

# Bloklar

---

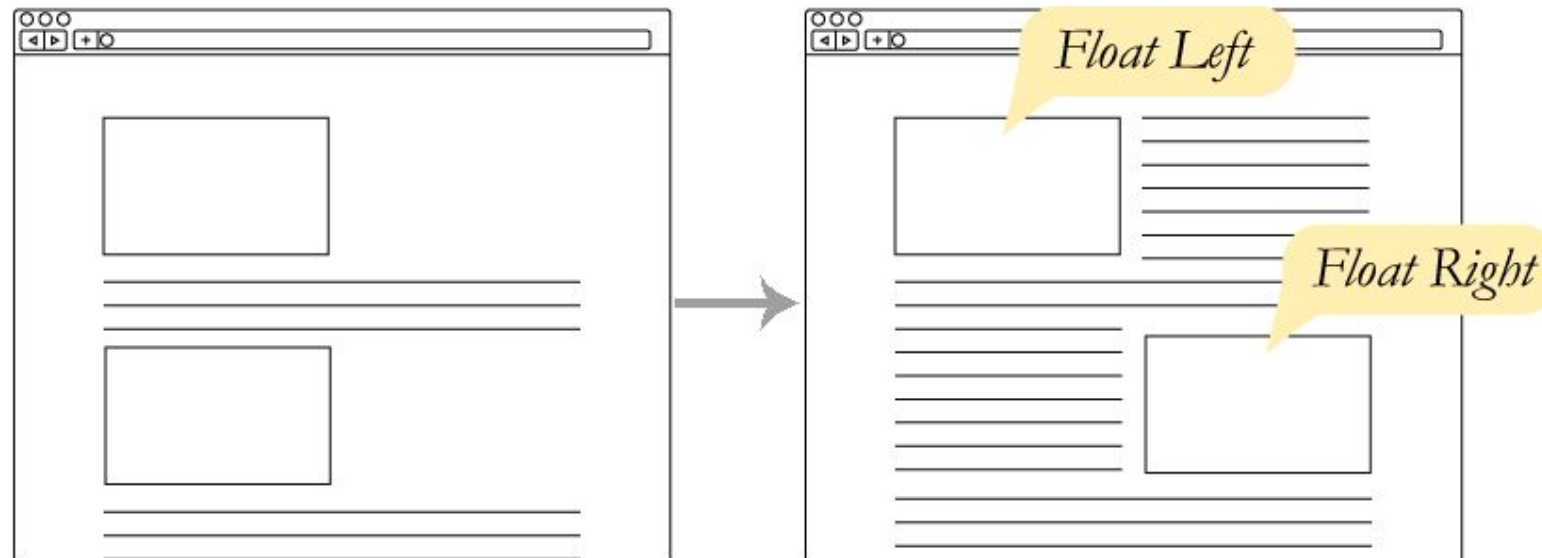
# Suzuvchi elementlar (Float)

---

## float

Qiymatlari: left | right | none

- kenglik (width) aniqlanishi kerak
- float qo'llanilgan inline elementlar block elementlarga aylanadi
- element joylashuvi bo'yicha teparoqqa chiqib ketmaydi
- float qo'llanilmagan elementlar oddiy oqimda (flow) taqdim etiladi



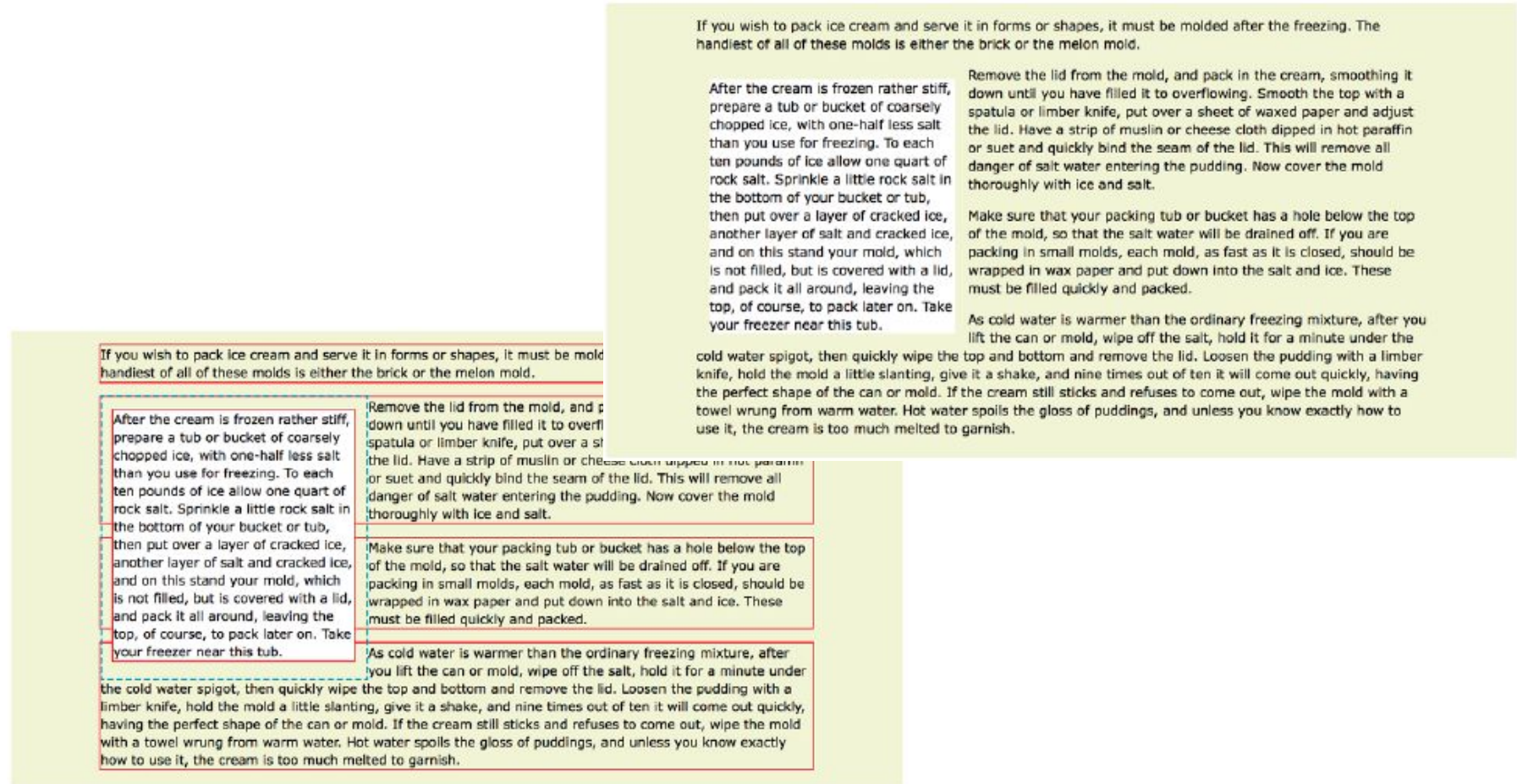
# Suzuvchi elementlar (Float)

```
p {  
  border: 2px red solid;  
}
```

```
#float {  
  float: left;  
  width: 300px;  
  margin: 1em;  
  background: white;  
}
```

...

```
<p>If you ...</p>  
<p id="float">After ...</p>  
<p>Remove...</p>  
<p>Make sure...</p>  
<p>As cold ...</p>
```



# Suzuvchi elementlarni tozalash

---

`clear` (faqatgina blokli elementlarga qo'llaniladi)

Qiymatlari: `left` | `right` | `both` | `none`

```
img {  
  float: left;  
  margin-right: .5em;  
}  
  
h2 {  
  clear: left;  
  margin-top: 2em;  
}
```



If pure raw cream is stirred rapidly, it swells and becomes frothy, like the beaten whites of eggs, and is "whipped cream." To prevent this in making Philadelphia Ice Cream, one-half the cream is scalded, and when it is very cold, the remaining half of raw cream is added. This gives the smooth, light and rich consistency which makes these creams so different from others.

## USE OF FRUITS

Use fresh fruits in the summer and the best canned unsweetened fruits in the winter. If sweetened fruits must be used, cut down the given quantity of sugar. Where acid fruits are used, they should be added to the cream after it is partly frozen.

