

[https://www.w3schools.com/css/css3\\_flexbox.asp](https://www.w3schools.com/css/css3_flexbox.asp)

## Flexbox Elements

A Flexbox modell használatának megkezdéséhez először meg kell határozni egy flex tárolót.

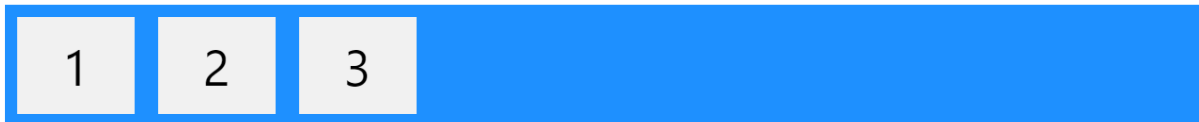


A fenti elem egy flex tárolót (a kék területet) képvisel, három flex elemmel.

```
<div class="flex-container">
  <div>1</div>
  <div>2</div>
  <div>3</div>
</div>
```

## CSS Flex Container

Az előző fejezetben leírtakhoz hasonlóan ez egy flex tároló (a kék terület) három flex elemmel:



A flex tároló rugalmassá válik, ha a display property-t flex-re állítja:

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

flex container properties:

- [flex-direction](#)
- [flex-wrap](#)
- [flex-flow](#)
- [justify-content](#)
- [align-items](#)
- [align-content](#)

## The flex-direction Property

A flex-direction tulajdonság meghatározza, hogy a tároló melyik irányba akarja rakni a flex elemeket.



Az column érték függőlegesen (felülről lefelé) halmozza a flex elemeket:

```
.flex-container {  
  display: flex;  
  flex-direction: column;  
}
```



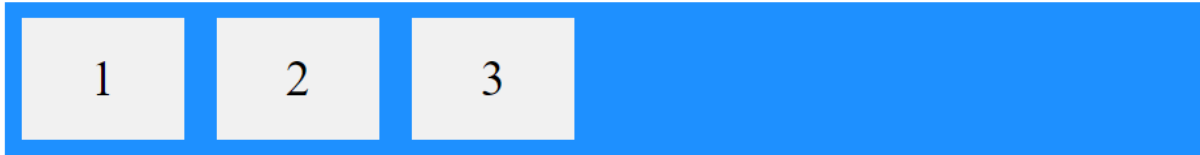
Az column-reverse érték függőlegesen (de lentől felfelé) halmozza a flex elemeket:

```
.flex-container {  
  display: flex;  
  flex-direction: column-reverse;  
}
```



A row értéke vízszintesen (balról jobbra) halmozza a flex elemeket:

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



A row-reverse értéke vízszintesen (de jobbról balra) halmozza a flex elemeket:

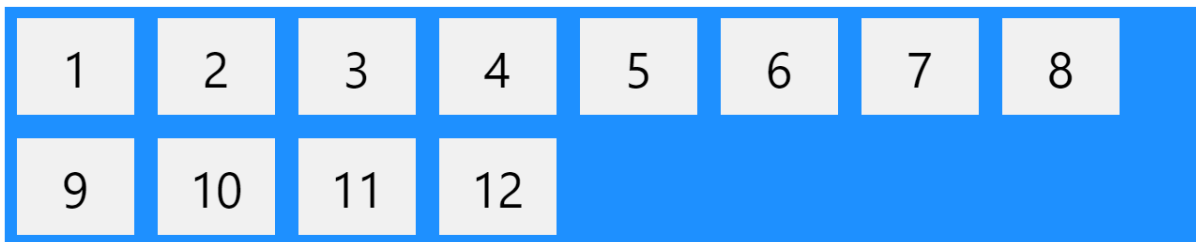
```
.flex-container {  
  display: flex;  
  flex-direction: row-reverse;  
}
```



## The flex-wrap Property

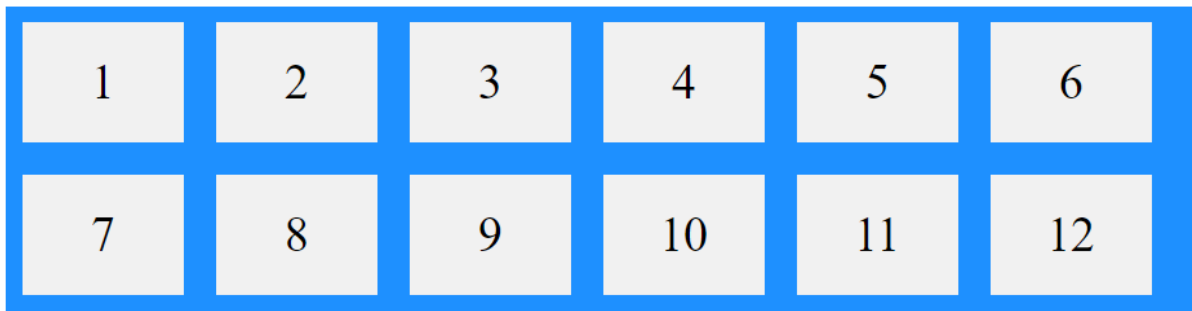
A flex-wrap tulajdonság megadja, hogy a flex elemeket be kell-e csomagolni vagy sem.

Az alábbi példák 12 flex elemet tartalmaznak, hogy jobban bemutassák a flex-wrap tulajdonságot.



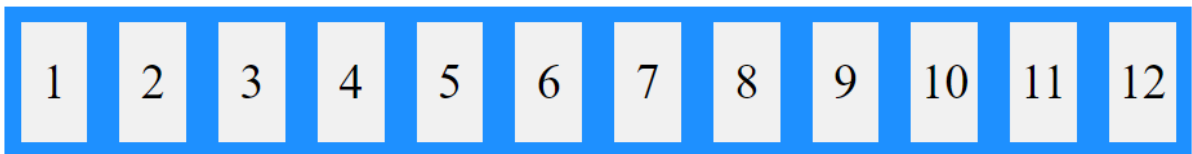
A wrap érték megadja, hogy a flex elemeket szükség esetén be kell csomagolni:

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



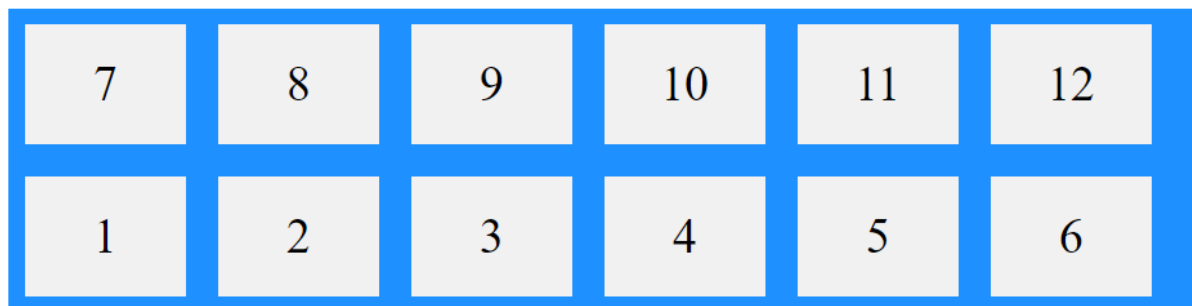
A nowrap érték megadja, hogy a flex elemek nem kerülnek feltekerésre (ez az alapértelmezett):

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



A wrap-reverse érték megadja, hogy a flex elemek szükség esetén fordított sorrendben tekerjenek:

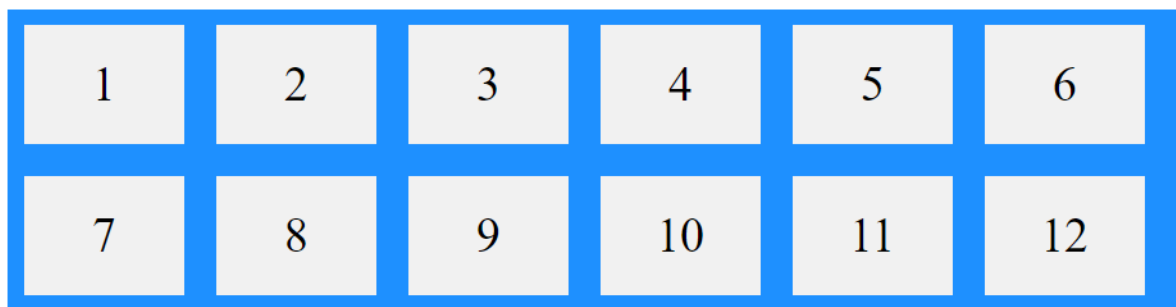
```
.flex-container {  
  display: flex;  
  flex-wrap: wrap-reverse;  
}
```



## The flex-flow Property

A flex-flow tulajdonság rövidített tulajdonság mind a flex-direction, mind a flex-wrap tulajdonságok beállításához.

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

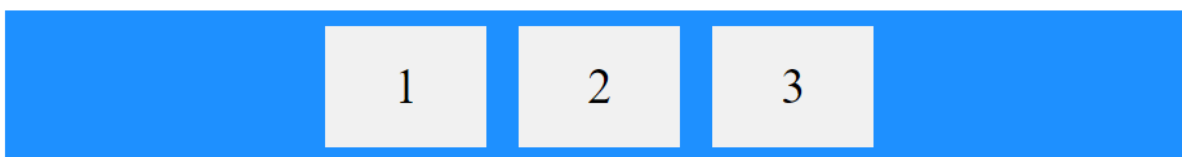


## The justify-content Property

A justify-content tulajdonság a flex elemek rendezésére (align) szolgál:

A center a flex elemeket a tároló közepére igazítja:

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



A flex-start érték összehangolja a flex elemeket a tároló elején (ez az alapértelmezett):

```
.flex-container {  
  display: flex;  
  justify-content: flex-start;  
}
```



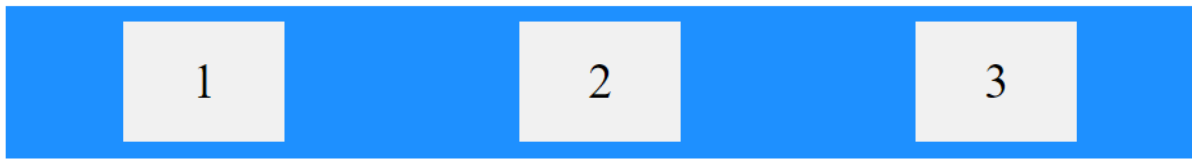
A flex-end érték igazítja a flex elemeket a tároló végére:

```
.flex-container {  
  display: flex;  
  justify-content: flex-end;  
}
```



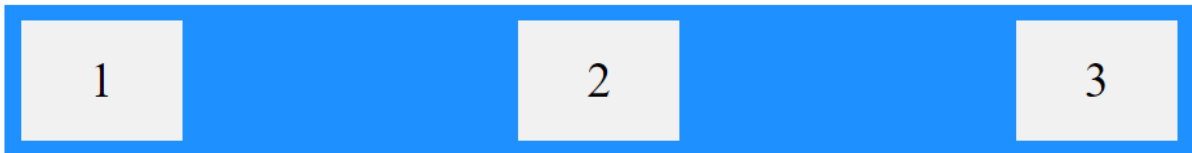
A space-around érték a flex elemeket a sorok előtti, közötti és utáni szóközzel jeleníti meg:

```
.flex-container {
  display: flex;
  justify-content: space-around;
}
```



A space-between érték a flex elemeket a sorok közötti szóközzel jeleníti meg:

```
.flex-container {
  display: flex;
  justify-content: space-between;
}
```

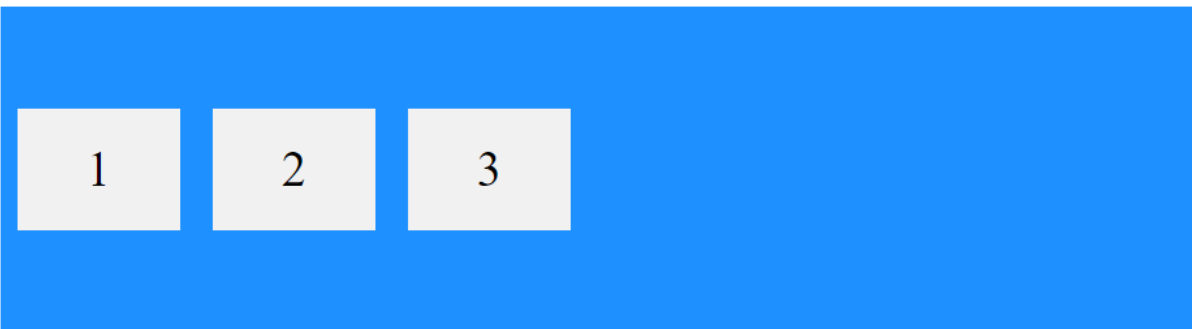


## The align-items Property

Az align-items tulajdonság a rugalmas elemek igazítására szolgál.

