

## **Miben különbözik az responsive design az adaptive design-től?**

A responsive és az adaptive design egyaránt megkísérli optimalizálni a felhasználói élményt a különböző eszközökön, alkalmazkodva a különböző nézetablakok méretéhez, felbontásaihoz, használati körülményeihez, vezérlési mechanizmusaihoz és így tovább.

A responsive design a rugalmasság elvén működik - egyetlen weboldal, amely bármilyen eszközön jól kinézhet. Az adaptív webhelyek media queries, flexible grids, and responsive images használnak a felhasználói élmény létrehozásához, amely a tényezők sokasága alapján rugalmas és változó.

Az adaptív design inkább hasonlít a progresszív fejlesztés modern meghatározására. Az adaptív design észleli az eszközt és az egyéb szolgáltatásokat, majd biztosítja a megfelelő funkciót és elrendezést egy előre meghatározott nézetablak-méret és egyéb jellemzők alapján. A webhely felismeri a használt eszköz típusát, és biztosítja az eszköz előre beállított elrendezését.

## **Mi a különbség a CSS „resetting” és „normalizing” között? Melyiket választanád, és miért?**

Resetting- az összes alapértelmezett böngészőstílus eltávolítása az elemektől. Például Az összes elem margói, padding-jei és betűméretei visszaállnak. A gyakori tipográfiai elemek stílusát újra be kell deklarálnia.

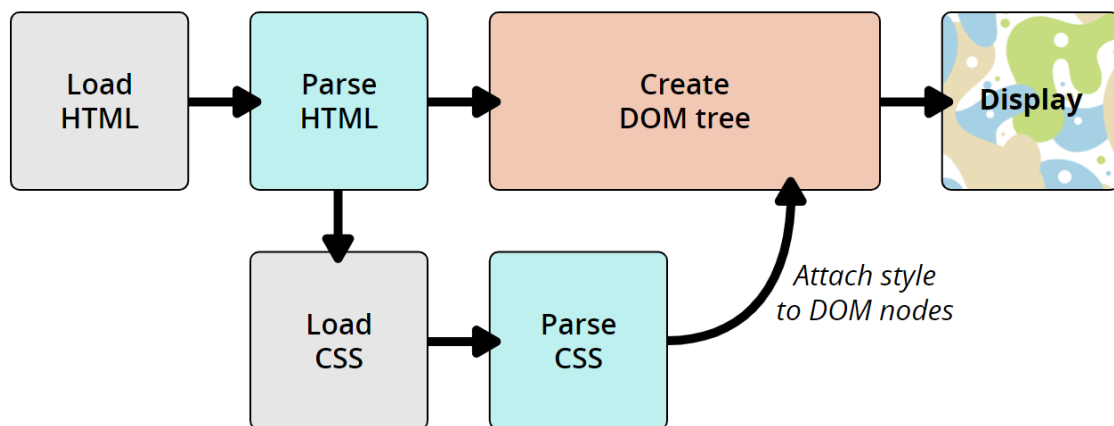
Normalizálás - megőrzi a hasznos alapértelmezett stílusokat, ahelyett, hogy mindent „kitörölne (unstyling)”. Javítja a böngésző függőségeinek hibáit is.

Célszerű a resetting-et választani, ha nagyon testreszabott vagy nem szokványos stílus design van, így sok saját stílust kell létrehozni és nincs szükség alapértelmezett stílusok megőrzésére.

## **Hogyan működik a CSS valójában ?**

Amikor a böngésző megjeleníti a dokumentumot, akkor kombinálnia kell a dokumentum tartalmát a stílusinformációival. A dokumentumot két szakaszban dolgozza fel:

- A böngésző a HTML-t és a CSS-t DOM-vá (Document Object Model) alakítja. A DOM képviseli a dokumentumot a számítógép memóriájában. Kombinálja a dokumentum tartalmát a stílusával.
- A böngésző megjeleníti a DOM tartalmát.



### Magyarázza el a "table-layout" tulajdonság használatát

A table-layout tulajdonság meghatározza a táblázat celláinak, sorainak és oszlopainak elrendezéséhez használt algoritmust.

`table-layout: auto|fixed|initial|inherit;`

- **auto** - A böngészők automatikus táblázatelrendezési algoritmust használnak. Az oszlopszélességet a cellák legszélesebb törhetetlenebb tartalma állítja be. A tartalom diktálja az elrendezést.
- **fix** - Az elrendezés az első sor alapján rögzített. Állítsa be ezek szélességét, és a táblázat többi része következik. Ha az első sorban nincs szélesség, az oszlopszélességeket egyenlően osztják el a táblázatban, függetlenül a cellákon belüli tartalomtól.
- **initial** - Ezt a tulajdonságot az alapértelmezett értékre állítja.
- **inherit** - Ezt a tulajdonságot örökli a szülő eleméből.