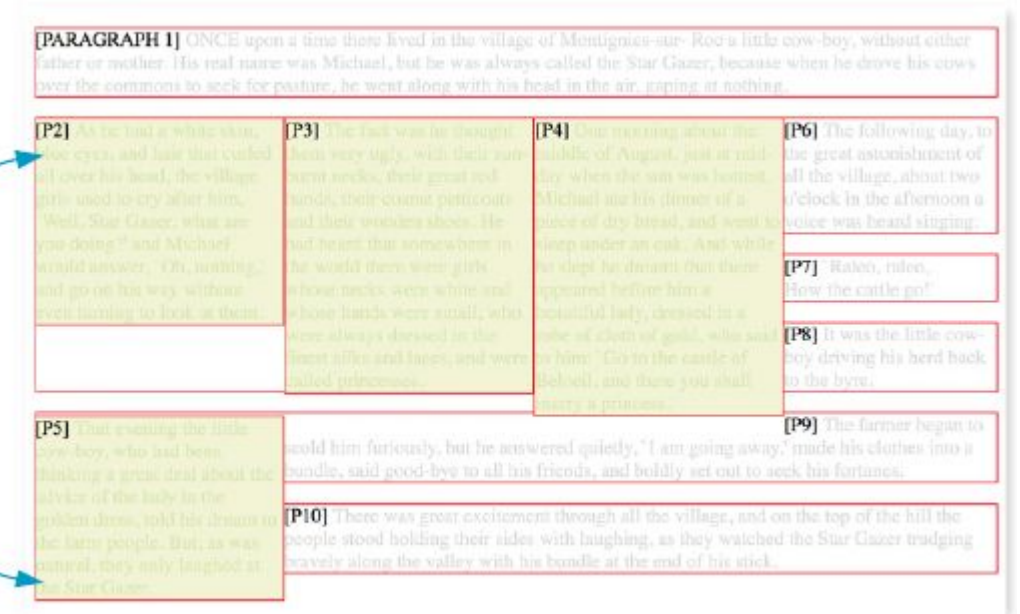
The time for freezing varies according to the quality of cream or milk or water; water ices require a longer time than ice creams. It is not well to freeze the mixtures too rapidly; they are apt to be coarse, not smooth, and if they are churned before the mixture is icy cold they will be greasy or "buttery."

# Bir nechta suzuvchi elementlar

```
<p>[PARAGRAPH 1]</p>
<p class="float">[P2]...</p>
<p class="float">[P3]...</p>
<p class="float">[P4]...</p>
<p class="float">[P5]...</p>
<p>[P6]...</p>
<p>[P7]...</p>
<p>[P8]...</p>
<p>[P9]...</p>
<p>[P10]...</p>
...
p.float {
  float: left;
  width: 200px;
  margin: 0px;
  background: #F2F5d5;
  color: #DAEAB1;
}
```

Elementlar bir qatorda chap tomondan joylashgan

Ammo joy yetarli bo'lmaganda elementlar pastdan va chapdan joylashadi



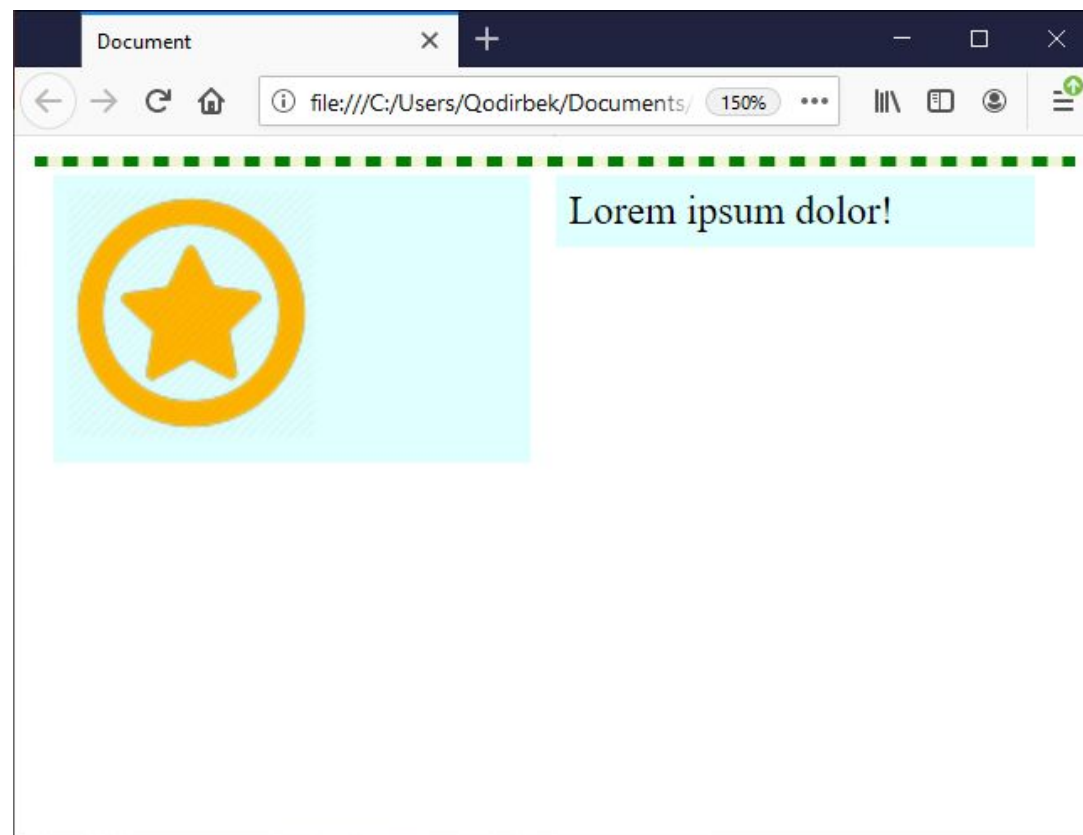
# Bir nechta suzuvchi elementlar

```
<div id="header">
  <div class="logo"></div>
  <div class="title">Lorem ipsum dolor!</div>
</div>
```

...

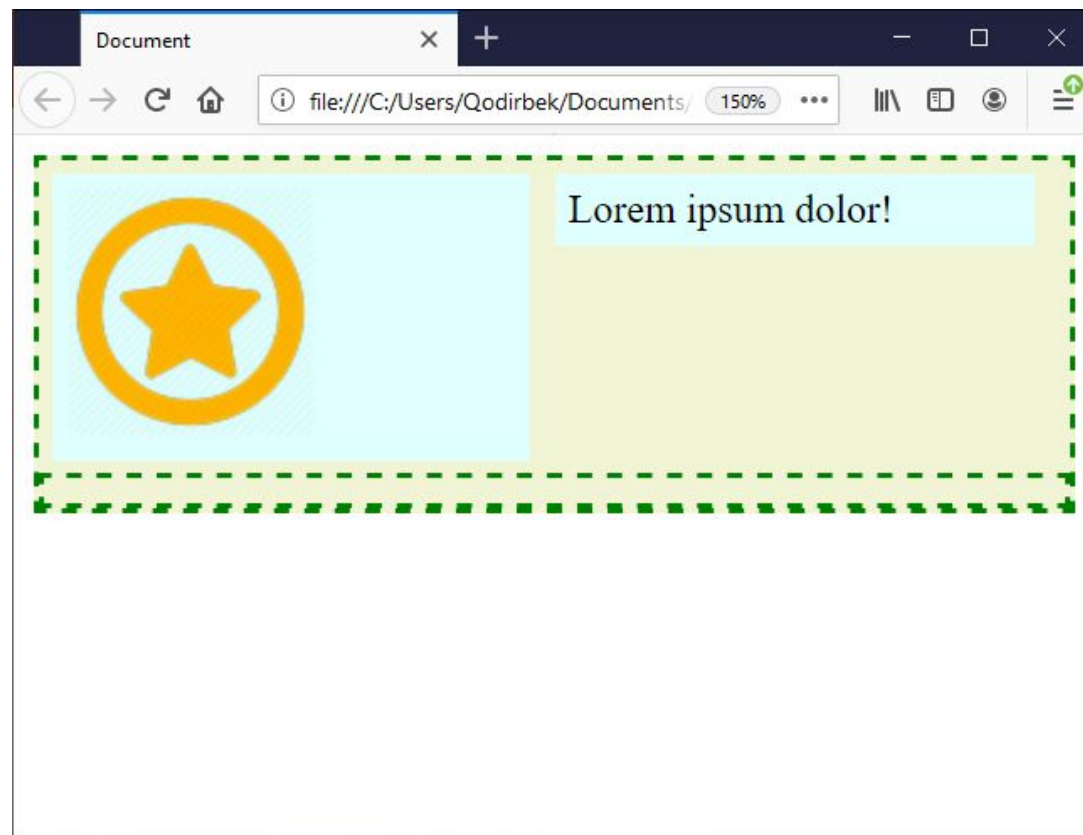
```
#header {
  background: #f2f5d5;
  border: 2px dashed green;
}
```

```
#header>div {
  float: left;
  width: 44%;
  padding: 5px;
  background-color: lightcyan;
  margin: 5px;
}
```