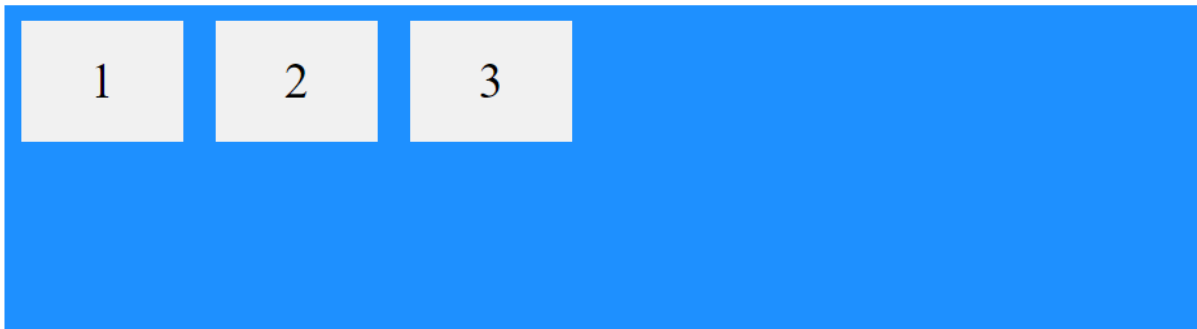
A center igazítja a tároló közepén lévő flex elemeket:

```
.flex-container {
  display: flex;
  height: 200px;
  align-items: center;
}
```



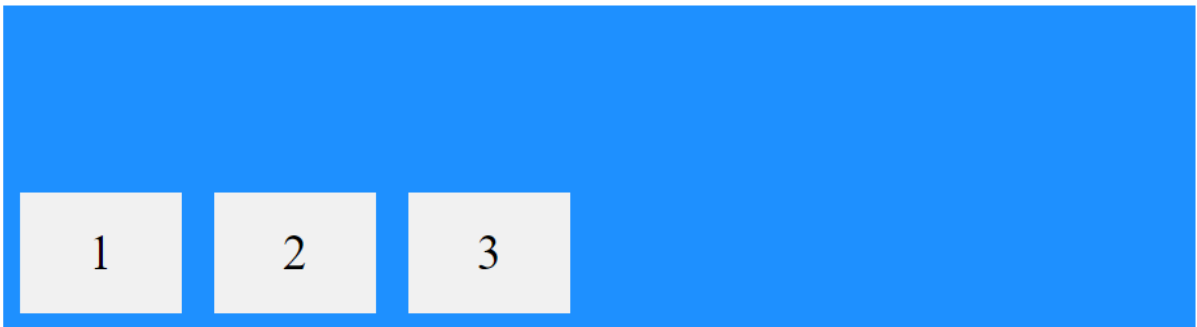
A flex-start érték összehangolja a rugalmas elemeket a tároló tetején:

```
.flex-container {
  display: flex;
  height: 200px;
  align-items: flex-start;
}
```



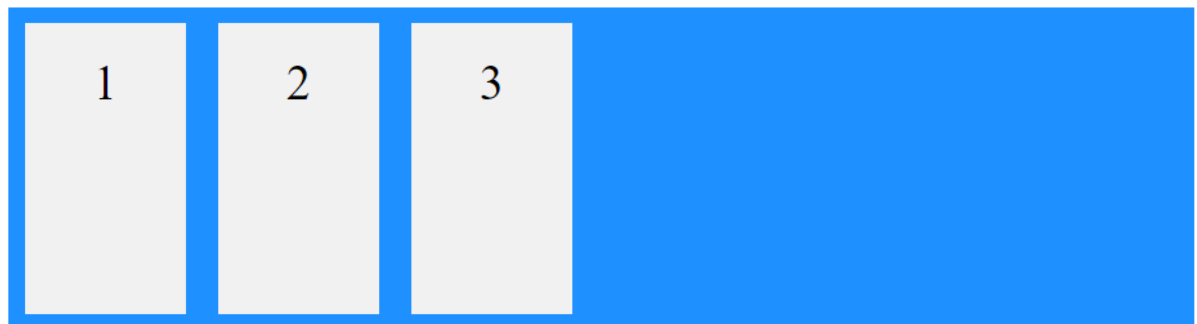
A flex-end érték igazítja a rugalmas elemeket a tároló alján:

```
.flex-container {  
  display: flex;  
  height: 200px;  
  align-items: flex-end;  
}
```



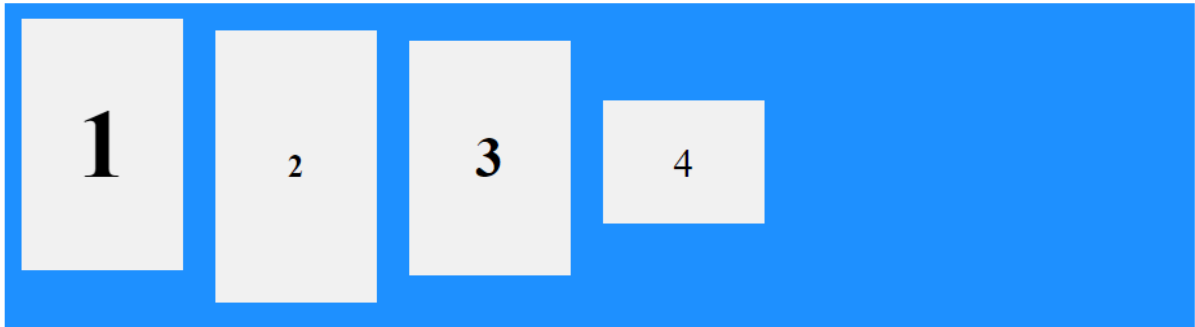
A stretch érték kinyújtja a flex elemeket a tároló kitöltésére (ez az alapértelmezett):

```
.flex-container {  
  display: flex;  
  height: 200px;  
  align-items: stretch;  
}
```



Az baseline igazítja a rugalmas elemeket, például az alpvonalakat:

```
.flex-container {  
  display: flex;  
  height: 200px;  
  align-items: baseline;  
}
```

