

Դաս 9 - Grid

Grid-ը դասավորության համակարգ է, որը հնարավորություն է տալիս երկու ուղղություններով (տողեր և սյուներ) ձևավորել ցանց: Ի տարբերություն **Flexbox**-ի, որը միակողմանի համակարգ է, Grid-ը երկկողմանի է և առավել ճկուն է մեծ նախագծերի համար:

```
.container {  
    display: grid;  
}
```

Ի տարբերություն **flexbox**-ի, տեգերը անմիջապես չեն փոփոխվի: **flex**-ը տալիս է մեկ սյունակի ցանց, այնպես որ ձեր տարրերը կշարունակեն ցուցադրվել մեկը մյուսի տակ, ինչպես դա անում են սովորական հոսքի ժամանակ:

Որպեսզի ցանցային տեսք ունենաք, մենք պետք է ավելացնել նոր սյունակներ:

```
.container {  
    display: grid;  
    grid-template-columns: 200px 200px 200px;  
}
```

Բացի պիկսելները, տոկոսները և այլ չափման միավորները, կարող եմք օգտագործել **fr (fraction)**

```
.container {  
    display: grid;  
    grid-template-columns: 1fr 1fr 1fr;  
}
```

Սյունակների միջև տարածությունն սահմանելու համար կարող եմք կիրառել **gap** հատկությունը, ընդ որում երեք եղանակով`

Column-gap - սյունակների միջև հեռավորություն

Row-gap - տողերի միջև հեռավորություն

Gap(grid-gap հիմք.) - սյուների և տողերի միջև հեռավորություն

```
.container {  
    display: grid;  
    grid-template-columns: 2fr 1fr 1fr;  
    gap: 20px;  
}
```

Grid-ը հնարավորություն է տալիս կրկնելու սյուները կամ տողերը ըստ մեր ցանկության, կիրառելով **repeat()** ֆունկցիան՝

```
.container {  
    display: grid;  
    grid-template-columns: repeat(3, 1fr);  
    gap: 20px;  
}
```

Մինչ այս մենք նշում էինք միայն սյունակները, իսկ տողերը ստեղծվում էին ավտոմատ կերպով: Ցանցերը իրենց առումով լինում են երկու տեսակի՝ **անուղղակի** և **ուղղակի**:

Ուղղակի ցանցը ստեղծվում է **grid-template-columns** կամ **grid-template-rows** օգնությամբ, իսկ **անուղղակի** ցանցը ընդլայնվում է ուղղակի ցանցերի սահմաններից դուրս, եթե պարունակությունը չի տեղավորվում **ուղղակի** ցանցերի մեջ: **Անուղղակի** ստեղծված ցանցերը ունեն ավտոմատ չափ, հետևաբար կոստենտը ցանկացած դեպքում տեղավորվելու է: Կարող ես նախապես սահմանել անուղղակի ցանցերի չափերը՝

```
.container {  
    display: grid;  
    grid-template-columns: repeat(3, 1fr);  
    grid-auto-rows: 200px;  
    gap: 20px;  
}
```

Այժմ սահմանենք լողացող տողերի չափեր: Եթե մեր տեգերի բարձրությունը գերազանցի մեր կողմից սահմանված 200px, ապա այն կսկսի ծածկել մյուս տեգերին, ինչը կհանգեցնի անկանխատեսելի արդյունքի: Քանի որ նախապես չգիտենք թե ներքին տեգերը ինչ չափի են լինելու կարող ենք կիրառել `minmax()` ֆունկցիան: Այն թույլ է տալիս սահմանել փոքրագույն և մեծագույն արժեքները:

```
.container {
  display: grid;
  grid-template-columns: repeat(3, 1fr);
  grid-auto-rows: minmax(100px, auto);
  gap: 20px;
}
```

Որոշ դեպքերում մեզ կարող է անհրաժեշտ լինի, ստեղծել այնքան սյունակներ, որքան կարող ենք սահմանված երկարությունում: Մենք կարող ենք սահմանել **grid-template-columns repeat()** ֆունկցիայի միջոցով, սակայն ֆիքսված թիվ փոխանցելու փոխարեն կարող ենք փոխանցել **auto-fit** հատուկ անունը:

```
.container {
  display: grid;
  grid-template-columns: repeat(auto-fit,
minmax(200px, 1fr));
  grid-auto-rows: minmax(100px, auto);
  gap: 20px;
}
```

Այժմ ցանցի ստեղծումից անցում կատարենք ցանցի վրա տեգերը տեղադրելուն:

Որպեսզի կարողանանք տեղադրել մեր տարրերը ցանցում կարող ենք կիրառել հետևյալ չորս հատկությունները`

grid-column-start,
grid-column-end,
grid-row-start,
grid-row-end,

Կարող եմք տեղակայել մեր տարրերը ցանցում նաև կիրառելով **grid-template-areas:** Դրա համար նախապես պետք է սահմանենք անուններ մեր տեգերին և տեղաբաշխենք դրանք ցանցում՝

```
.container {  
    display: grid;  
    grid-template-areas:  
        "header header"  
        "sidebar content"  
        "footer footer";  
    grid-template-columns: 1fr 3fr;  
    gap: 20px;  
}  
  
header {  
    grid-area: header;  
}  
  
article {  
    grid-area: content;  
}  
  
aside {  
    grid-area: sidebar;  
}  
  
footer {
```

```
grid-area: footer;
}
```

Եթե ցանկանում ես դատարկ տեղ թողնել, կարող ես կիրառել .
հատուկ սիմվոլը:

Ներկառուցված ցաներ

Կարող ես անգամ ցանցերի մեջ տեղադրել մեկ այլ ցանցեր: