

# Fullstack 2024

**HTML & CSS** 5. prezentáció



# Elem név, Class, ID - Mikor melyiket?

- elemnév: `p { ... }`
  - elsődlegesen
- class `.osztalynev { ... }`
  - alap formázástól eltérő elemek
  - pl: 10 `<p>` bekezdésből két bekezdés más a többi 8-tól -> class
- id `#egyedi-azonosito { ... }`
  - ahol szükséges
  - bookmark-ok, input-ok

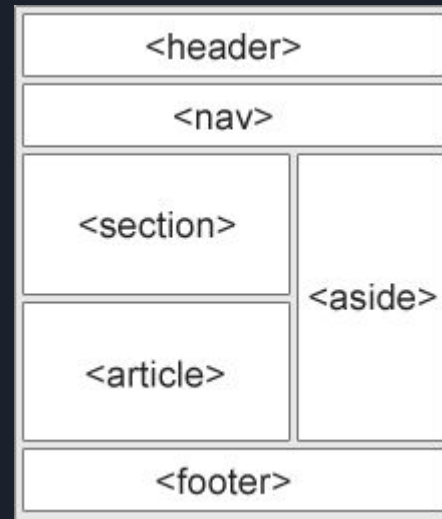
# Semantic HTML

Mitől és mitől nevezünk **szemantikus** egy html elem?

- semantic = “elements with meaning”
- h1, form, table, a, stb.
- nem szemantikus: div, span
- HTML szemantikus elemek

`<footer>` `<header>` `<main>`  
`<nav>` `<section>` `<article>` `<aside>`

`<summary>` `<time>` `<mark>` `<details>` `<figcaption>`  
`<figure>`



Referencia: [w3schools](https://www.w3schools.com/html/html5_semantic_elements.asp)



# CSS - Pszeudo osztályok (Pseudo classes)

```
selector:pseudo-class {  
  property: value;  
}
```

- elemek speciális állapotok formázása (hivatkozások: visited, active...)
- :hover, : active, :last-child, :first-child, :focus, :nth-child()
- teljes pszeudo class lista [itt](#)

# CSS - Pszeudo osztályok - linkek formázása

- `a:link` - a normal, unvisited link
- `a:visited` - a link the user has visited
- `a:hover` - a link when the user mouses over it
- `a:active` - a link the moment it is clicked

```
/* unvisited link */  
a:link {  
    color: red;  
}
```

```
/* visited link */  
a:visited {  
    color: green;  
}
```

```
/* mouse over link */  
a:hover {  
    color: hotpink;  
}
```

```
/* selected link */  
a:active {  
    color: blue;  
}
```



# Pszeudo elemek - pseudo elements

- jelölés: két darab kettőspont -> pseudo elem (::)
- ::first-line - bekezdés első sora
- ::first-letter - elem tartalmának első betűje
- ::before, ::after
- ::selection

```
selector::pseudo-element {  
  property: value;  
}
```

Referencia: [link](#)

# Listák formázása

- lista elem marker változtatása: `list-style-type`
  - értékek: circle, square, upper-roman - [példa](#)
- kép mint listaelem jelölő `list-style-image` [példa](#)
- jelölő pozicionálása: `list-style-position` [példa](#)
- rövidítés (shorthand): `list-style`
  - `list-style: square inside url("sqpurple.gif");`
- böngésző alapértelmezett beállításának törlése (reset)

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



# Feladat - horizontális navigációs menü

- 4 darab gomb
- `<a>` elemek
- padding/margin/ background
- `:hover` class - egér rávitelre sötétebb színű zöld szín legyen a háttér
- elemek vízszintes pozicionálása **float**-tal

## Horizontális navigációs menü

Link 1

Rólunk

Szolgáltatások

Kapcsolat





# CSS Display: block / inline / none

- a { `display:block` }
- alapértelmezett érték felülírása
- block elemek: div, p, h1
- inline elemek: img, span, a
- none - az elem eltüntetése, (pl. script tag alapértelmezetten )

`display:none;` vs `visibility:hidden;`

- `display:none` -> kiveszi az elemet a dokumentum flow-ból
- `visibility:hidden` -> láthatatlan, de a pozícióját megtartja, a többi elemhez képest is



# CSS Display: inline-block

- a { `display:inline-block` }
- van magasság és szélesség
- alsó/felső margók összecsúsznak, oldalsó margók marad
- különbség block-hoz képest: inline viselkedik, nem töri új sorba,

Referencia [w3schools](#), [példa](#)



# CSS overflow

Mi történjen akkor, ha az elem tartalma túl nagy, hogy kiferjen a neki szánt helyen

- **overflow:** <visible | hidden | scroll | auto>
- visible (alapértelmezett)
- csak block elemeken működik megadott (height) magasság értékkel

visible - Default. The overflow is not clipped. The content renders outside the element's box

hidden - The overflow is clipped, and the rest of the content will be invisible

scroll - The overflow is clipped, and a scrollbar is added to see the rest of the content

auto - Similar to scroll, but it adds scrollbars only when necessary

Példa

# CSS opacity - átlátszóság beállítása

```
img {  
  opacity: 0.5;  
}
```

```
img:hover {  
  opacity: 1.0;  
}
```

- értékek 0 és 1 között



[Referencia](#)