# Bir nechta suzuvchi elementlar

```
<div id="header">
  <div class="logo"></div>
  <div class="title">Lorem ipsum dolor!</div>
</div>
...
#header {
  ...
}
#header>div {
  ...
}
#header:after {
  content: " ";
  display: block;
  clear: both;
  background-color: #f2f5d5;
  border: 2px dashed green;
  height: 10px;
}
```



# Joylashuv

---

## position

Qiymatlari: static | relative | absolute | fixed | sticky

## top, right, bottom, left

Qiymatlari: *uzunlik* | *foiz* | auto

static – oddiy joylashuv sxemasi, elementlar oqimga qanday kelsa shunda joylashadi.

relative – nisbiy joylashuv, elementni oqimdagi dastlabki joylashuviga nisbatan harakatlantiradi, bunda elementning oddiy oqimdagi joylashuvdagi maydoni bo'sh maydonligicha qoldiriladi.

absolute – absolyut joylashtirilgan elementlar oqimdan olib to'liq olib tashlanadi va ko'rinish sohasi yoki o'zi joylashgan elementga nisbatan joylashadi.

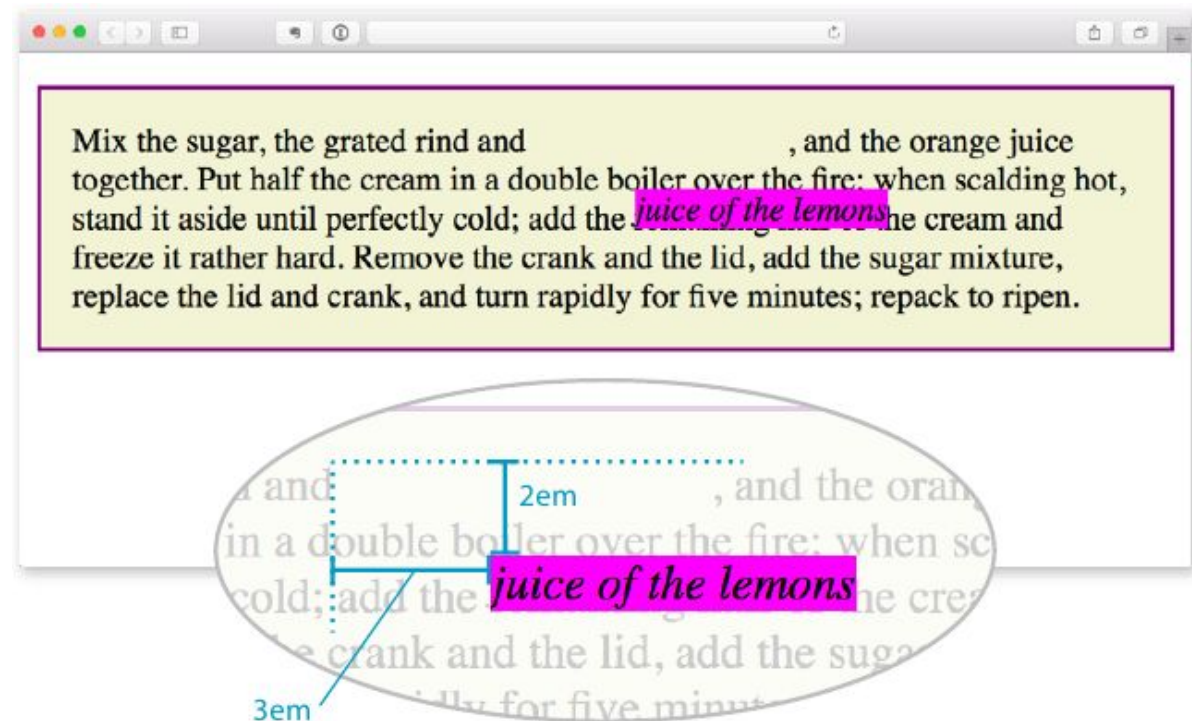
fixed – element ko'rinish sohasiga (viewport) nisbatan joylashtiriladi va scrolling bo'lganida ham bitta joylashuvida qoladi.

sticky – foydalanuvchining scroll sohasiga nisbatan joylashadi, ya'ni scroll sohasi kelgunicha relative qiymatidagi kabi, scroll sohasi o'tib ketganda esa fixed qiymatidagi kabi ishlaydi.



# Nisbiy joylashuv (relative)

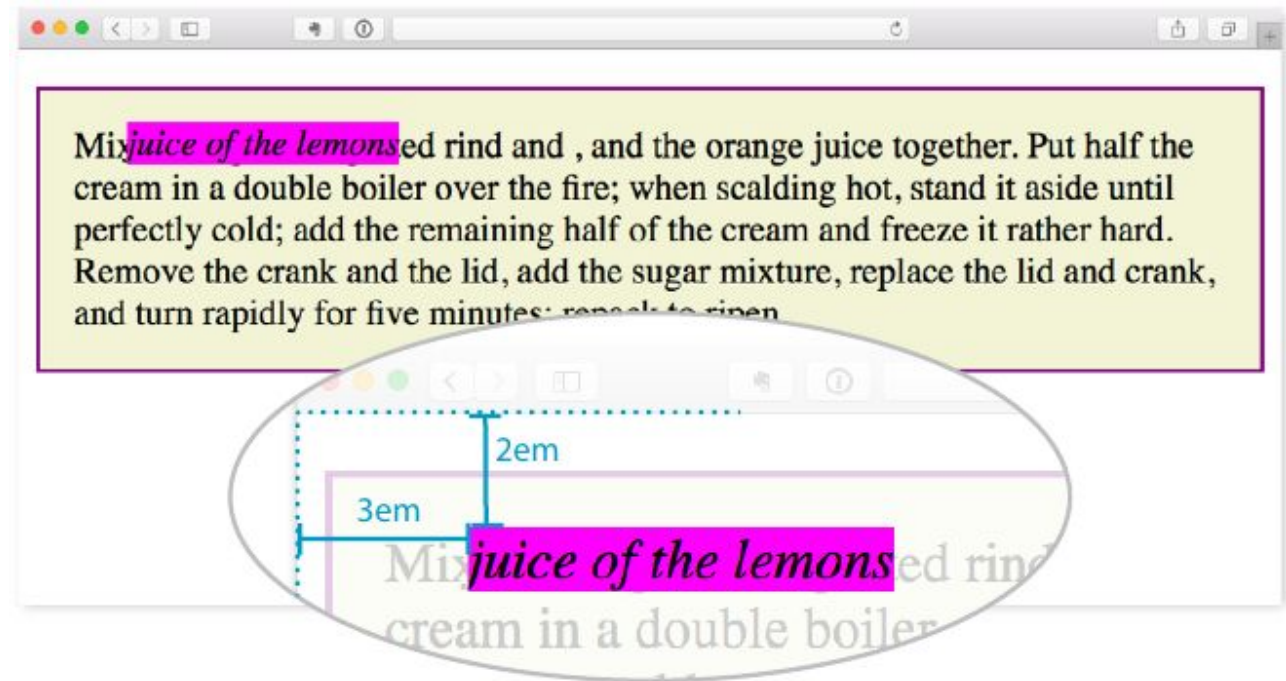
```
em {  
  position: relative;  
  top: 2em;  
  left: 3em;  
  background-color: fuchsia;  
}
```