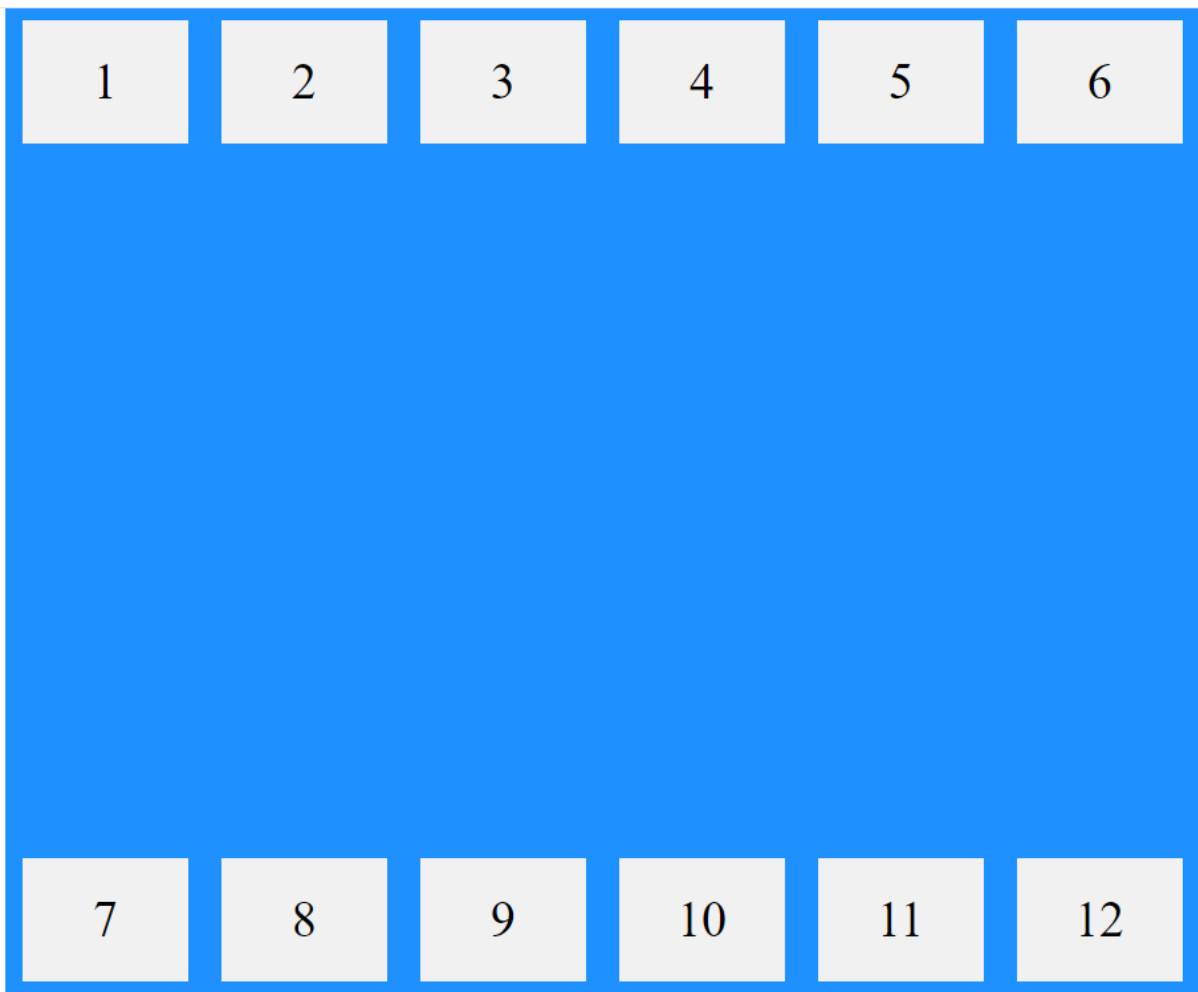


## The align-content Property

Az align-content tulajdonság a flex vonalak igazítására szolgál.

A space-between érték a flex vonalakat egyenlő térközzel jeleníti meg:

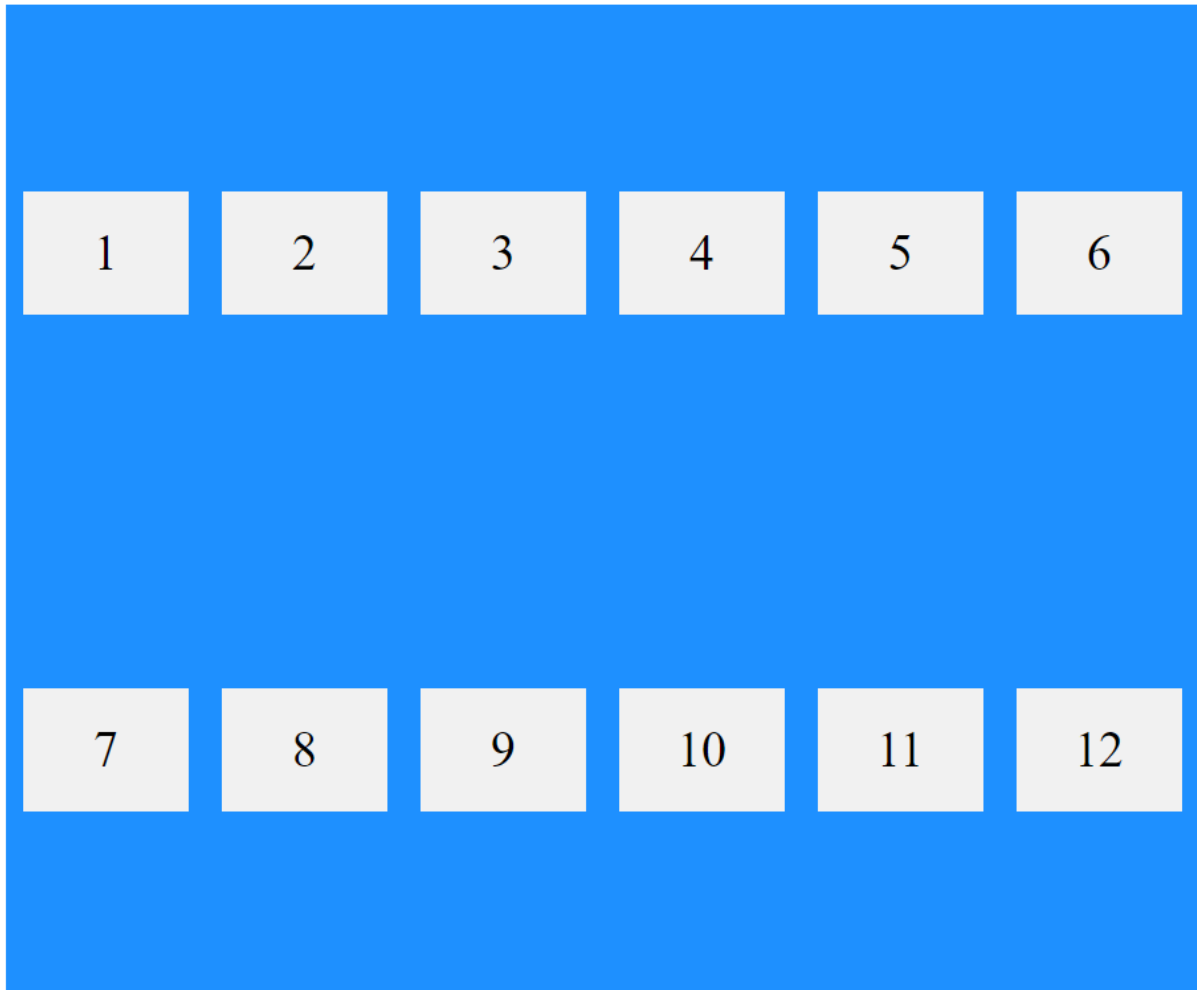
```
.flex-container {  
  display: flex;  
  height: 600px;  
  flex-wrap: wrap;  
  align-content: space-between;  
}
```





