## Grid utasítások

A grid rendszer 2018-ban kifejlesztett technika, és az oldalon elhelyezett elemek pozícionálása szolgál.

Az oldalunkon elhelyezett elemek két csoportba oszthatók: vagy szülőelem, vagy gyerek elem. (persze előfordulhat, hogy egyazon elem néha szülőelemként, néha pedig gyerekelemként viselkedik).

**Az szülőelemet képzeletben oszlopokra és sorokra bontjuk, s megadjuk, hogy az elképzelt rács melyik téglalapjában melyik html elelmünk fog elhelyezkedni.**

Ez alapján (a flexbox-hoz hasonlóan) az utasítások itt is két csoportba oszthatók. Az utasítások egy része a szülőelemre alkalmazható, amásik része pedig a gyerek elemekre.

|  |  |
| --- | --- |
| **Szülőelemre alkalmazható utasítások**  **A rács kialakítása** | **Gyerekelemre alkalmazható utasítások**  **Elhelyezkedés a rácson** |
| display:grid;  grid-gap: 10px;  grid-template-columns:110px 110px 110px; | grid-column-start: 2;  grid-column-end: 3;  grid-row-start: 1;  grid-row-end: 2;  illetve rövidítve:  grid-column: 2 / 3;  grid-row: 1 / 2; |
| grid-template-areas:  "név név név név "  "mas mas ot ot"  "negy negy negy negy"; | grid-area: név; |
| justify-items: start | end | center | stretch; | align-self: end | start | center | strech; |

Hozzuk létre az alábbi oldalt a szörnyecskékkel. A képek mérete egységesen 100px x 100px. Külön css-ben készítsük el az alap formázásokat, s csatoljuk a html fájlunkhoz. Ezeket minden további magyarázat nélkül mellékelem, a korábban tanulak alapján elkészíthető:

|  |  |
| --- | --- |
| A html: |  |
| A css: |  |

### példa

A továbbiakban vagy külön css-ben készítjük el az egyes szerkezeti kialakításaokat (nem a formázáshoz használt css-ben) vagy belső stíluslapot használunk.

Az article tag-en belül szeretnénk a képeket elrendezni a mintáknak megfelelően. Az article tag tartalmazza a képeket. Minden képhez saját div-ben van elhelyezve.

Az article tag a szülőelem a div tag-ek pedig a gyerek elemek.

Állítsuk be a szülőelemre (article) a display:grid; utasítást, ezzel jelezzük, hogy mostatntól grid elemként használjuk a tárolót. Ez azonban még kevés, **azt is meg kell mondanunk, hogy hány oszlopa, esetleg hány sora (vagy mindkettő) lesz a kialakított rácsunknak, amiben elhelyezzük a gyerek elemeket.**

Erre többféle lehetőség is mutatkozik. Az alábbi például létrehoz 4 oszlops és 3 soros rácsot, rendre a példában megadott szélességekkel.

grid-template-columns: 110px 110px 110px 110px;

grid-template-rows: 200px 100px 300px;

A grid-gap utasítással a rácsok közötti távolságot állíthatjuk be.

article {

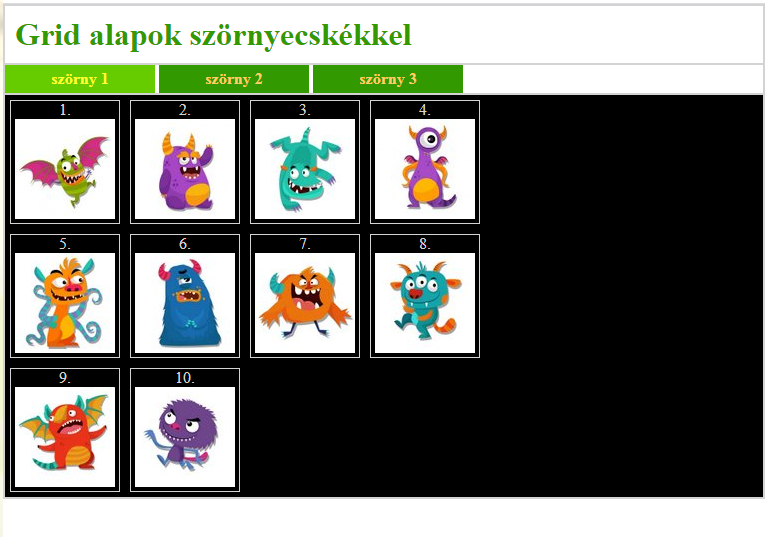
/\*szülőelem\*/

display:grid;

grid-gap: 10px;

grid-template-columns: 110px 110px 110px 110px ;

}



grid-template-columns: 1fr 1fr 1fr 1fr ;

Így az oszlopok szélessége egyenletesen oszlik el a szülőelem teljes szélességében.



grid-template-columns: auto auto auto auto;

utasításnak hasonló a hatása azonos szélességű elemek esetén. Nem azonos szélességű elemek esetén a gyerekelemek szélességéhez fog igazodni az egyes oszlopok szélessége.

