attrecto
NEXT TECH DIGITAL SOLUTIONS

HTML, CSS, BOOTSTRAP

ISO 9001 Certified

FT FINANCIAL TIMES 1000 Europe's Fastest Growing Companies

50™ Technology Fast 50 2016 CENTRAL EUROPE

500 Technology Fast 500 2016 EMEA

HTML

- HTML - HyperText Markup Language
- Tim Berners-Lee hozta létre, ezzel leírható egy weboldal
- Kifejezetten az internetre készült speciális leíró nyelv
- SGML általános leíró nyelv alapján készült
- weboldal felépítését, tartalmát adhatjuk meg
- alkalmas formázásra is, de korlátozottan (helyette: CSS)

HTML felépítése

- <!DOCTYPE html> Dokumentum típus definíció, leírja a HTML verzióját (böngésző tudja milyen verziót kell megjelenítenie, HTML5)
- <html> gyökér elem
- <head> meta információkat tartalmaz a dokumentumról
  - <title> dokumentum címe
  - <style>
  - <link>
  - ...
- <body> az oldalon látható tartalom van benne

```
<!DOCTYPE html>
<html>
 <head>
   <title>Oldal címe</title>
 </head>
 <body>

   <h1>fejléc</h1>
   <p>bekezdés</p>
 </body>
</html>
```

HTML elementek és attribútumok

- Elementek
  - kezdő tag
  - element tartalmi része
  - záró tag
- Üres elementnek nincs záró tagje
- Attribútumok
  - információt tartalmaznak az elementről

```
<tagname> tartalom </tagname>

<p> Bekezdés </p>

<br>

<a href=""></a>
```

Linkek

- Külső link
- Belső link
- Könyvjelző

```
<h2>Linkek</h2>
<div>
  <label>külső</label>
  <a class="formula" href="https://formula.hu"
target="_blank">This is a link to Formula.hu</a>
</div>
<div>
  <label>belső (weboldalon belüli)</label>
  <a href="other-template.html">To Other HTML</a>
</div>
<div>
  <label>könyvjelző</label>
  <a href="#table">To Table</a>
</div>
```

Képek, ikonok

- Képek
  - <img>
    - src attribútumának lehet beállítani
    - alt (kép nem tölthető be)
    - title (hover)
    - ha fontos része a tartalomnak
  -
    - háttérnek lehet beállítani
    - ha a kép nem a tartalom része
    - css manupiláció
  - <picture>
    - responsive design (kép skálázás helyett több kép)
- Ikonok
  - <i>, , külső library használatával. pl: font-awsome, css manupiláció (font-size,...)

```
<img src="image.jpg" alt="W3Schools.com"
width="104" height="142">
```

```
<div style="
    background-image: url(image.jpg);
    background-position: center;
    width: 100px;
    height: 100px;
    background-size: cover;
    display: inline-block;">
</div>
```

```
<head>
  <link rel="stylesheet"
href="https://cdnjs.cloudflare.com/ajax/libs/font
-awesome/4.7.0/css/font-awesome.min.css">
</head>

<body>
  <i class="fa fa-cloud"></i>
</body>
```

Tábla

- <table>
- <thead> - fejrész tartalma
- <tbody> - törzsrész tartalma
- <tfoot> - lábrész tartalma
- <tr> - sor
- <th> - fejléc cella
- <td> - törzs cella
- css
- fő előnye
  - cellák magassága, szélessége soronként és oszloponként együtt változnak
- ne használjuk layout-hoz
  - szemantikailag helytelen
  - kevésbé rugalmas mint pl a div

```
<table style="width:100%">
  <thead>
   <tr>
     <th>Firstname</th>
     <th>Lastname</th>
     <th>Age</th>
   </tr>
  </thead>
  <tbody>
   <tr>
     <td>Jill</td>
     <td>Smith</td>
     <td>50</td>
   </tr>
   <tr>
     <td>Eve</td>
     <td>Jackson</td>
     <td>94</td>
   </tr>
  </tbody>
</table>
```