# Absolyut joylashuv (absolute)

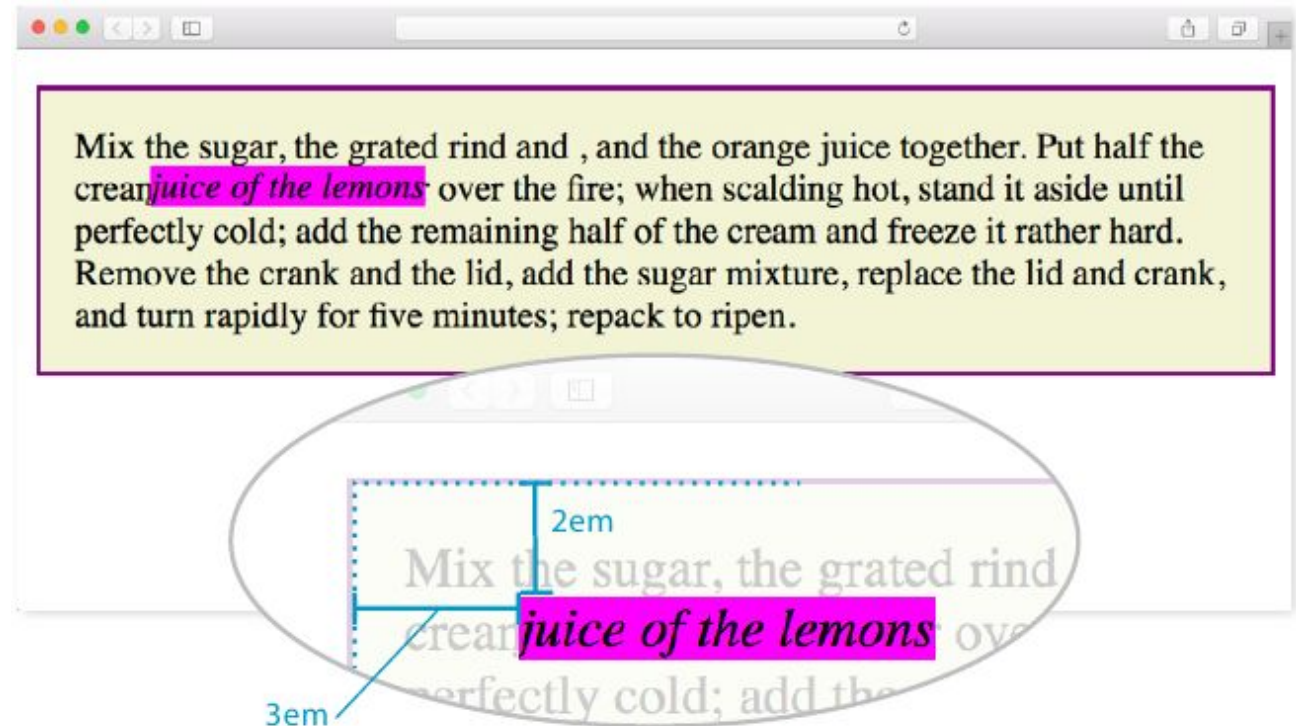
```
em {  
  position: absolute;  
  top: 2em;  
  left: 3em;  
  background-color: fuchsia;  
}
```

\* Absolyut element ierarxiya bo'yicha joylashuv o'rnatilgan elementga nisbatan absolyut joylashadi, agar joylashuv o'rnatilmagan elementlar mavjud bo'lmasa html elementiga nisbatan joylashadi.



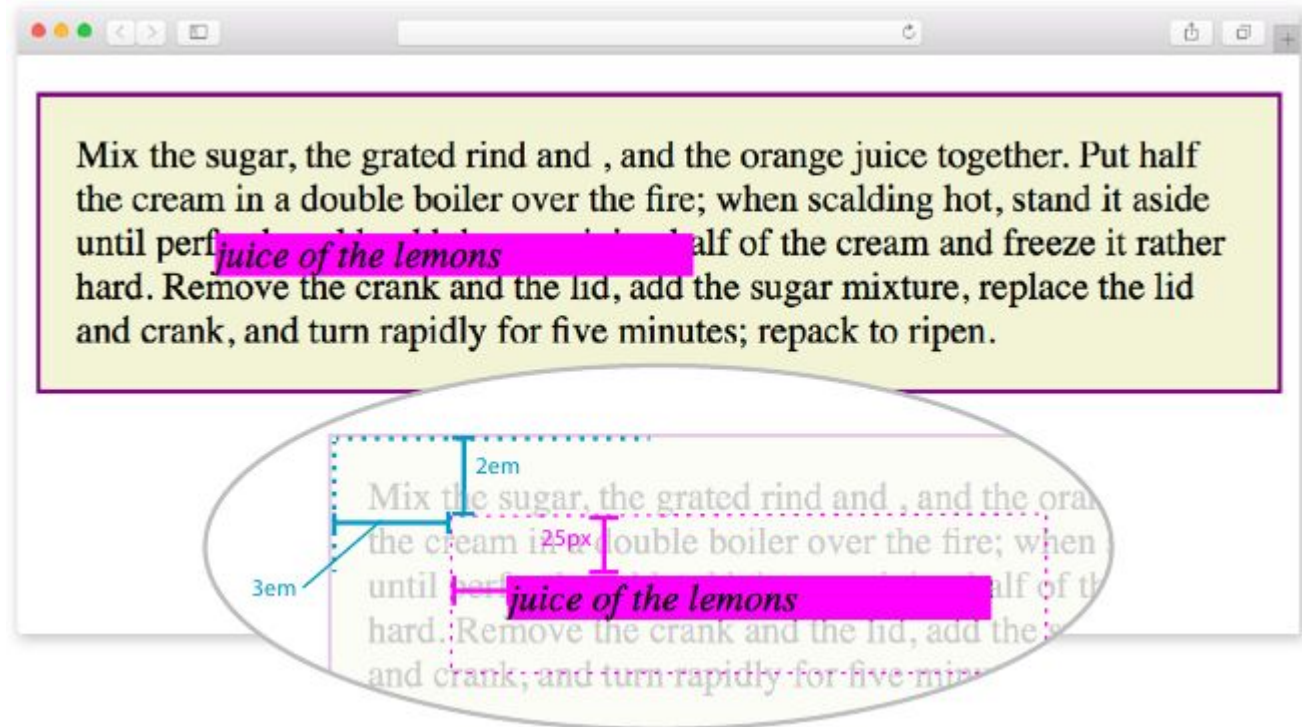
# Absolyut joylashuv (absolute)

```
p {  
  position: relative;  
  padding: 15px;  
  background-color: #F2F5D5;  
  border: 2px solid purple;  
}  
  
em {  
  position: absolute;  
  top: 2em;  
  left: 3em;  
  background-color: fuchsia;  
}
```



# Absolyut joylashuv (absolute)

```
em {  
  width: 200px;  
  margin: 25px;  
  position: absolute;  
  top: 2em;  
  left: 3em;  
  background-color: fuchsia;  
}
```

