

Դաս 8 - Flexbox

Flex-ը նախատեսված է **Էլեմենտների** արդյունավետ դասակարգման համար: **Flexbox**-ը (կարճ՝ **Flex**) հիմնականում օգտագործվում է **Էլեմենտների** համար, որոնք դասավորված են մեկ ուղղությամբ՝ տողերով կամ սյուներով: Այս տեխնոլոգիայի վրա սկսել են աշխատել դեռ 2009 թվականից: 2012թ. ին ցուցադրվել է արդեն պատրաստի տարբերակը, իսկ 2015թ. ից արդեն հասանելի է եղել գրեթե բոլոր բրաուզերներում: **Flexbox**-ը ապահովում է ավելի մեծ ճկունություն, երբ մենք աշխատում ենք **Էլեմենտների** դասավորության հետ: **Flexbox**-ը լուծում է շատ խնդիրներ, որոնք առաջանում են, երբ անհրաժեշտ է ճկունորեն կարգավորել տարրերի դիրքը և չափը:

Flexbox-ը հարմար է այն դեպքերի համար, երբ անհրաժեշտ է դասավորել **Էլեմենտները** միայն մեկ ուղղությամբ՝ կամ տողերով (**horizontal row**), կամ սյուներով (**vertical column**). Այն թույլ է տալիս.

- Վերահսկել **Էլեմենտների** չափերը.
- Ավտոմատ լրացնել ազատ տարածքը.
- Դասավորել **Էլեմենտները** հավասարապես կամ ճկուն կերպով:

Flexbox-ի հիմնական տարրը **flex-container**-ն է, որի մեջ պարունակվում են **flex-items**-ները:

- **Flex Container** – **flex** տեգ է, որը պարունակում է տեգեր որոնք պետք է դասավորվեն ըստ **flex** հատկությունների:
- **Flex Items** – կոնտեյների մեջ գտնվող տարրերը, որոնք կդասավորվեն ըստ սահմանված կանոնների.

Flexbox-ի հիմնական հատկությունները՝

display: flex - դարձնում է տեգը **flex-container**, ինչը թույլ է տալիս ներսի տեգերին դասավորել ըստ flexbox կանոնների:

flex-direction - Նշում է թե որ ուղղությամբ պետք է դասավորվեն տեգերը:

```
.container {  
  flex-direction: row;  
}
```

row - Տարրերը դասավորվում են տողով՝ կողք կողքի:

row-reverse - Տարրերը դասավորվում են տողով՝ հակառակ հերթականությամբ:

column - Տարրերը դասավորվում են սյունով՝ իրար տակ:

column-reverse - Տարրերը դասավորվում են սյունով՝ հակառակ հերթականությամբ:

flex-wrap - Սահմանում է տեգերի վարքագիծը, երբ նրանց ընդհանուր երկարությունը կամ բարձրությունը գերազանցում է **flex-container**-ինը:

```
.container {  
  flex-wrap: wrap;  
}
```

nowrap- Տարրերը նոր տող (սյուն) չեն տեղափոխվում:

wrap- Տարրերը տեղափոխվում նոր տող (սյուն) եթե կա անհրաժեշտություն: (տողադարձ)

wrap-reverse- Տարրերը տեղափոխվում նոր տող (սյուն) հակառակ հերթականությամբ:

flex-flow - Քանի որ սովորաբար flex-container օգտագործելիս մշտապես կիրառվում են **flex-direction** և **flex-wrap**, դրա համար սահմանվել է **flex-flow** -ն որը միավորում է այս երկու հատկությունները:

```
.container {  
  flex-flow: row wrap;  
}
```

