

---

# HTML5 + CSS + Responsive Design

Szemantikus HTML, Flexbox, Grid, Media Query-k

---

---

# Miért fontos ez a téma?

- Webforgalom 60–70%-a mobilról jön
- A Google rangsorolásban előnyben részesíti a mobilbarát oldalakat
- A **szemantikus HTML** jobb SEO és „Accessibility”
- **Flex + Grid** = modern, tiszta elrendezés

SEO = **S**earch **E**ngine **O**ptimization

# HTML5 szemantikus elemek

- `<header>` fejlécek
- `<nav>` navigáció
- `<main>` fő tartalom
- `<section>` fejezet/tematikus blokk
- `<article>` önálló tartalmi egység
- `<aside>` oldalsáv
- `<footer>` lábléc

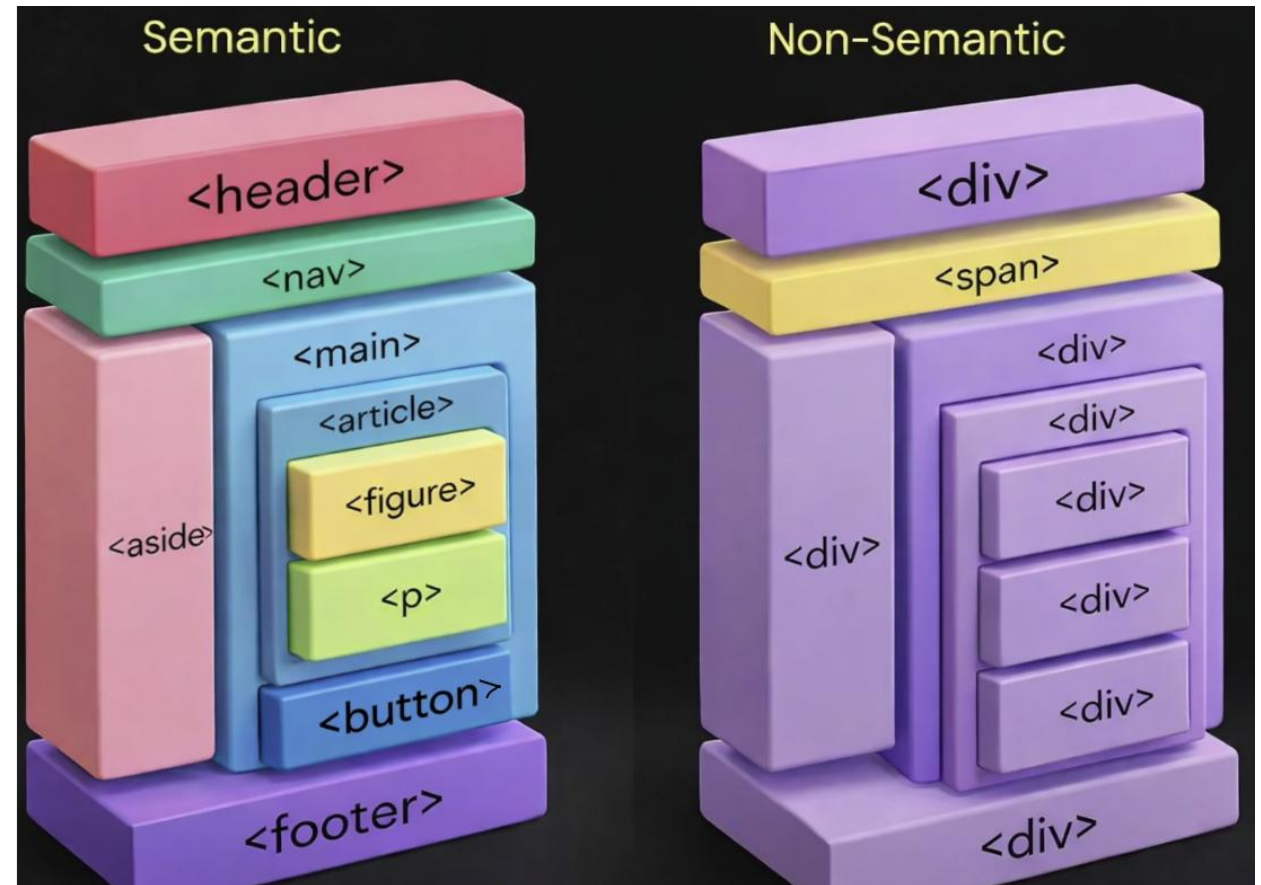
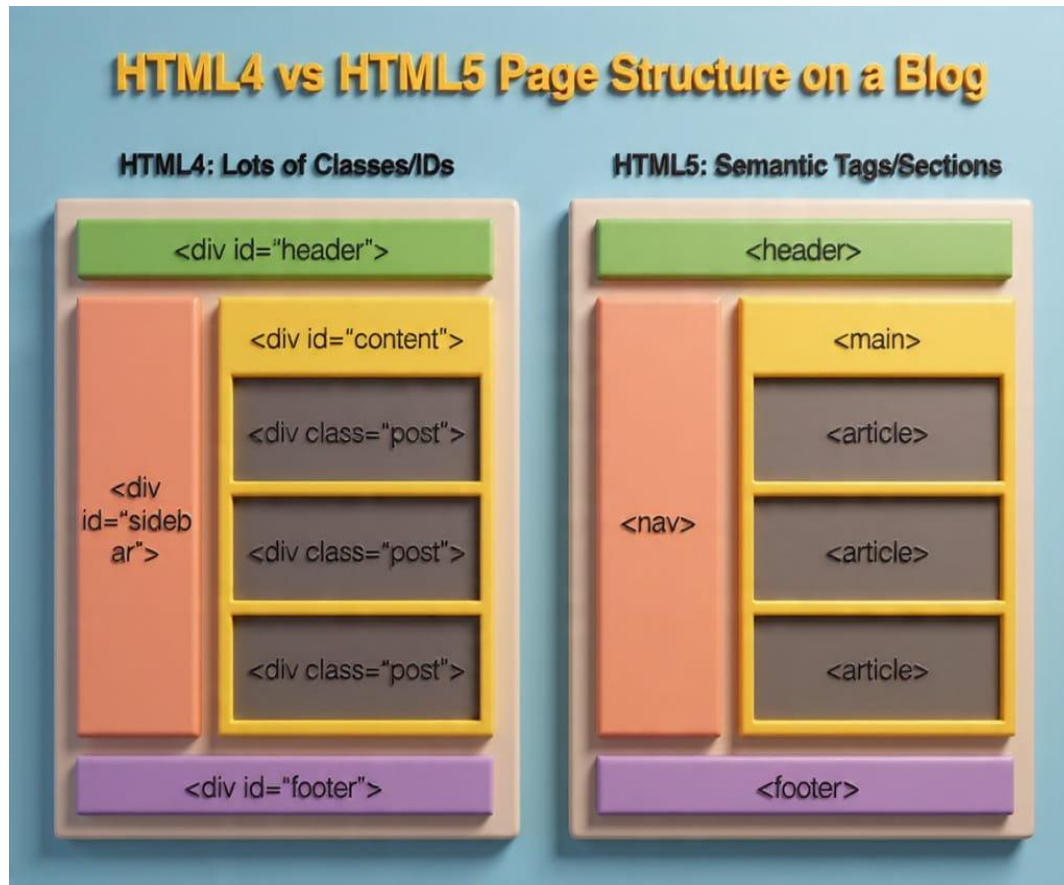