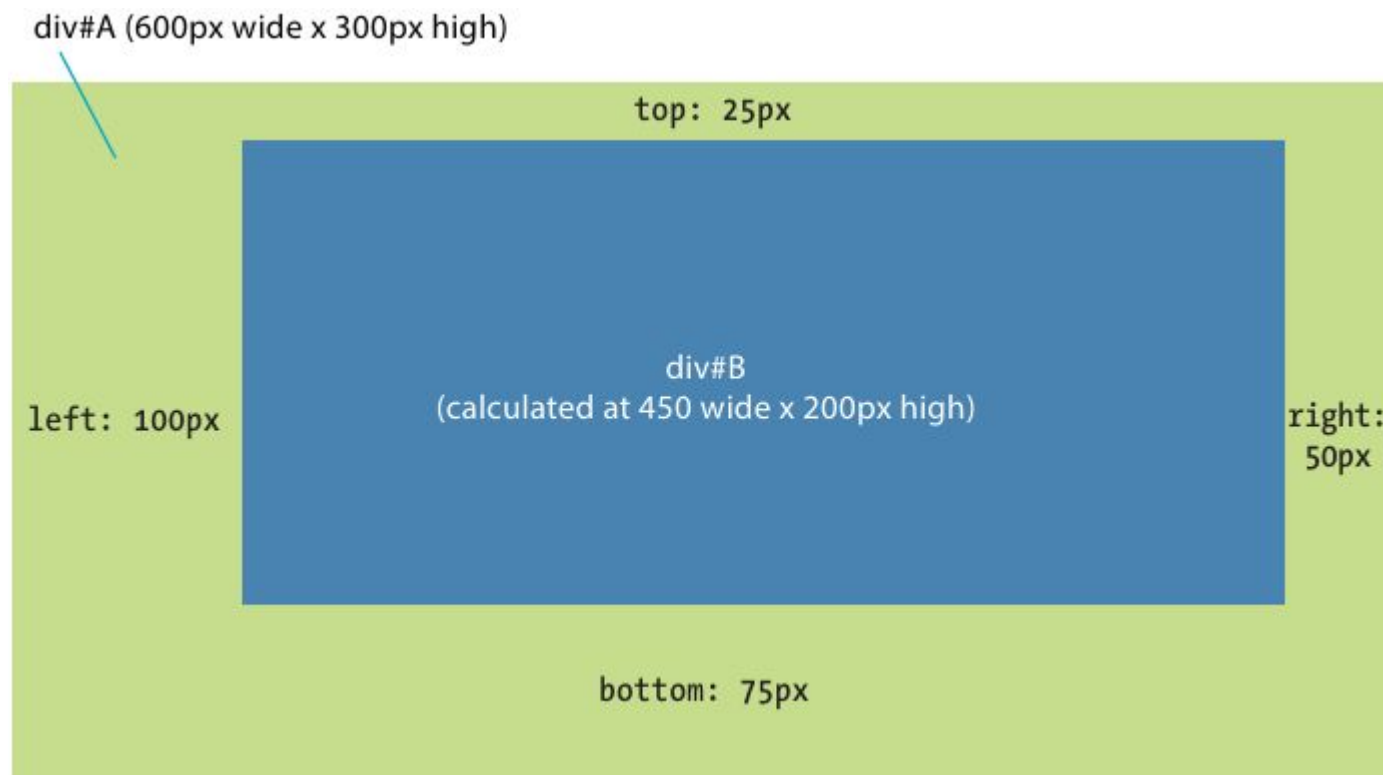


# Joylashuv va o'lcham

```
<div id="A">
  <div id="B">&nbsp;</div>
</div>

...

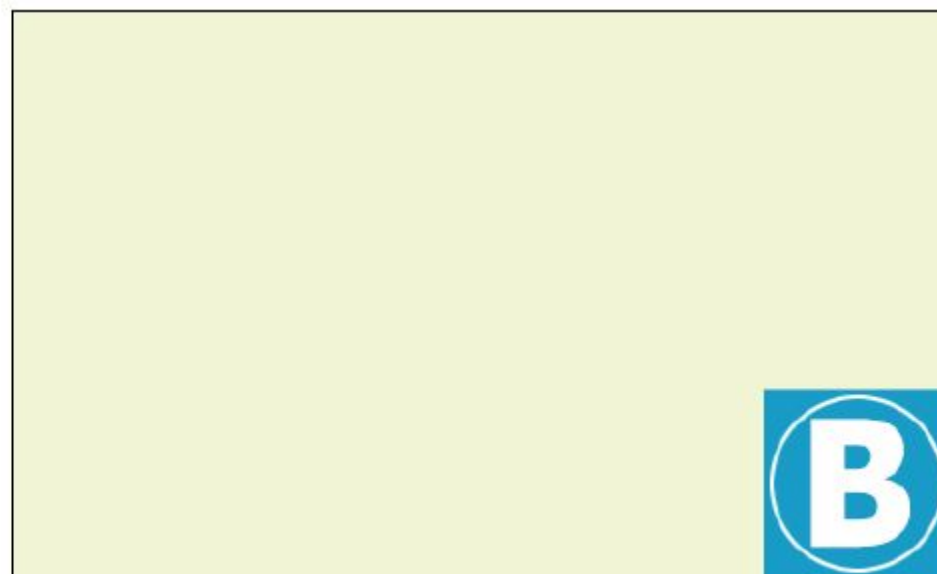
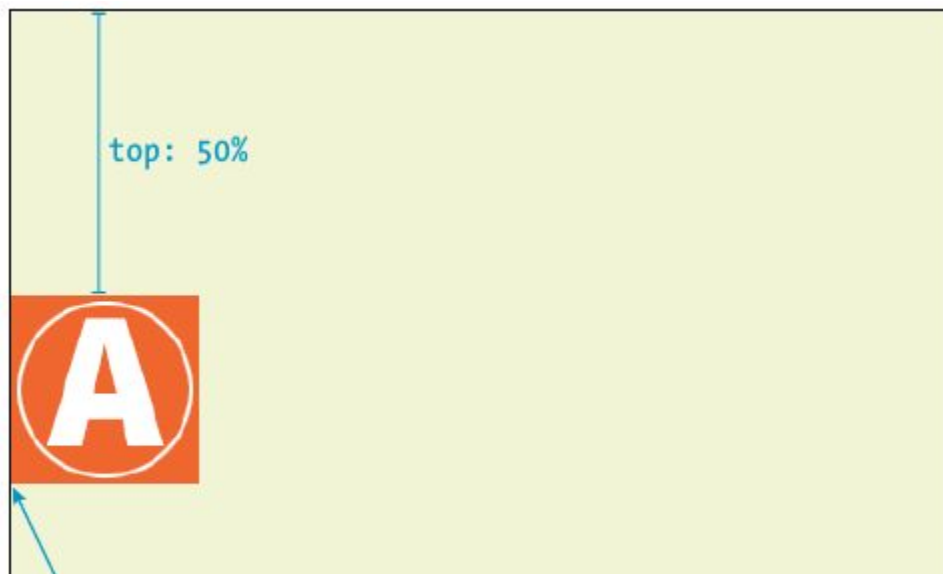
div#A {
  position: relative;
  width: 600px;
  height: 300px;
  background-color: #C6DE89;
}
div#B {
  position: absolute;
  top: 25px;
  right: 50px;
  bottom: 75px;
  left: 100px;
  background-color: steelblue;
}
```