}

justify-content - Սահմանում է, թե ինչպես պետք է տարրերը տարածվեն գլխավոր առանցքի ուղղությամբ (հորիզոնական, եթե **row**, կամ ուղղահայաց, եթե **column**):

flex-start: Տարրերը դասավորվում են սկիզբից:

flex-end: Տարրերը դասավորվում են վերջից:

center: Տարրերը դասավորվում են կենտրոնում:

space-between: Տարրերի միջև հավասար տարածություն է սահմանվում, իսկ առաջին և վերջին տարրերը տեղակայվում են եզրերի մոտ:

space-around: Տարրերի միջև հավասար տարածություն է սահմանվում, իսկ եզրերի միջև՝ սահմանվում տարածության կեսը:

space-evenly: Տարրերի միջև հավասար տարածություն է սահմանվում, ներառյալ նաև եզրերը:

```
.container {  
  justify-content: center;  
}
```

align-items - Սահմանում է, թե ինչպես պետք է տարրերը տարածվեն գլխավոր առանցքի ուղղությամբ (հորիզոնական, եթե **column**, կամ ուղղահայաց, եթե **row**):

flex-start: Տարրերը դասավորվում են վերևում:

flex-end: Տարրերը դասավորվում են ներքևում:

center: Տարրերը դասավորվում են կենտրոնում:

stretch: Տարրերը զբաղեցնում են ողջ հասանելի բարձրությունը:

baseline: Տարրերը դասավորվում են ըստ բազային դիրքի:

align-content - Սահմանում է, տարրերի միջև ազատ տարածության բաշխումը, եթե այդ տողերը զբաղեցնում են քիչ բարձրություն քան:

flex-grow: հատկությունը սահմանում է, թե որքան պետք է տեղն աճի իր հարևանների համեմատ՝ եթե ազատ տարածք կա **flex-container**-ում: Այս հատկությունը ընդունում է դրական թվային արժեք, որը ցույց է տալիս աճելու գործակիցը:

```
.item {  
    flex-grow: 2;  
}
```

Այս դեպքում `.item` կլաս պարունակող տեգերը կաճեն 2 անգամ, քան այն տարրերը, որոնց սահմանել ենք **flex-grow 1**:

flex-shrink: հատկությունը թույլ է տալիս որոշել, թե որքան պետք է փոքրանա տարրը, երբ կոնտեյներում տեղը քիչ է: Այս հատկությունը նույնպես ընդունում է թվային արժեք: Եթե այն մեծ է, տարրը ավելի շատ կպակասի:

```
.item {  
    flex-shrink: 1;  
}
```

flex-basis: հատկությունը սահմանում է, թե որքա՞ն տարածք պետք է սկզբում զբաղեցնի տարրը մինչև նրա մեծացումը կամ փոքրացումը: Այն գործնականում աշխատում է որպես տարրի սկզբնական չափի սահմանիչ:

```
.item {  
    flex-basis: 200px;  
}
```

flex: կարճ գրառումն է, որը միավորում է երեք հատկությունները՝ **flex-grow**, **flex-shrink** և **flex-basis**-ը:

```
.item {  
    flex: 1 0 100px;  
}
```

Այս դեպքում տարրը կունենա **flex-grow: 1**, **flex-shrink: 0** և **flex-basis: 100px**:

1 — տարրը աճում է ազատ տարածքի 1 մասով:

0 — տարրը չի փոքրանում, եթե տարածքը փոքրանում է:

100px — տարրի նախնական չափսը 100px է:

order հատկությունը թույլ է տալիս վերահսկել տարրերի հերթականությունը՝ անկախ նրանից, թե նրանք որ հերթականությամբ են գրված HTML-ում:

order-ի արժեքը ըստ լռելյայն կարգի 0 է: Ավելի փոքր արժեքները նշանակում են ավելի վաղ, ավելի մեծ արժեքները՝ ավելի ուշ ցուցադրում:

```
.item {  
    order: 2;  
}
```

align-self-ը թույլ է տալիս վերահսկել մեկ կոնկրետ տարրի դիրքը իր կոնտեյնների ներսում՝ անկախ մյուսների հետ հարաբերությունից:

Այս հատկությունը ընդունում է նույն արժեքները, ինչ align-items, բայց ազդում է միայն մեկ տարրի վրա:

```
.item {  
    align-self: center;  
}
```

<https://flexboxfroggy.com/>