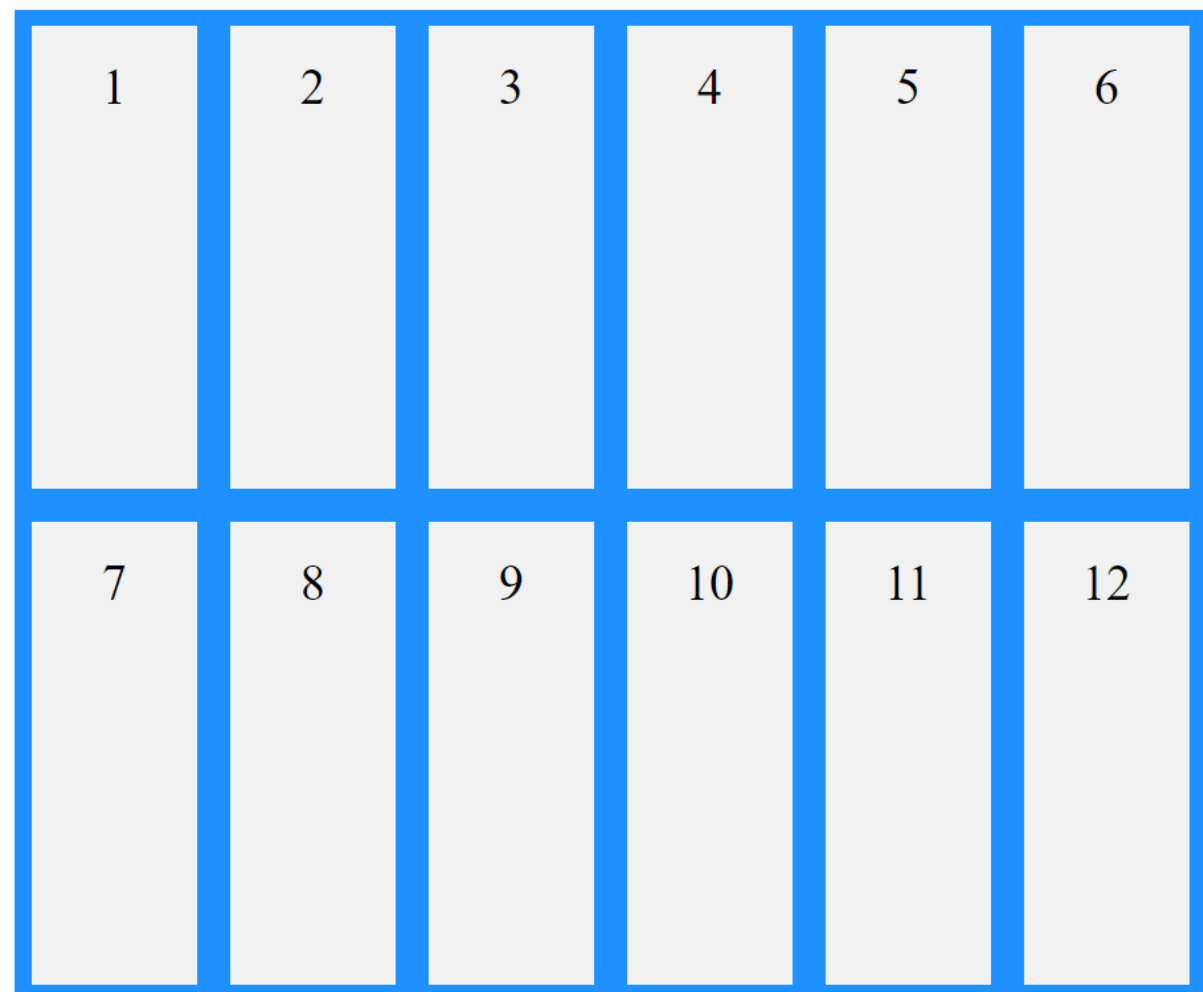
A space-around érték a flex vonalakat jeleníti meg előtte, közöttük és utánuk:

```
.flex-container {  
  display: flex;  
  height: 600px;  
  flex-wrap: wrap;  
  align-content: space-around;  
}
```



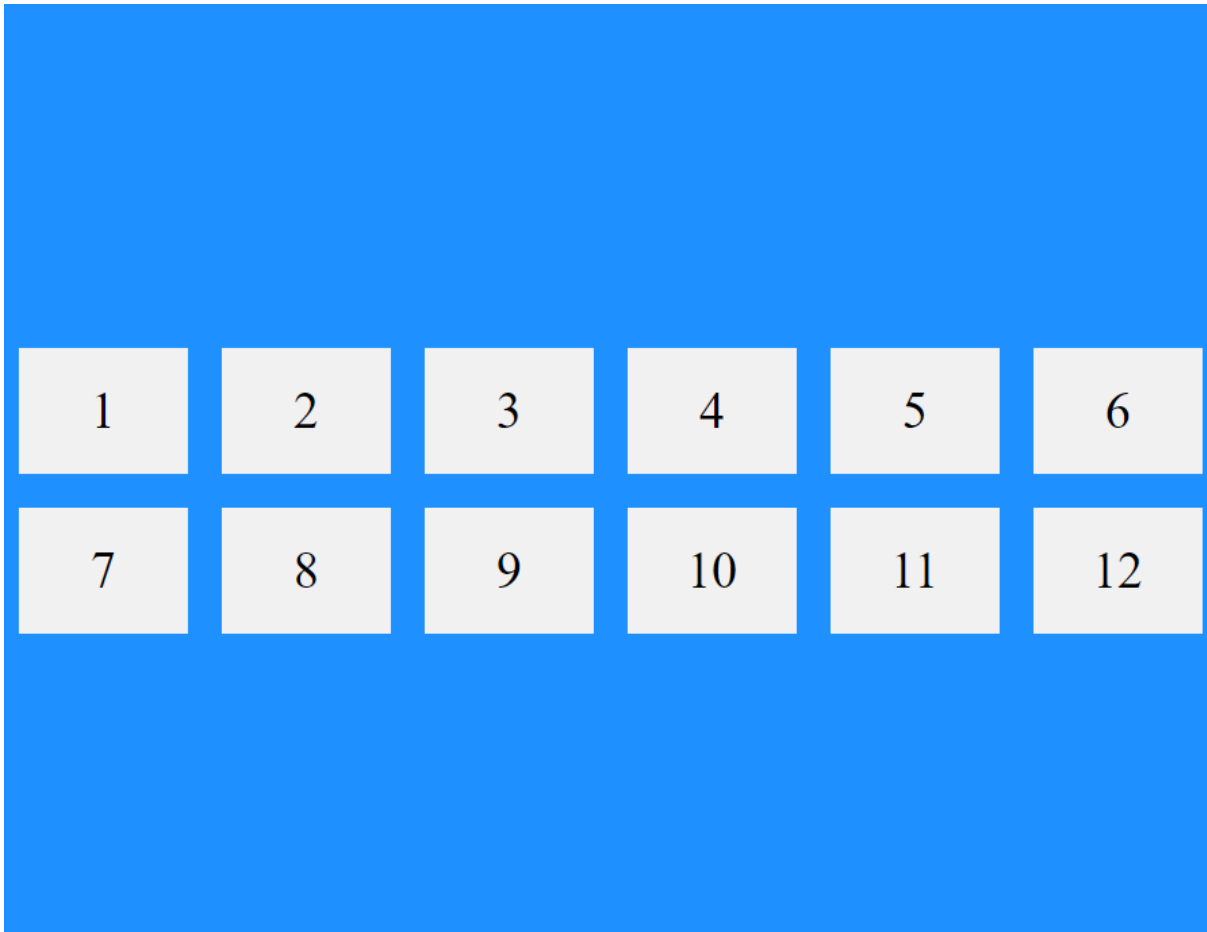
A stretch érték kinyújtja a flex vonalakat, hogy elfoglalják a fennmaradó helyet (ez az alapértelmezett):

```
.flex-container {  
  display: flex;  
  height: 600px;  
  flex-wrap: wrap;  
  align-content: stretch;  
}
```