# Joylashuv

---

```
img#A {  
  position: absolute;  
  top: 50%;  
  left: 0%;  
}
```

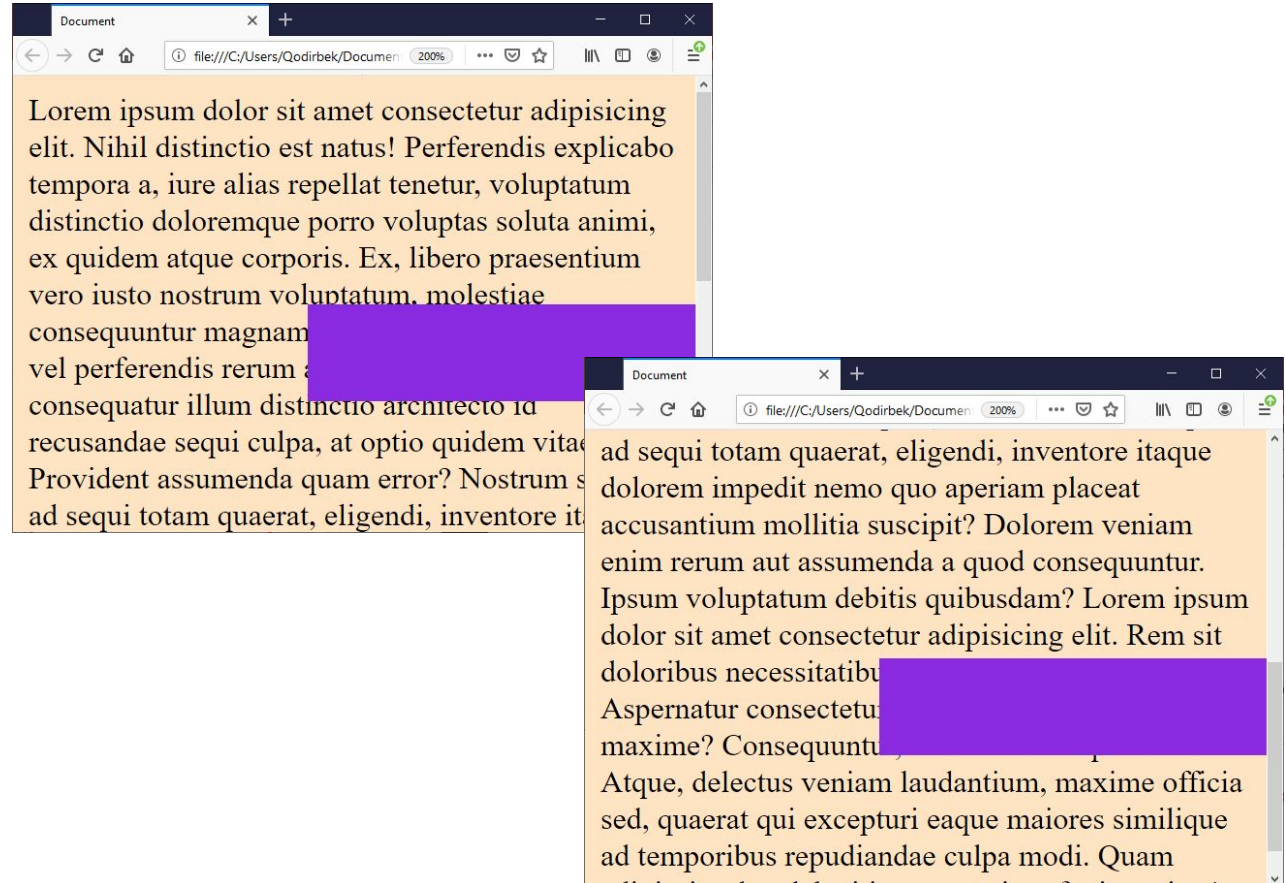


left: 0%

bottom: 0%; right: 0%

# Joylashuv - fixed

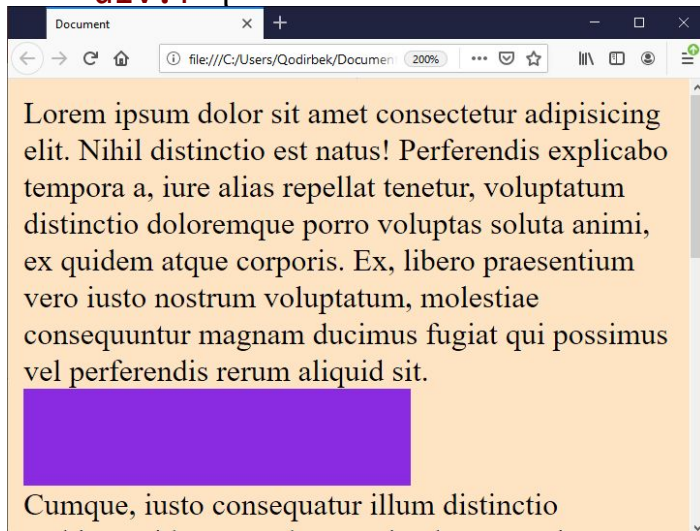
```
body {  
    background-color: bisque;  
}  
  
div.f {  
    position: fixed;  
    top: 50%;  
    right: 0;  
    background-color: blueviolet;  
    width: 200px;  
    height: 50px;  
}
```



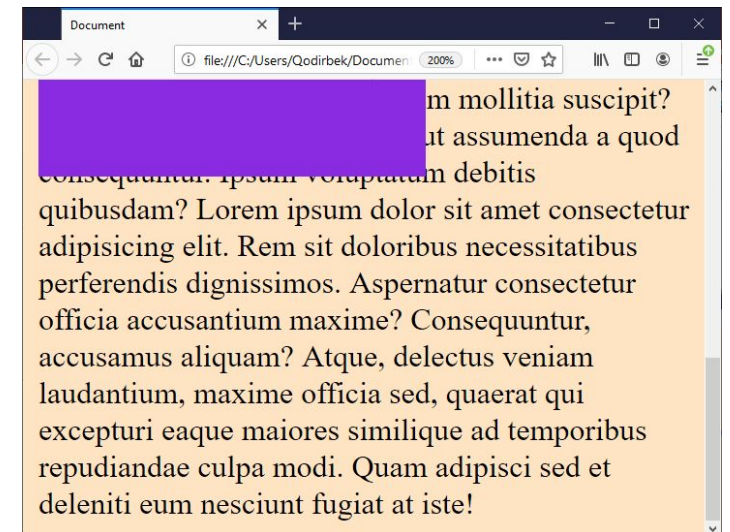
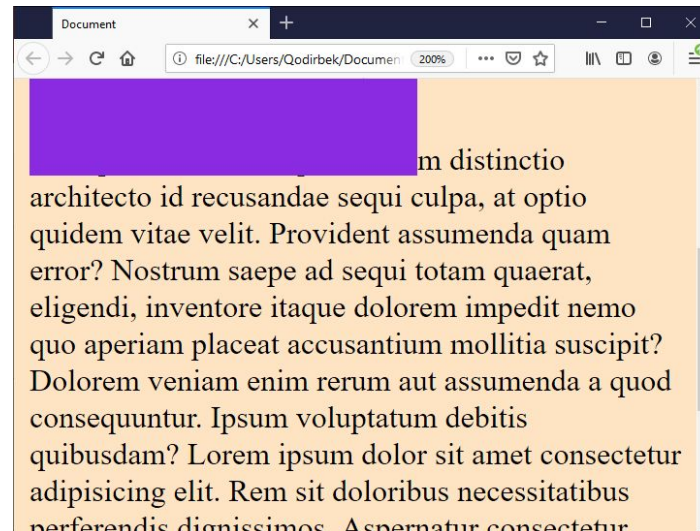
# Joylashuv - sticky

```
body {  
    background-color: bisque;  
}
```

```
div.f {
```



```
et;
```





# Elementlar tartibi

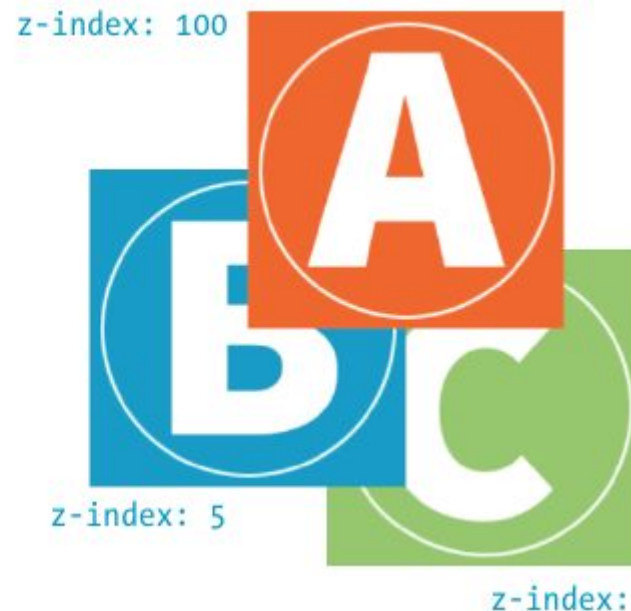
## z-index

Qiymatlari: *son* | *auto*

```
<p id="A"></p>
<p id="B"></p>
<p id="C"></p>
```

...

```
#A {
  position: absolute;
  top: 175px;
  left: 200px;
}
#B {
  position: absolute;
  top: 275px;
  left: 100px;
}
#C {
  position: absolute;
  top: 325px;
  left: 250px;
}
```