### Milyen módon lehet vizuálisan elrejteni a tartalmat (és csak a képernyőolvasók számára elérhetővé tenni)?

Ezek a technikák kapcsolódnak az akadálymentességhez (a11y).

- **visibility: hidden.** Az elem azonban még mindig az oldal folyamatában van, és még mindig helyet foglal.
- **width: 0; height: 0.** az elem egyáltalán nem foglal helyet a képernyőn, aminek következtében nem jelenik meg.
- **position: absolute; left: -9999px..** Helyezze a képernyőn kívülre.
- **text-indent: -9999px.** Ez csak a blokkelemek szövegén működik.
- **.sr-only osztály** a Bootstrap számára

## Mit jelent \* {box-sizing: border-box; }? Milyen előnyei vannak?

Alapértelmezés szerint az elemek box-sizing: content-box van alkalmazva, és csak a tartalom méretét veszik figyelembe.

box-sizing: a border-box megváltoztatja az elemek szélességének és magasságának kiszámításának módját, a border és a padding is beleszámít a számításba.

Az elem magasságát most a tartalom magassága + függőleges padding+ függőleges border szélessége határozza meg.

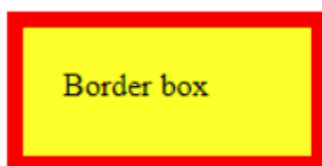
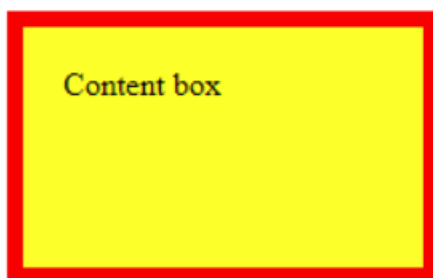
Az elem szélességét most a tartalom szélessége + vízszintes padding + vízszintes border szélessége határozza meg.

```
div {
  width: 160px;
  height: 80px;
  padding: 20px;
  border: 8px solid red;
  background: yellow;
}

.content-box {
  box-sizing: content-box;
  /* Total width: 160px + (2 * 20px) + (2 * 8px) = 216px
   Total height: 80px + (2 * 20px) + (2 * 8px) = 136px
   Content box width: 160px
   Content box height: 80px */
}

.border-box {
  box-sizing: border-box;
  /* Total width: 160px
   Total height: 80px
   Content box width: 160px - (2 * 20px) - (2 * 8px) = 104px
   Content box height: 80px - (2 * 20px) - (2 * 8px) = 24px */
}

<div class="content-box">Content box</div>
<br>
<div class="border-box">Border box</div>
```



## Magyarázza el a CSS-specifitás (specificity) alapvető szabályait

Az ID-választók specifitása magasabb, mint az attribútum-választóké.

```
/*wins*/  
a#a-02 { background-image : url(n.gif); }  
a[id="a-02"] { background-image : url(n.png); }
```

A kontextus selectorok specifikusabbak, mint egyetlen elemválasztó.

A beágyazott stíluslap közelebb áll a stílushoz.

Az utolsó definiált szabály felülír minden korábbi, ütköző szabályt.

```
p { color: red; background: yellow }  
p { color: green } // wins
```

Az osztályválasztó tetszőleges számú elemválasztót ver meg.

```
.introduction {} //wins  
html body div div h2 p {}
```

Az univerzális választó specifikussága 0, 0, 0, 0.

## Van valami oka annak, hogy a `translate()` szeretné használni az abszolút pozicionálás helyett, vagy fordítva? És miért?

A `translate()` a CSS transzformáció értéke. A `transform` vagy az `opacity` megváltoztatása nem váltja ki a böngésző újra renderelést, de konpozíciókat vált ki; mint az abszolút pozicionálás megváltoztatása visszavezetést vált ki. A `transform` hatására a böngésző GPU-réteget hoz létre az elem számára, de az abszolút pozicionálási tulajdonságok megváltoztatása a CPU-t használja. Ezért a `translate()` hatékonyabb, és rövidebb renderelési időket eredményez a simább animációk számára.

A `translate()` használatakor az elem továbbra is elfoglalja eredeti terét (hasonló `position: relative`-hoz), ellentétben az abszolút pozicionálással.