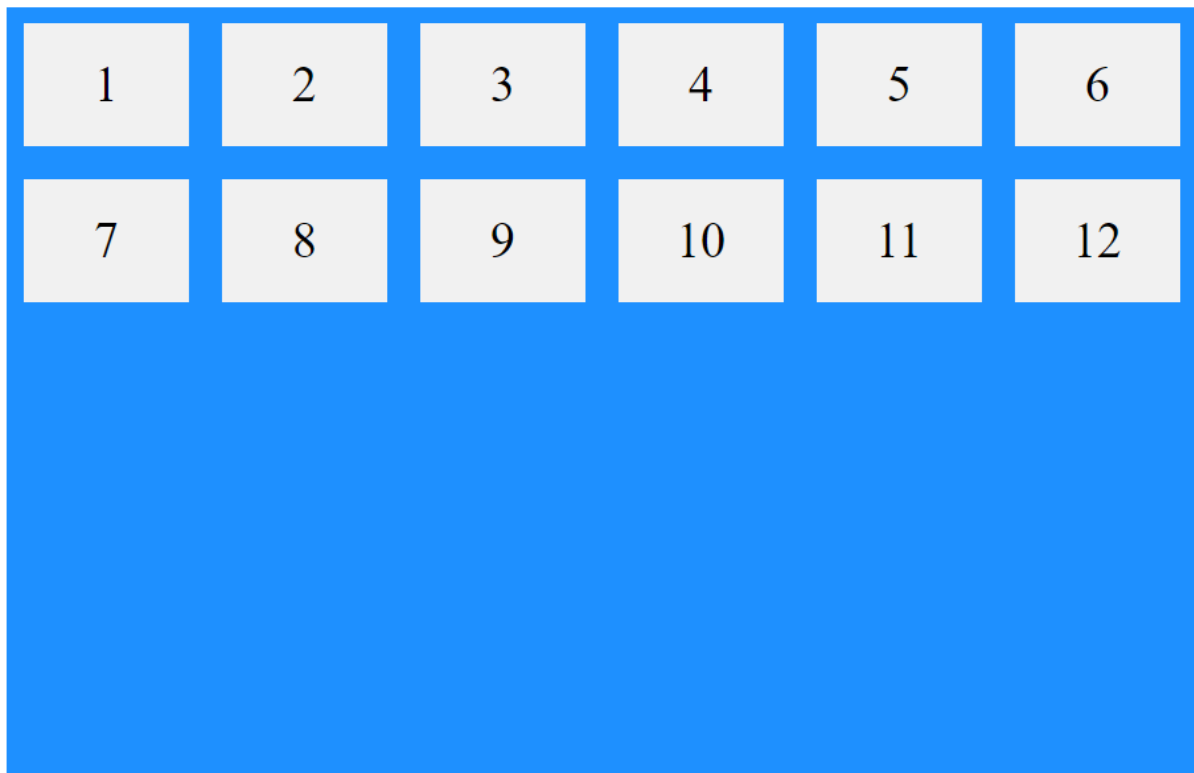
A center értéknél a flex vonalak jelennek meg a container közepén:

```
.flex-container {  
  display: flex;  
  height: 600px;  
  flex-wrap: wrap;  
  align-content: center;  
}
```



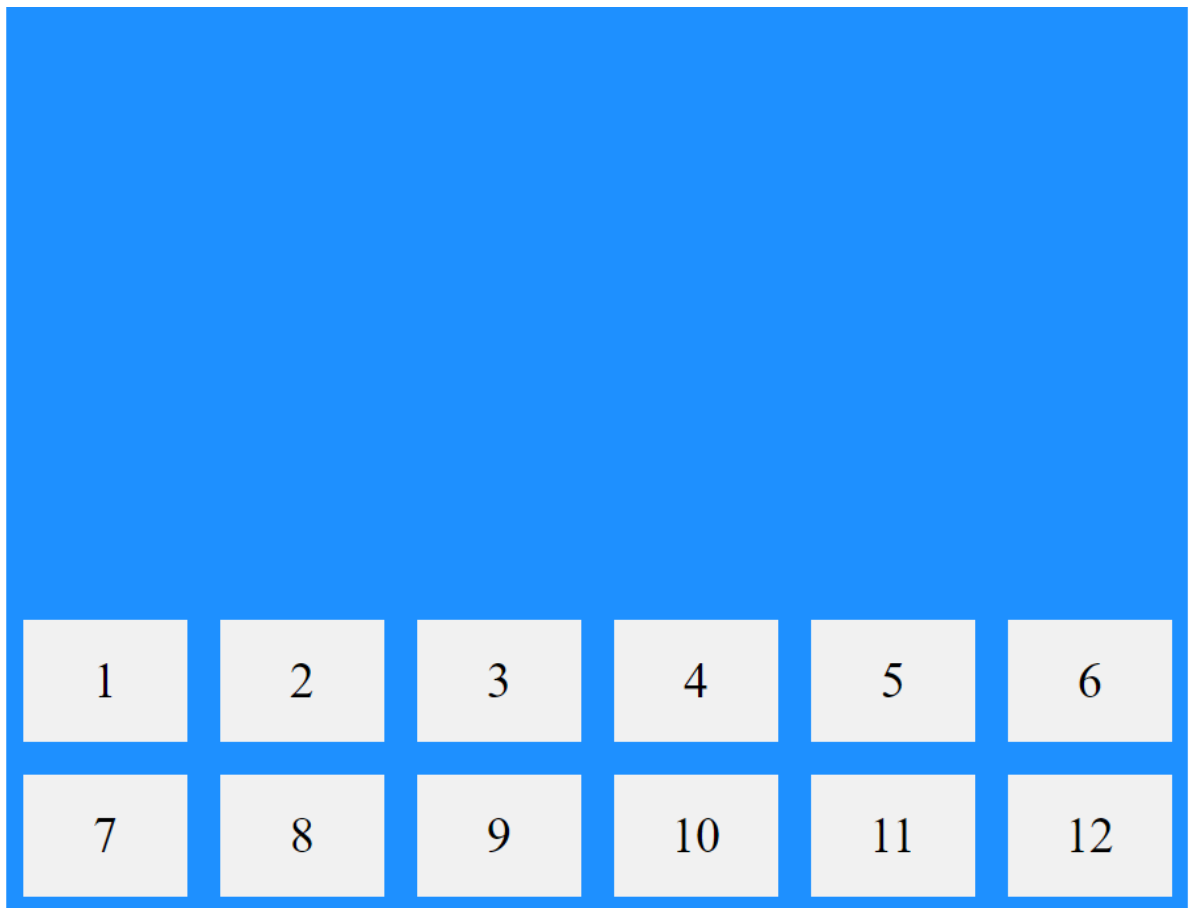
A flex-start érték a flex vonalakat jeleníti meg a container elején:

```
.flex-container {  
  display: flex;  
  height: 600px;  
  flex-wrap: wrap;  
  align-content: flex-start;  
}
```