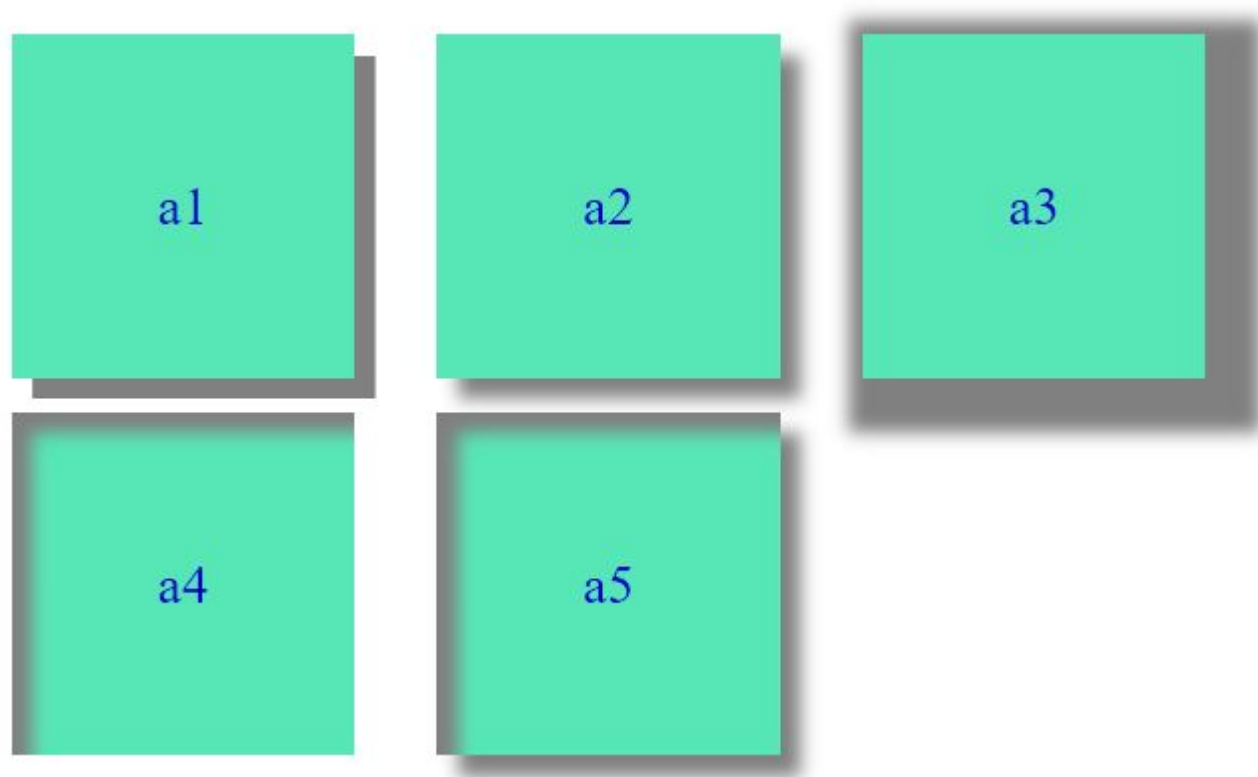
```
#A {
  z-index: 100;
  position: absolute;
  top: 175px;
  left: 200px;
}
#B {
  z-index: 5;
  position: absolute;
  top: 275px;
  left: 100px;
}
#C {
  z-index: 1;
  position: absolute;
  top: 325px;
  left: 250px;
}
```

# Soya

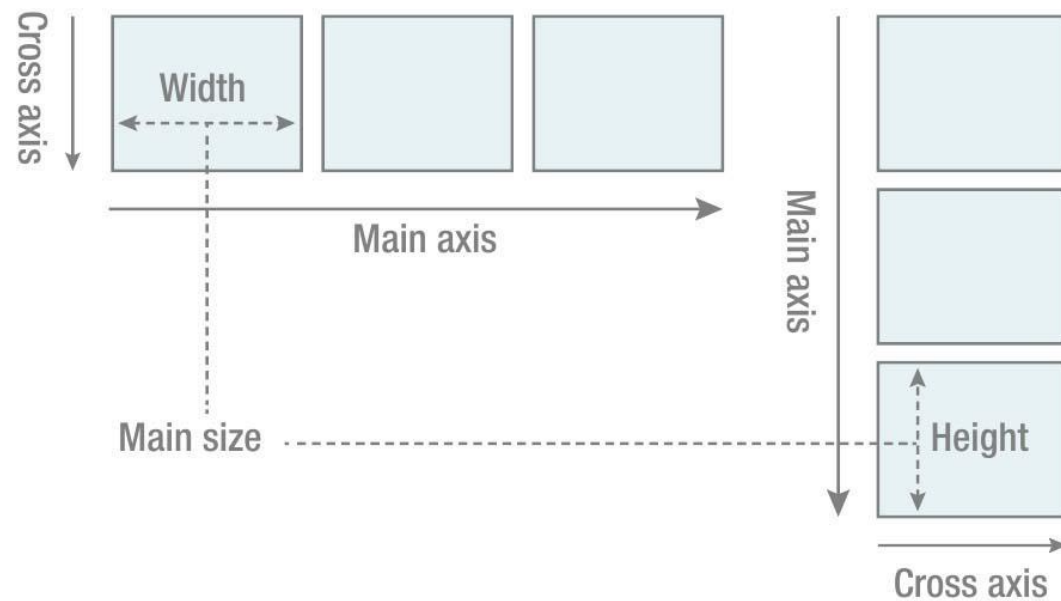
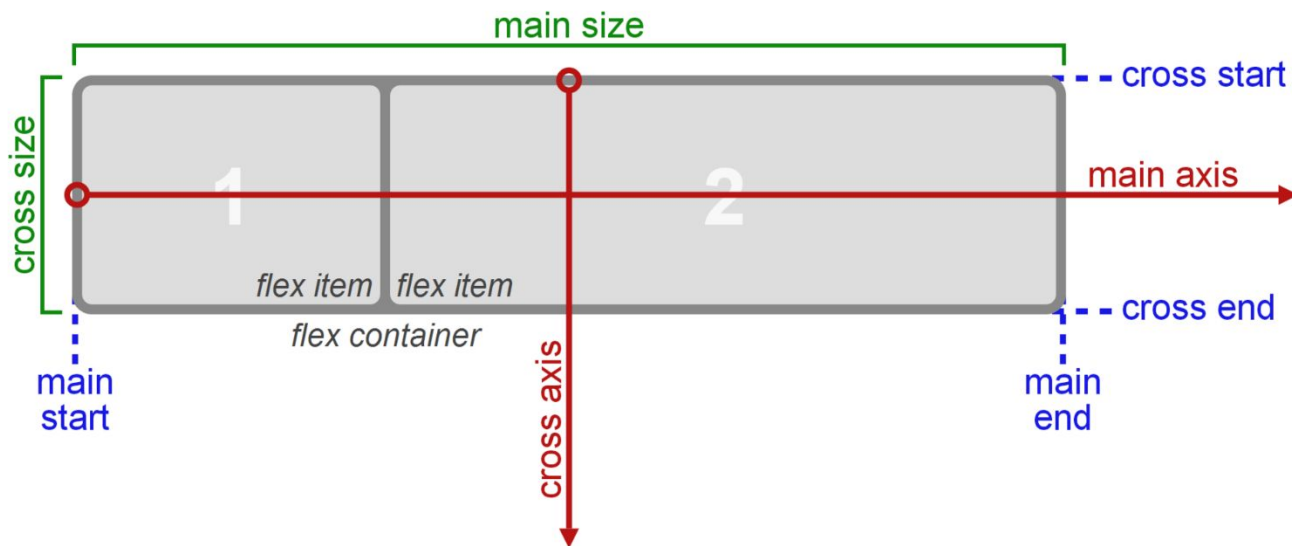
## box-shadow

Qiymatlari: none | *h-offset* *v-offset* [*blur*] [*spread*] [*color*] [*inset*];

```
.a1 {  
  box-shadow: 6px 6px gray;  
}  
.a2 {  
  box-shadow: 6px 6px 5px gray;  
}  
.a3 {  
  box-shadow: 6px 6px 5px 10px gray;  
}  
.a4 {  
  box-shadow: 6px 6px 5px gray inset;  
}  
.a5 {  
  box-shadow: 6px 6px 5px gray inset,  
             6px 6px 5px gray;  
}
```