JavaScript

- A <body> végére érdemes helyezni a scripteket
  - javítja a megjelenési sebességet, mert a script összeállítása lelassítja a megjelenítést
- Használata
  - külső fájlba (ajánlott)
  - <script> tagek közé

```
<body>
   <h1>This is a Other Template</h1>
   <a href="HTML-template.html">Back</a>
   <input type="button"
onclick="setBackgroundColor('red');"
value="setBgColor">
   <!--
      <script>
            function setBackgroundColor(color){

document.getElementsByTagName('body')[0].style.ba
ckgroundColor = color;
            }
      </script>
      -->
   <script src="sample.js"></script>
</body>
```

Form

- <form>
  - action (eseménykezelőt vár ami a submitot kezeli BE)
  - onsubmit (FE)
  - <input>
    - típusai
      - text
      - radio
      - submit
      - ...
    - attribútumai
      - value
      - disabled
      - readonly
  - <fieldset> csoportok a formon belül

```
<form onsubmit="submitForm()">
 <fieldset>
     <legend>Radio type</legend>
     <input type="radio" name="gender"
value="male" checked> Male<br>
     <input type="radio" name="gender"
value="female"> Female<br>
     <input type="radio" name="gender"
value="other"> Other
  </fieldset>

  <fieldset>
     <legend>Checkbox type</legend>
     <input type="checkbox" name="vehicle1"
value="Bike"> I have a bike<br>
     <input type="checkbox" name="vehicle2"
value="Car"> I have a car
  </fieldset>

  <label>Button type</label>
  <input type="button" value="inputButton">

  <input type="submit" value="Submit">
</form>
```

Form

- <select> dropdown lista
  - <option>
- <textarea>
  - rows (sorok száma)
  - cols (oszlopok száma)

```
  <label>Text type</label>
  <input type="text">
  <input type="text" value="Readonly" readonly>

  <label>Password type</label>
  <input type="password">

  <fieldset>
     <legend>Select</legend>
     <select name="cars">
       <option value="volvo">Volvo</option>
       <option value="saab">Saab</option>
       <option value="fiat">Fiat</option>
       <option value="audi">Audi</option>
     </select>
  </fieldset>


  <fieldset>
     <legend>TextArea</legend>
     <textarea name="message" rows="10"
cols="30">
        The cat was playing in the garden.
     </textarea>
  </fieldset>
```

CSS

- Weboldalak formázására szolgál
- Sokkal több, komolyabb, látványosabb formázásra ad lehetőséget, mint a HTML
  - animációk
  - lenyílómenü
  - ...
- Könnyebb módosítani egy adott formázást
  - Például ha van a weblapunkon egy vörös színnel írt szöveg 100 helyen, amit a HTML-lel állítottunk be, hogy vörös legyen és ezt módosítani szeretnénk, akkor 100 helyen át kell írnunk. Ezzel ellentétben a CSS lehetőséget ad arra, hogy csak egyetlen egy helyen kelljen módosítani és azzal minden helyen módosul egyszerre.

Szintaxis

- h1 - szelektor
- color, font-size - tulajdonság
- blue, 12px - érték
- Szelektorok
  - tag - h1, p, ...
  - id - #id
  - class - .class
  - custom - p.center (p tag amin van center class is)
  - group - több tagnek is ugyanaz a stílus definíció
  - ...

```
h1 {
   color: blue;
   font-size: 12px;
}

#id {

}

.class {

}

p.center {

}

h1, h2, p {
   text-align: center;
   color: ;
}
```

CSS hozzáadása HTML element-hez

- Inline - style attribútumot használva
- Internal - HTML <head> szekción belül
- External - másik fájlban (ajánlott)

Inline stílus a legmagasabb prioritású ezek közül, csak !important-tal lehet felülírni.