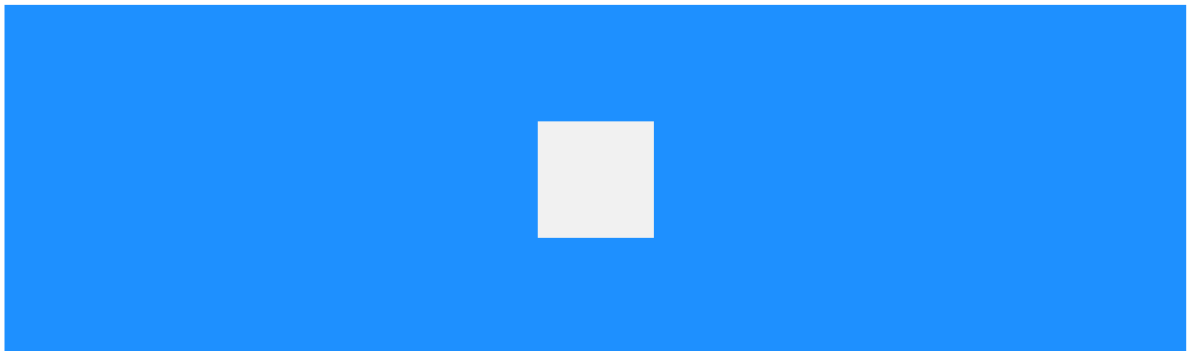


A flex-end érték megjeleníti a flex vonalakat a container végén:

```
.flex-container {  
  display: flex;  
  height: 600px;  
  flex-wrap: wrap;  
  align-content: flex-end;  
}
```



## Perfect Centering



`justify-content` and `align-items` properties to `center`, and the flex item will be perfectly centered:

```
.flex-container {  
  display: flex;  
  height: 300px;  
  justify-content: center;  
  align-items: center;  
}
```

## CSS Flex Items

## Child Elements (Items)

A flex konténer közvetlen gyermek elemei automatikusan rugalmas (flex) elemekké válnak.

```
<div class="flex-container">
  <div>1</div>
  <div>2</div>
  <div>3</div>
  <div>4</div>
</div>
```



## The order Property

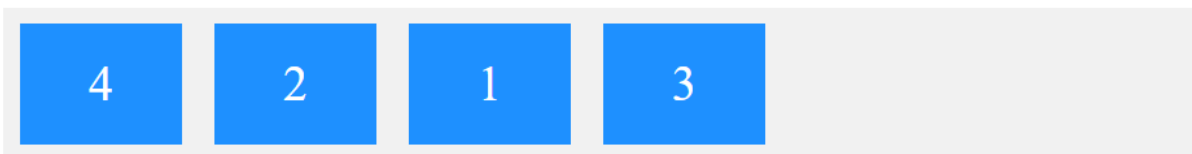
Az order tulajdonság a flex elemek sorrendjét határozza meg.

A kód első flex elemének nem kell első elemként megjelennie az elrendezésben.

Számnak kell lennie, az alapértelmezett érték 0.

Az order tulajdonság megváltoztathatja a flex elemek sorrendjét:

```
<div class="flex-container">
  <div style="order: 3">1</div>
  <div style="order: 2">2</div>
  <div style="order: 4">3</div>
  <div style="order: 1">4</div>
</div>
```



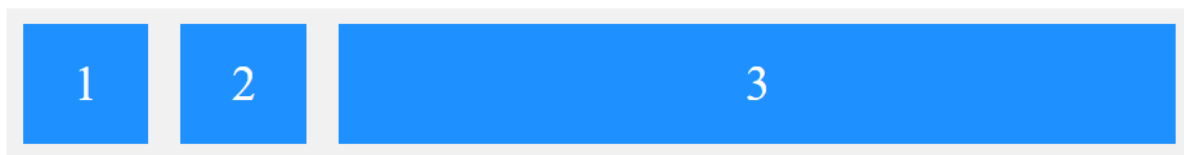
## The flex-grow Property

A flex-grow tulajdonság megadja, hogy egy flex elem mennyit fog nőni a többi flex elemhez képest.

Az értéknek számnak kell lennie, az alapértelmezett érték 0.

Hajtsa végre a harmadik flex elem nyolcszor gyorsabb növekedését, mint a többi flex elem:

```
<div class="flex-container">
  <div style="flex-grow: 1">1</div>
  <div style="flex-grow: 1">2</div>
  <div style="flex-grow: 8">3</div>
</div>
```



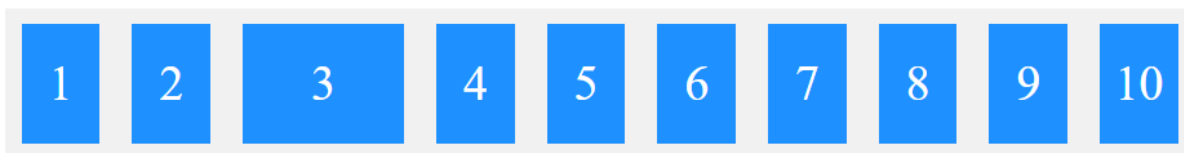
## The flex-shrink Property

A flex-shrink tulajdonság megadja, hogy a flex elem mennyit fog zsugorodni a többi flex elemhez képest.

Az értéknek számnak kell lennie, az alapértelmezett érték 1.

