

# CSS-Cascading Style Sheets

# Mi is az a CSS?

- **CSS (Cascading Style Sheets):** Stíluslap-nyelv weboldalak formázásához.
- **Elválasztja a tartalmat és a megjelenést:** HTML a szerkezet, CSS a kinézet.
- **Egységes formázás:** Több oldal is használhatja ugyanazt a CSS-t.
- **Könnyen karbantartható:** A stílusok egyszerűen módosíthatók.
- **Reszponzív dizájn:** Különböző eszközökhöz alkalmazkodik.



# Hogyan formázunk egy weboldalt CSS segítségével?

- A weboldal elemeit különböző CSS szelektorokkal formázhatjuk:
- **Elemtípusok:** Például a div elem formázására használhatunk olyan tulajdonságokat, mint a szín, margó és betűméret: `div { color: blue; margin: 10px; font-size: 18px; }`.
- **ID használata:** Egyedi elemekhez, mint például a fejléc, az ID szelektort alkalmazzuk: `#header { background-color: lightgray; height: 100px; text-align: center; }`.
- **Osztályok használata:** Ha több elemre szeretnénk ugyanazt a stílust alkalmazni, használjuk az osztályokat: `.button { border: 1px solid black; padding: 10px 20px; border-radius: 5px; }`.
- **Beágyazott stíluslapok:** A stílusok HTML-ben is megadhatók `<style>` blokkokkal: `<style> p { color: green; font-size: 16px; } </style>`.
- **Különböző szelektorok:** Elem szelektorokkal (`p { margin: 15px; }`), ID-kkel és osztályokkal formázhatjuk az elemeket, míg az attribútum szelektor például így néz ki: `input[type="text"] { border: 1px solid black; }`.

# Pseudo-elemek

- ▶ A pseudo-elemek segítségével formázni tudjuk a weboldal elemeinek bizonyos részeit
- ▶ A pseudo-elemeket az elemek után írjuk, ::-al elválasztva őket
- ▶ Gyakran használt pseudo-elemek:
  - ▶ ::first-line
  - ▶ ::first-letter
  - ▶ ::before
  - ▶ ::after
  - ▶ ::selection

```
p{  
    color: red;  
}  
p::first-letter{  
    color: aqua;  
}  
p::before{  
    content: "Heló";  
}  
p::after{  
    content: "🤪";  
}
```



# Pseudo-osztályok

- ▶ A pseudo-osztályok segítségével változtatni tudunk az elemeken ha ezekkel történik valami
- ▶ Hasonlóan a pseudo-elemekhez, a pseudo-osztályokat az elemek után írjuk, :-al elválasztva őket
- ▶ A leggyakrabban használt pseudo-osztályok:
  - ▶ :hover
  - ▶ :link
  - ▶ :visited

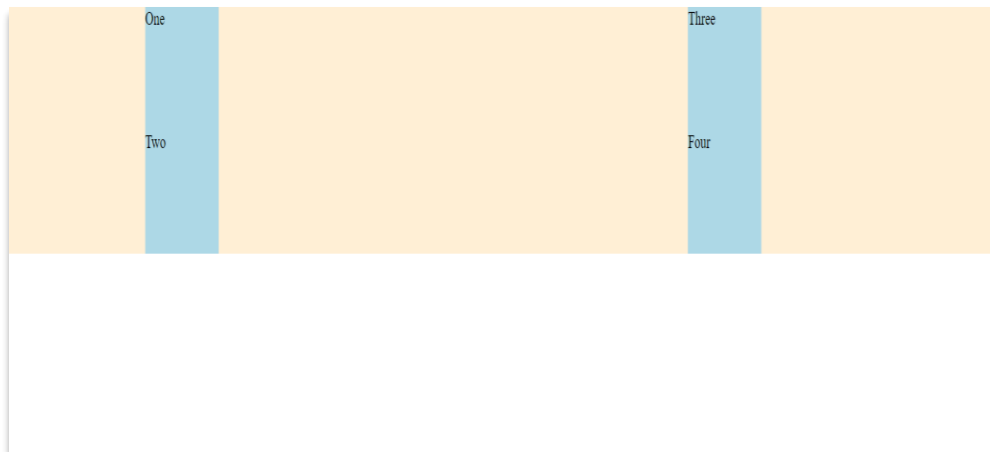
```
p {  
    display: none;  
    background-color: yellow;  
    padding: 20px;  
}  
  
div:hover p {  
    display: block;  
}
```

# Abszolút és relatív elhelyezés

- ▶ A CSS-ben két fő **pozicionálási mód** létezik:
- ▶ **Relatív elhelyezés:** Az elem a normál helyzetéhez képest lesz elmozdítva, de a helye a dokumentum áramlásában megmarad. **Példa:**
  - ▶ `position: relative; top: 20px; left: 10px;` - az elem 20px-re kerül le és 10px-re balra az eredeti helyétől.
- ▶ **Abszolút elhelyezés:** Az elem teljesen kikerül a normál dokumentumáramlásból, és egy szülő elemhez viszonyítva helyezkedik el. **Példa:**
  - ▶ `position: absolute; top: 50px; right: 30px;` - az elem a legközelebbi pozícionált szülő elemhez képest 50px-re lesz a tetejétől és 30px-re a jobb szélétől.

# A display tulajdonság

- ▶ A display tulajdonságot arra használjuk, hogy meghatározzuk hogyan jelenjen meg egy elem a weboldalon.
- ▶ A legtöbb elem alapértelmezett display értéke block vagy inline
- ▶ Gyakrabban használt, hasznos display értékek:
  - ▶ None
  - ▶ Grid
  - ▶ Flex



```
.container {  
  display: flex;  
  justify-content: center;  
  align-items: center;  
  flex-direction: column;  
  flex-wrap: wrap;  
  background-color: #papayawhip;  
  height: 200px;  
}  
.item {  
  width: 100px;  
  height: 100px;  
  background: #lightblue;  
}
```

# Alkalmazkodó webdesign

- ▶ A @media segítségével beállíthatunk olyan változtatásokat, amelyek csak bizonyos feltételek teljesülésekor lépnek érvénybe
- ▶ Ez hasznos lehet, ha azt szeretnénk, hogy többféle méretű kijelzőre is jól nézzen egy weboldal

```
@media (max-width: 1024px)
{
    h1{
        font-size: 2rem;
    }

    header{
        display: block;
        height: 20%;
    }
}
```

```
@media (min-width: 500px){
    div{
        background-color: #brown;
    }
}
```