Adjuk meg az egyes szörnyecskék pozícióját! Ehhez meg kell adnunk a létrehozott rácson a sor-oszlop koordinátákat minden egyes képet tartalmazó div esetében:

article div:nth-child(1) { /\*Az első div helye \*/

grid-column-start: 2; /\*kezdő oszlop \*/

grid-column-end: 3; /\*utolsó utáni oszlop \*/

grid-row-start: 1; /\*kezdő sor \*/

grid-row-end: 2; /\*utolsó utáni sor \*/

}

Vagy rövidítve:

article div:nth-child(2) {

grid-column: 2 / 3;

grid-row: 1 / 2;

}

Másfajta rövidítés:

article div:nth-child(2) {

grid-area: 1 / 2 / 2 / 3;

}

Helyezd el a szörnyecskéket csak a css megváltoztatásával a mintának megfelelően: (színek és szemszámok szerint vannak sorba, ileltve oszlopba rendezve)



A col és a span megadja,hogy hány rácsrészt foglal el az adott gyerekelem. A col az egy oszlopba tartozó rácsokat, a span pedig az egy sorba tartozó rácsokat fogja össze.

Az előző példában a 6. div-re alkalmazva a mintán láthatóan módosul a kék szörnyet tartalmazó div helyzete: (módosítsd a 9. és az 1. elemhez tartozó kódot is, hogy a mintának megfelelően alakuljon az oldal

article div:nth-child(6) {

grid-column:2 / span 4;

grid-row: 1 /2;

}



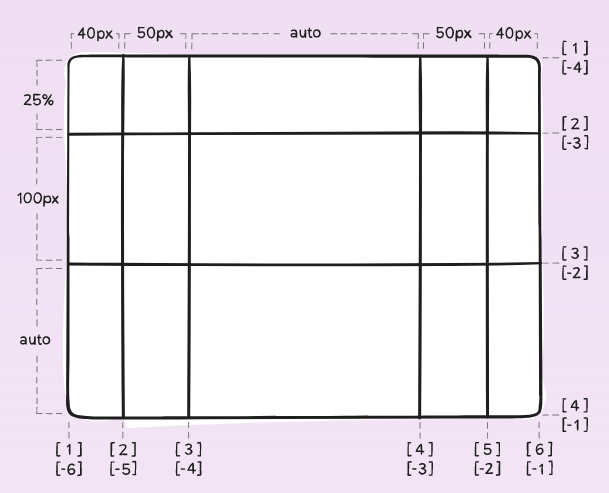
### Példák rácsok megadására:

.container {

grid-template-columns: 40px 50px auto 50px 40px;

grid-template-rows: 25% 100px auto;

}

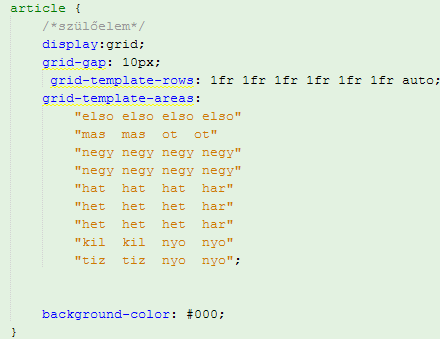


### példa: grid-area megadása nevekkel

Nevezzük el az egyes gyereklemeket a grid-area utasítással. Pl.:

article div:nth-child(1) { grid-area: elso; }

Képzeljünk el egy rácsot, amely 4 oszlopból és 9 sorból áll. A szülőelemnél meg kell adnunk, hogy melyik rácselemet melyik nevű gyerek foglalja el. Arra kell csak vigyázni, a területek összefüggő téglalapok legyenek, és az összes elemet felsoroljuk. A sorrend tetszőleges lehet:



Fontos: Ha elírjuk a nevet, az egész oldalszerkezetből semmi nem fog látszódni!

Próbáljuk ki úgy is, hogy a grid-template-rows: 1fr 1fr 1fr 1fr 1fr 1fr auto; sort kihagyjuk, vagy minden értékét auto-ra állítjuk! Figyeljük meg a különbséget!