---

# Szemantikus HTML előnyei

- Kód könnyebben olvasható
- Jobb SEO
- Hozzáférhetőség (Accessibility)
- Kevesebb div, kevesebb káosz

# HTML4 vs HTML5



# Példa szemantikus HTML-re

```
<div class="header">
  <h1>My Blog</h1>
</div>

<div class="main">
  <div class="article">
    <h2>First Post</h2>
    <p>This is my first blog post.</p>
  </div>
</div>
```

Non-Semantic HTML

```
<header>
  <h1>My Blog</h1>
</header>

<main>
  <article>
    <h2>First Post</h2>
    <p>This is my first blog post.</p>
  </article>
</main>
```

Semantic HTML

---

# CSS Flexbox (áttekintés)

- Egy dimenziós elrendezés (sor vagy oszlop)
- Rugalmas igazítás
- Nagyon egyszerűen kezelhető
- Tipikus: menük, kártyák, oszlopok

# Flexbox példa

```
.cards {  
  display: flex;  
  gap: 10px;  
}  
  
.card {  
  background: #eee;  
  padding: 20px;  
  flex: 1;  
}
```

- **Card-ok** elhelyezése
- **display: flex;** - elemek sorba helyezése a Flexbox segítségével.



# Flexbox használata (HTML4 és HTML5)

```
<div class="cards">
  <div class="card">
    <h2>Card 1</h2>
    <p>First card content.</p>
  </div>

  <div class="card">
    <h2>Card 2</h2>
    <p>Second card content.</p>
  </div>

  <div class="card">
    <h2>Card 3</h2>
    <p>Third card content.</p>
  </div>
</div>
```

Non-Semantic HTML

```
<section class="cards">
  <article class="card">
    <h2>Card 1</h2>
    <p>First card content.</p>
  </article>

  <article class="card">
    <h2>Card 2</h2>
    <p>Second card content.</p>
  </article>

  <article class="card">
    <h2>Card 3</h2>
    <p>Third card content.</p>
  </article>
</section>
```

Semantic HTML

---

# CSS Grid (áttekintés)

- Két dimenziós (rács) elrendezés
- Oszlopok + sorok
- Bonyolultabb layoutok egyszerű létrehozása
- Galériák, dashboardok, teljes oldaltervek

# Grid példa

```
.grid {  
  display: grid;  
  gap: 10px;  
  grid-template-columns: repeat(3, 1fr);  
}
```

- Háromoszlopos rács elhelyezése
- **display: grid;** - aktiválja a Grid elrendezést

# Grid használata (HTML4 és HTML5)

```
.grid {  
  display: grid;  
  gap: 10px;  
  grid-template-columns: repeat(3, 1fr);  
}  
  
.item {  
  background: #eee;  
  padding: 20px;  
  text-align: center;  
}
```

```
<div class="grid">  
  <div class="item">Item 1</div>  
  <div class="item">Item 2</div>  
  <div class="item">Item 3</div>  
</div>
```

```
<section class="grid">  
  <article class="item">Item 1</article>  
  <article class="item">Item 2</article>  
  <article class="item">Item 3</article>  
</section>
```

---

# Responsive Design

- Cél: weboldal jól jelenjen meg minden eszközön

Például:

- **Mobil** → egy oszlop
  - **Tablet** → két oszlop
  - **Desktop** → három oszlop
- 
- Ehhez használjuk a **media query**-ket: feltételhez kötött CSS szabályok

# FlexBox + Media Query (egyszerű példa)

```
@media (max-width: 768px) {  
  .cards {  
    flex-direction: column;  
  }  
}
```

A **flex-direction: column;** definiálja az oszlopos elrendezést a **cards** CSS stílus elemen.

# Grid + Media Query (összetett példa)

```
.gallery {  
  display: grid;  
  grid-template-columns: repeat(3, 1fr);  
}  
  
@media (max-width: 992px) {  
  .gallery {  
    grid-template-columns: repeat(2, 1fr);  
  }  
}  
  
@media (max-width: 768px) {  
  .gallery {  
    grid-template-columns: 1fr;  
  }  
}
```

Itt láthatjuk a háromlépcsős responsive mintát:

- nagy kijelzőn 3 oszlop
- közepes kijelző (tablet) 2 oszlop
- mobilon 1 oszlop

# CSS keretrendszerek

- Miért alakultak ki a CSS keretrendszerek?
  - Ismétlődő CSS/layout problémák, egyedi CSS megoldások
  - A keretrendszerek erre adnak megoldást, egy egységes CSS szabálykészletet.
- Mit ad egy CSS keretrendszer?
  - Előre definiált layout (Flexbox / Grid)
  - Beépített responsive design
  - Újrahasznosítható komponensek
- Népszerű CSS keretrendszerek:
  - Bootstrap
  - Tailwind
  - Material UI



---

# Összefoglalás

- **HTML5** → szemantikus, tiszta struktúra
- **Flexbox** → Egy dimenzióban rugalmas elrendezés
- **Grid** → Két dimenziós rács
- **Media Query** → alkalmazkodás különböző eszközökhöz