Ne hagyja, hogy a harmadik flex elem annyira összezsugorodjon, mint a többi flex elem:

```
<div class="flex-container">
  <div>1</div>
  <div>2</div>
  <div style="flex-shrink: 0">3</div>
  <div>4</div>
  <div>5</div>
  <div>6</div>
  <div>7</div>
  <div>8</div>
  <div>9</div>
  <div>10</div>
</div>
```

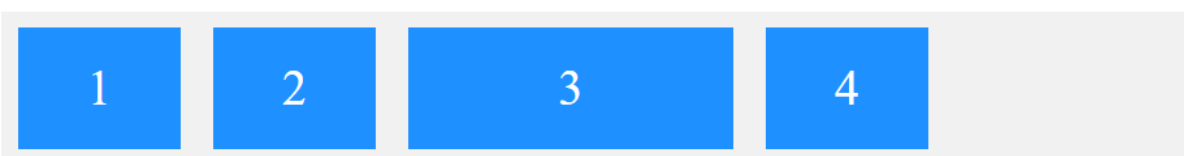


## The flex-basis Property

A flex-basis tulajdonság megadja a flex elem kezdeti hosszát.

Állítsa be a harmadik flex elem kezdeti hosszát 200 képpontra:

```
<div class="flex-container">
  <div>1</div>
  <div>2</div>
  <div style="flex-basis: 200px">3</div>
  <div>4</div>
</div>
```

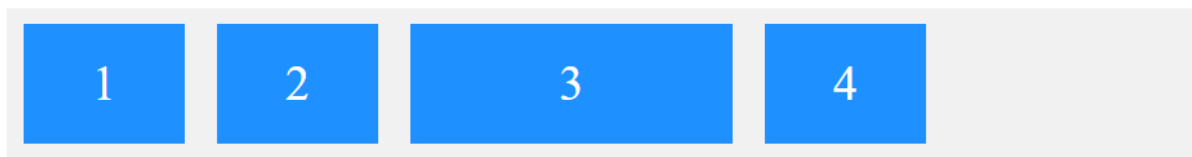


# The flex Property

A flex tulajdonság a flex-grow, flex-shrink és flex-based tulajdonságok rövidített tulajdonsága.

Tegye a harmadik flex elemet nem növeszthetővé (0), nem zsugoríthatóvá (0) és 200 pixel kezdeti hosszúságúvá:

```
<div class="flex-container">
  <div>1</div>
  <div>2</div>
  <div style="flex: 0 0 200px">3</div>
  <div>4</div>
</div>
```



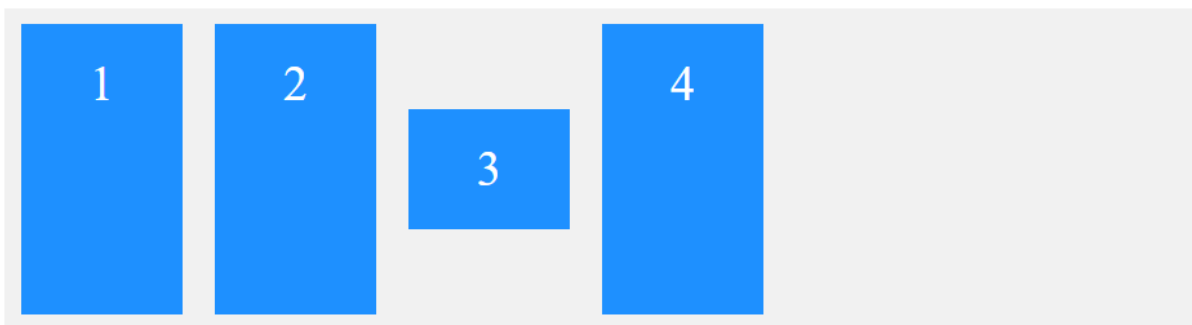
# The align-self Property

Az align-self tulajdonság megadja a kijelölt elem igazítását a flex tárolóban.

Az align-self tulajdonság felülírja a tároló align-items tulajdonságának alapértelmezett igazítását.

Igazítsa a harmadik flex elemet a container közepére:

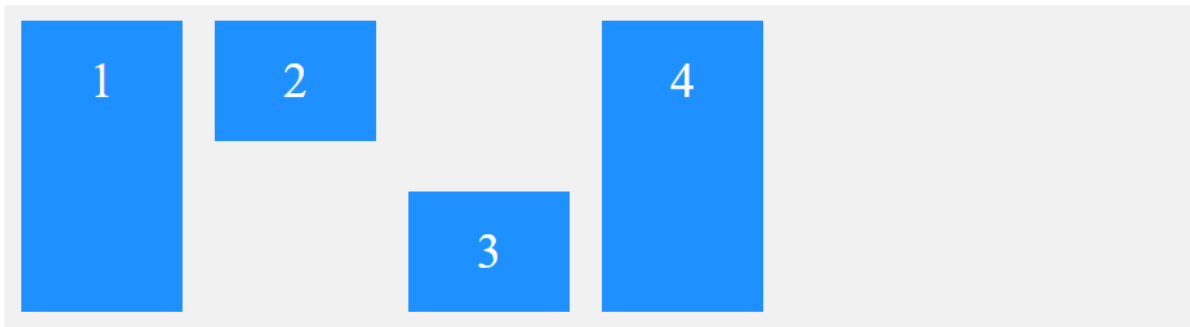
```
<div class="flex-container">
  <div>1</div>
  <div>2</div>
  <div style="align-self: center">3</div>
  <div>4</div>
</div>
```



Igazítsa a második flex elemet a tartály tetejére, a harmadik flex elemet pedig a container aljára:

```
<div class="flex-container">
  <div>1</div>
  <div style="align-self: flex-start">2</div>
  <div style="align-self: flex-end">3</div>
  <div>4</div>
</div>
```





# CSS Flex Responsive

## Responsive Flexbox

A Média lekérdezések fejezetből megtudta, hogy média lekérdezésekkel különböző elrendezéseket hozhat létre különböző képernyőméretekhez és eszközökhöz.

Laptop and Desktops:	Mobile phones and Tablets:
1	1
2	2
3	3

Például, ha két oszlopos elrendezést szeretne létrehozni a legtöbb képernyőmérethez, és egy oszlopos elrendezést a kis képernyőméretekhez (például telefonokhoz és táblagépekhez), akkor a flex irányt sorról oszlopra módosíthatja egy adott töréspont (800px az alábbi példában):

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

/* Responsive layout - makes a one column layout instead of a two-column layout */
@media (max-width: 800px) {
  .flex-container {
    flex-direction: column;
  }
}
```

Egy másik módszer az, hogy megváltoztatjuk a flex elemek rugalmas tulajdonságának százalékos arányát, hogy különböző elrendezéseket hozzunk létre a különböző képernyőméretekhez. Ne feledje, hogy a `flex-wrap: wrap;` property-t használni a flex tárolón, hogy ez a példa működjön:

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

```
.flex-item-left {  
  flex: 50%;  
}  
  
.flex-item-right {  
  flex: 50%;  
}  
  
/* Responsive layout - makes a one column layout instead of a two-column layout */  
@media (max-width: 800px) {  
  .flex-item-right, .flex-item-left {  
    flex: 100%;  
  }  
}
```