**1. Mikor használunk táblázatot és mikor nem?**

•

Táblázatokat gyakran használunk a szövegszerkesztési feladatok során is, hogy az **adatokat átlátható módon, srokoba és oszlopokba rendezve prezentáljuk**1. A webes táblázatok elsődleges célja is ez2.

•

A weblapkészítés korábbi időszakaiban a táblázatokat gyakran arra is használták, hogy az **oldal struktúráját** ezzel alakítsák ki1. Például a táblázat első sora volt a fejléc, a második sor két cellából állt, a bal oldali tartalmazta a menüt, a jobb oldali a tartalmat1.

•

Azonban a stíluslapokban (CSS) rejlő lehetőségek ma már **nem indokolják, hogy az oldal struktúrájának kialakítására, pusztán prezentációs célokból használjuk a táblázatokat**1. Ez a megoldás ma már **nem megfelelő**1.

•

Vannak azonban olyan esetek, amikor természetesen **nagyon előnyös** a táblázatok használata1.

**2. A táblázat alapvető részei: sor és cella.**

•

A táblázat megadása a **<table>** tag használatával lehetséges3. A táblázat tartalmát a <table> és </table> tagek között kell elhelyeznünk24.

•

A <table> tageken belül a **<tr>** (table row) és </tr> elemek között adhatjuk meg a táblázat **sorait**23. A <tr> tag használható a <thead>, <tbody>, <tfoot> és <table> tagek gyerekelemeként5.

•

A <tr> elem gyerekelemei a cellákat definiáló **<th>** és **<td>** elemek lehetnek5.

•

A **<td>** (table data) elem a sorokon belüli **adatcellák** létrehozására szolgál23. A <td> elem a <tr> elem gyereke26.

•

Az egyes sorok/oszlopok **fejléceit** a **<th>** (table header) tagek között adhatjuk meg23. Ezek alapesetben **kiemelve (félkövér)** jelennek meg23. Nagyon fontos, hogy a táblázatok fejléc elemeit **ne egyszerűen csak vizuálisan emeljük ki**, hanem a <th> elemet használjuk erre a célra2.... Ez segíti a képernyőolvasó programokat a táblázat tartalmának helyes értelmezésében és azonosításában7.... A <th> elem a <tr> elem gyereke211.

**3. Táblázat struktúrájának pontosítása caption, thead, tbody és a footer.**

•

A táblázat struktúrájának **precízebb megadására** használhatjuk a következő elemeket, különösen hosszabb táblázatok esetén8...:

◦

**<caption>**: A táblázat **feliratát** adja meg8.... A felirat jól leírja a táblázat tartalmát14, és akár több bekezdésből is állhat, rövid ismertetést tartalmazva15. Ez az elem közvetlenül a <table> elem után kell, hogy álljon, vagyis a <table> elem **első gyerekeként** kell elhelyeznünk8....

◦

**<thead>**: Ebbe kerül a táblázat **fejléce**8....

◦

**<tfoot>**: Ebbe kerül a táblázat **lábléce**8....

◦

**<tbody>**: Ebbe kerül a táblázat **törzse**8....

•

Ez a fajta felosztás lehetővé teszi a böngészőprogramoknak, hogy a táblázat törzsét **különválasszák a fejléctől és lábléctől**13. Ennek előnye például az lehet, hogy hosszabb táblázatok nyomtatásánál a fejléc és lábléc elemek **minden kinyomtatott oldalon szerepelni fognak** (amennyiben a böngésző támogatja ezt a funkciót)1319.

•

Az ajánlott sorrend a HTML kódban: Először a fejlécet (<thead>) kell megadnunk, azt követően pedig a láblécet (<tfoot>), és csak ezután adhatjuk meg a táblázat törzsét (<tbody>)13. Általános sorrend a <table> elemen belül (ahol használják őket): <caption>, <colgroup>, <thead>, <tfoot>, <tbody>15....

**4. Táblázat alapvető formázásai.**

•

Alapesetben a <th> tagek **félkövéren** jelennek meg23.

•

Ahhoz, hogy a táblázat cellái körül **szegély** is megjelenjen, használhatjuk a **border** paramétert a <table> tag esetén420. Ennek 1 az értéke, ha szegélyt akarunk420, és üresen kell hagyni, ha nem420.

•

Megjegyzés: **Stíluslapokkal (CSS) sokkal esztétikusabban** tudjuk formázni a táblázatokat, szebb szegélyeket is alkalmazhatunk2021. A modern webfejlesztésben a táblázatok megjelenítését **CSS-sel vezéreljük**, nem pedig régi HTML attribútumokkal (pl. border, cellpadding, cellspacing)2122.

•

A **sortable** logikai attribútum a <table> tagen4. Ha szerepeltetjük, akkor a táblázat adatait **sorrendbe rendezhetjük** a fejlécek szerint (ha a böngésző támogatja)4.

•

CSS segítségével megváltoztatható a táblázat **fejlécének (<th>) kinézete**, például háttérszín, szövegszín, betűvastagság beállításával23.

•

**Reszponzív táblázatok** esetén kis felbontáson a szokásos táblázatos felépítés nem használható24. Egy lehetséges megoldás a HTML és CSS kód módosítása, például a fejléc elrejtése CSS-sel és a cellák blokk-szintű (egymás alatti) megjelenítése, a fejléc szövegének pedig az adott cella elé történő kiírása (pl. az alt attribútum értékéből)24.

**5. Oszlop és sorcsoportok.**

•

Szükség lehet arra, hogy a szomszédos cellákat **vízszintesen, vagy függőlegesen** (vagy mindkét módon) **összevonjuk**12....

•

A **vízszintes összevonáshoz** a **colspan** paramétert kell használnunk12....

•

A **függőleges összevonáshoz** a **rowspan** paramétert kell használnunk12....

•

Ezeket az attribútumokat egyaránt alkalmazhatjuk a **<td>** és **<th>** tageken6....

•

A megadott szám jelzi, hogy **hány cellát** akarunk összevonni611. Például colspan="2" vagy rowspan="3"611.

•

Ha összevontunk cellákat, akkor a táblázatban már **csak azokat a cellákat kell szerepeltetni**, amelyeket nem érintett az összevonás25.

•

A cellaösszevonások hatással lesznek a fejléccella hatásköreinek (scope) beállítására is2627.

•

A táblázat **oszlopcsoportjait** adhatjuk meg a **<colgroup>** taggel12.... Ezek tulajdonságait is beállíthatjuk28. A <colgroup> tagnek a <caption> tag után, a <thead> tag előtt kell szerepelnie a HTML kódban1628.

•

A <colgroup> tagen belül szerepelhet a **<col>** tag is2829. Ez az egyes oszlopokba tartozó cellák tulajdonságait írja le2829. Ha a <colgroup> elemnek nincs span attribútuma, akkor használható a <col>28.

•

Ha a <colgroup> tagnek nincs beágyazott <col> eleme, akkor a **span** paraméterrel lehet megadni, hogy **hány oszlopot** csoportosít az adott tag1630. Például <colgroup span="2"></colgroup> jelzi, hogy az első két oszlop alkot egy oszlopcsoportot16.... A <col> tagen is használható a span a csoportosított oszlopok számának megadására29.