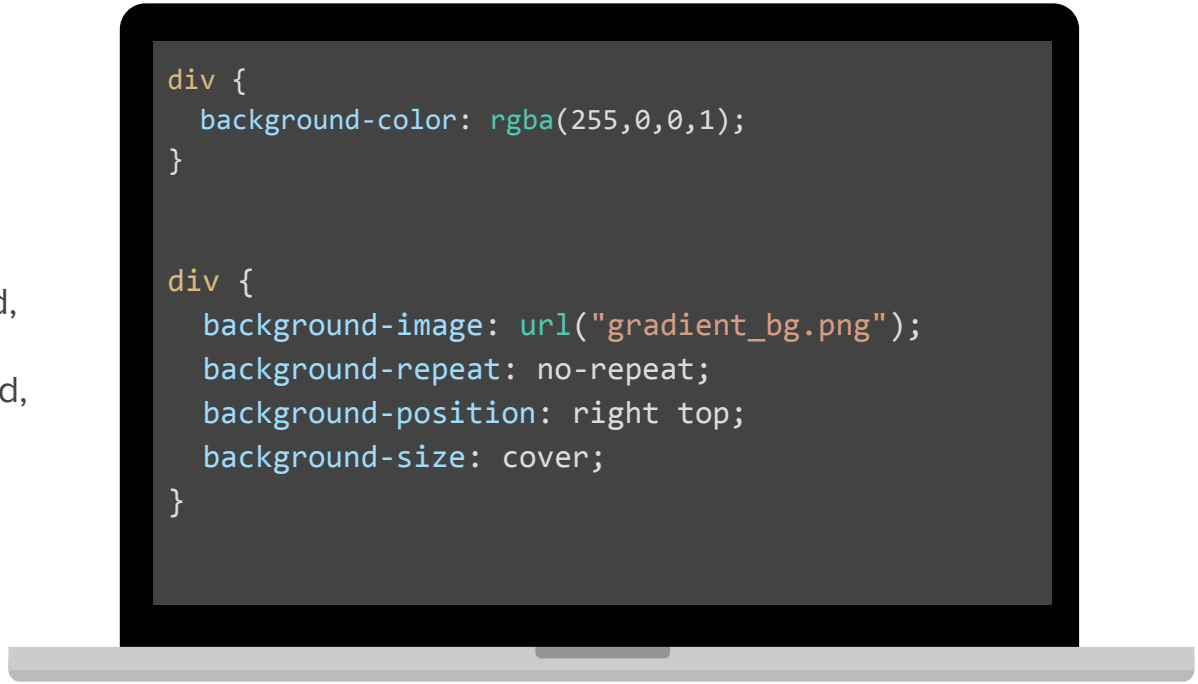
```html
<h1 style="color:blue;">
  This is a Blue Heading
</h1>
```

```html
<style>
  body {background-color: powderblue;}
  h1   {color: blue;}
  p    {color: red;}
</style>
```

```html
<link rel="stylesheet" href="styles.css">
```

Színek, Hátterek

- background-color
- color
- border-color
- color-values
  - rgb, rgba
    - rgb(255,0,0) (red)
    - rgba(255,0,0,1) (red, nem átlátszó)
    - rgba(255,0,0,0) (red, átlátszó)
  - hex
    - #ff0000 (red)
- background-image
- background-repeat
- background-position
- background-size



Box model

- margin
- padding
- border
- box-sizing: border-box
    - ezt használva a tartalom (content) szélesség a padding méretével kevesebb lesz

Display vs Visibility, Position, Overflow

- Display vs visibility
  - display: none a helyet sem foglalja
  - visibility: hidden a helyet foglalja
  - visibility: collapse nem foglaja a helyet, de csak table esetén működik
- Position
  - static (alapértelmezett)
  - relative (az eredeti helyén marad)
  - fixed (mindig ugyanott lesz a viewporton)
  - absolute (relativehoz képest, alapértelemezetten a html taghez)
  - sticky (megtartja a helyét, görgetés után lesz fixed)
- Overflow
  - visible, hidden
  - hosszú szöveg kipontozása

```
display: block;
display: none;
visibility: hidden;
visibility: visible;


P {
  display: block;
  overflow: hidden;
  text-overflow: ellipsis;
  white-space: nowrap;
}
```

Combinators

- descendant selector (space)
- child selector (>)
- adjacent sibling selector (+) (szomszédos ami utána van, csak az első)
- general sibling selector (~) (szomszédos ami utána van, mind)

```
.combinators div p { /*child of div*/
   background-color: yellow;
}

.combinators div > p { /*child of div (not
grandchild)*/
   background-color: red;
}

.combinators div + p { /*next element that is p
(just first)*/
   background-color: green;
}

.combinators div ~ p { /*next element that is p
(all)*/
   background-color: blue;
}
```

