 10  
"sajt" * 5;        // NaN (Not a Number)
```

- Itt is létezik a más nyelvekből ismerős **típuskényszerítés (typecasting)**
 - Boolean**(expr) : az expr kifejezést boolean típusúvá alakítja
 - Number**(expr) : az expr kifejezést number típusúvá alakítja
 - String**(expr) : az expr kifejezést string típusúvá alakítja



- A HTML tagek egymásba ágyazhatók → hierarchikus faszerkezet
- **HTML DOM** (Document Object Model): HTML objektumok hierarchiája
 - DOM fa (DOM Tree): vizuálisan szemlélteti az objektumok hierarchikus viszonyait
 - JavaScriptben különböző DOM műveleteket végezhetünk

```
<html>
  <head>
    <title>DOM Példa</title>
  </head>
  <body>
    <h1>Az oldal címe</h1>
    <a href="https://u-szeged.hu/">SZTE</a>
  </body>
</html>
```



- Fontosabb DOM szelektorok
 - `document.getElementById()`: adott `id`-vel rendelkező elem lekérése
 - `document.getElementsByTagName()`: adott tagnévvel rendelkező elemek lekérése
 - `document.getElementsByClassName()`: adott `class` értékkel rendelkező elemek lekérése
- `innerHTML`: egy objektum HTML tartalma
 - HTML markup használható
- `innerText`: egy objektum szöveges tartalma
 - HTML markup nem használható (egyszerű szöveggént lesz megjelenítve)



- Példa

```
<p id="text">A DOM műveletekkel jókat lehet szórakozni.</p>
<button type="button" onclick="document.getElementById('text').innerText = 'sajt'">
  Kattints ide!
</button>

<script>
  const myText = document.getElementById("text");
  myText.innerHTML = "<b>Haha, itt nem ez volt eredetileg!</b>";
</script>
```