Pseudo class-ok és Pseudo Elementek

- Classes
  - :hover
  - :link
  - :not
  - :first-child
  - :focus
  - nth-child() (paramater: number, even, odd)
- Elements
  - ::first-line
  - ::first-letter
  - ::before
  - ::after
  - ::selection

```
table td:last-child {
   text-align: right;
}

table td:not(:last-child), th:not(:last-child) {
   border-right: 1px dotted;
}

table tbody tr:nth-child(even) {
   background-color: white;
}

a:visited {
   color: yellow;
}
a:link {
   background-color: red;
}
.combinators::first-line {
   font-size: 30px;
}
.combinators:hover {
   color: green;
}
```

Transition (animációk)

- két dolgot kell specifikálni
  - melyik tulajdonságon legyen az animáció (alapértelmezetten mindenen)
  - animáció időtartama (alapértelmezetten 0)
- az animáció érték változáskor kezdődik

```
<div class="div-image"></div>

.div-image {
   width: 100px;
   cursor: pointer;
   transition: width 1s, transform 2s;

   background-image: url(image.jpg);
   background-position: center;
   height: 100px;
   background-size: cover;
}

.div-image:hover {
   width: 200px;
   -webkit-transform: rotate(180deg); /* Safari */
   transform: rotate(180deg);
}
```

Flexbox

- könnyebb a használatával reszponzív weboldalt készíteni
- egy szülő és több gyermek elementből áll
- szülő element: display:flex;
- szülő elementen használható tulajdonságok
  - Flex-direction (column, row, column-reverse, ..)
  - Flex-wrap (wrap (break line if need), nowrap (all child inline), ...)

```
<div class="flex-container">
   <div>1</div>
   <div>2</div>
   <div>3</div>
   <div>4</div>
   <div>5</div>
   <div>6</div>
</div>

.flex-container {
   display: flex;
   height: 600px;
   flex-wrap: wrap;
   align-content: center;
   background-color: DodgerBlue;
}

.flex-container > div {
   background-color: #f1f1f1;
   width: 100px;
   margin: 10px;
   text-align: center;
   line-height: 75px;
   font-size: 30px;
}
```

Flexbox

- szülő elementen használható tulajdonságok...
  - Justify-content (align children, horizontal)
    - Center,
    - flex-start,
    - Space-around (equal space around children)
    - Space-between (equal space between children)
  - Align-items (vertical align children in parent)
  - Align-content (align item rows)

Media queries

- Ellenőrizni lehet vele
  - viewport szélességét, magasságát
  - készülék szélességét, magasságát
  - tájolást (portrait / landscape)
  - felbontást

```
@media screen and (max-width: 1080px) {
    .flex-container {
        background-color: transparent;
    }
}
```

Sass

- http://sass-lang.com/guide
- A CSS kiterjesztése
- CSS-re fordul, hogy a böngészőkben működjön
- Funkciók
  - változók
  - beágyazások
  - mixinek
  - importálás
  - öröklés
- Előnyök pl
  - kevesebb kód ismétlés
  - jobb karbantarthatóság
  - matematikai műveletek használata (+, -, *, /, %)

```
$width: 200px;

nav {
  ul {
    margin: 0;
    padding: 0;
    list-style: none;
  }
}

@mixin border-radius($radius) {
  -webkit-border-radius: $radius;
     -moz-border-radius: $radius;
      -ms-border-radius: $radius;
          border-radius: $radius;
}
.box { @include border-radius(10px); }

// _reset.scss

html,
body,
ul,
ol {
  margin:  0;
  padding: 0;
}

// base.scss
@import 'reset';
body {
  font: 100% Helvetica, sans-serif;
  background-color: #efefef;
}
```

Bootstrap

Bootstrap: A legnépszerűbb HTML, CSS, JS keretrendszer reszponzív weboldalak készítésére.

Ingyenes sablonok
https://startbootstrap.com/template-categories/all/

Komponensek
https://v4-alpha.getbootstrap.com/components/alerts/

Elrendezés
https://v4-alpha.getbootstrap.com/layout/overview/

Köszönöm a figyelmet!

Attrecto Zrt.
Attrecto Next Tech Digital Solutions

H-9024 Győr, Wesselényi str. 6.
info@attrecto.com