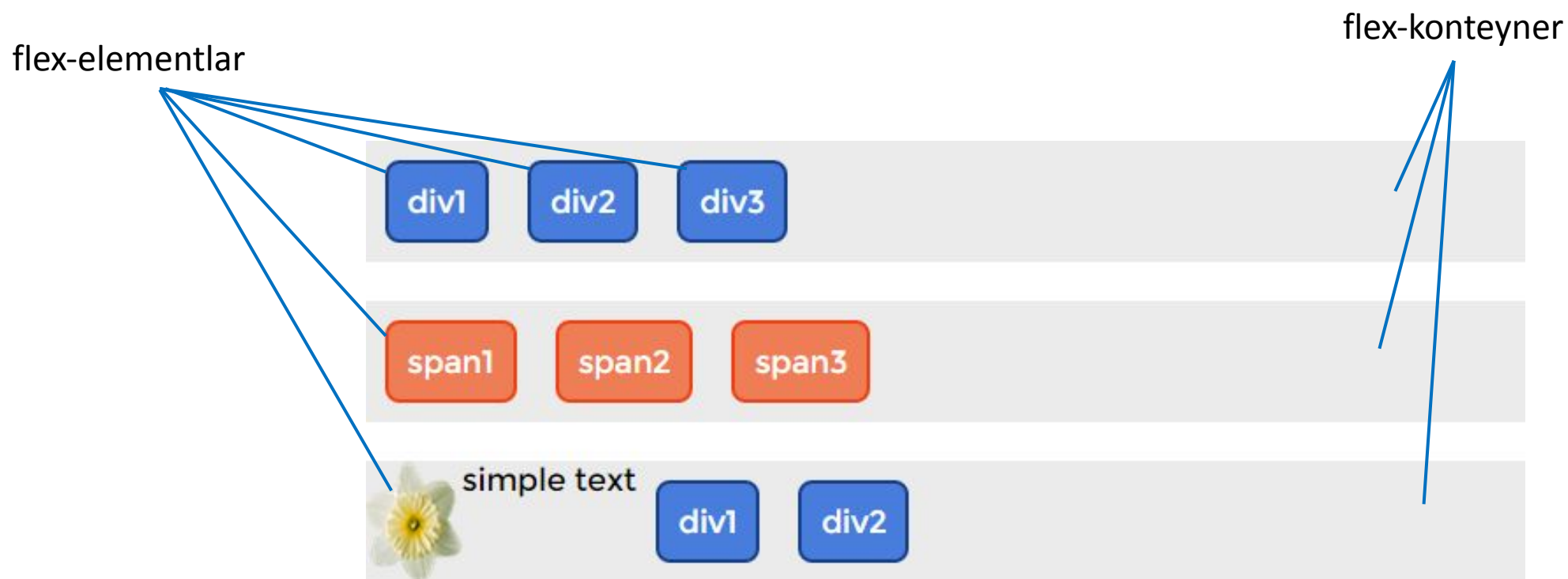


# Moslashuvchan elementlar - Flex



# Flex-konteyner

`display: flex` xossasi orqali o'rnatiladi.



# Flex elementlar yo'nalishi

---

**flex-direction** – asosiy o'q bo'yicha yo'nalishi

Qiymatlari: row | row-reverse | column | column-reverse

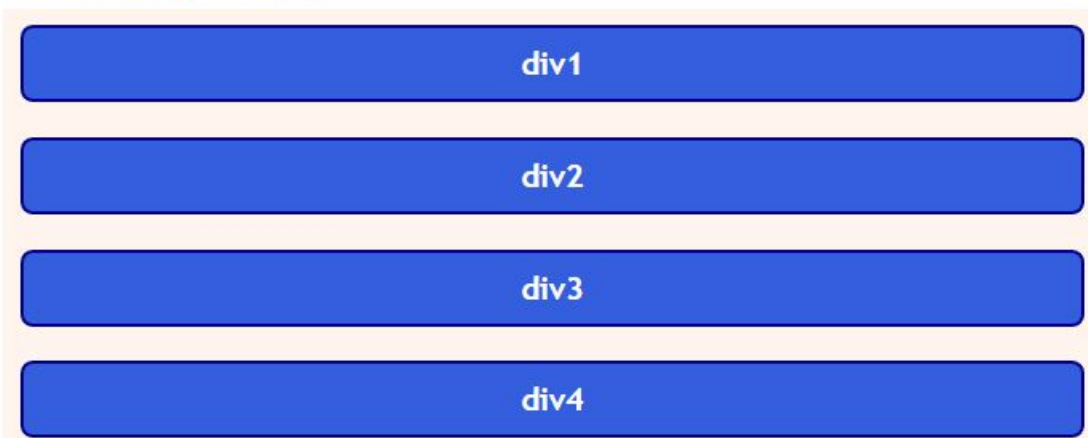
flex-direction: row;



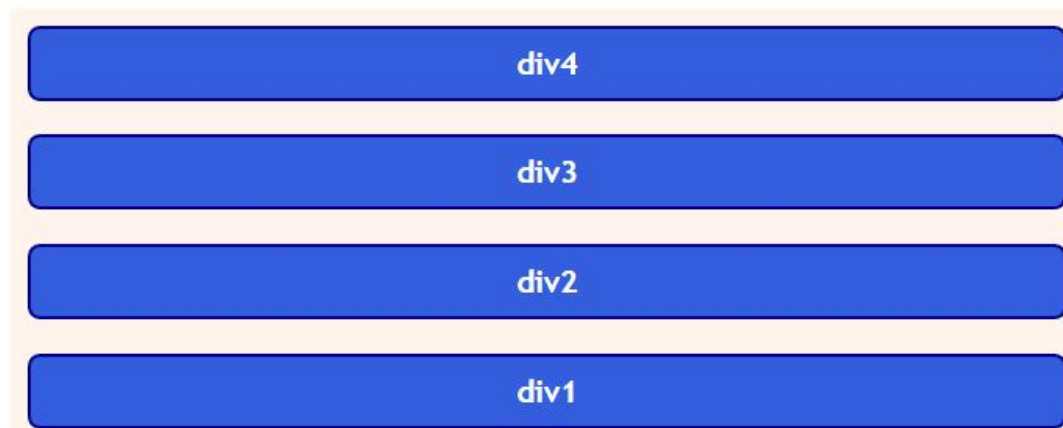
flex-direction: row-reverse;



flex-direction: column;



flex-direction: column-reverse;



# Flex-konteyner ko'p qatorlilikini boshqarish

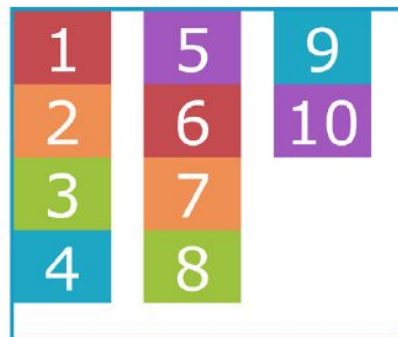
## flex-wrap

Qiymatlari: nowrap | wrap | wrap-reverse

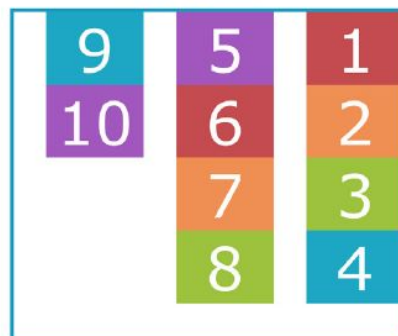
flex-wrap: nowrap; (default)



flex-wrap: wrap;



flex-wrap: wrap-reverse;



flex-wrap: wrap;



flex-wrap: wrap-reverse;



flex-wrap: nowrap; (default)



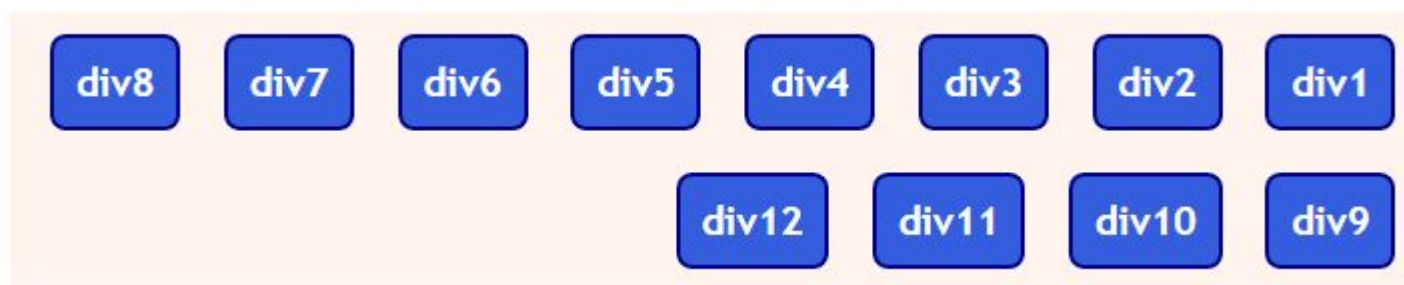
# Yo'nalish va ko'p qatorlilik

---

**flex-flow**

Qiymatlari: *flex-direction flex-wrap*

**flex-flow**: row-reverse wrap;



# Elementlarni tartibi

---

**order** – flex elementlarning flex-konteynerdagi joylashish tartibini aniqlaydi

Qiymatlari: *son*





# Moslashuvchanlik

---

## flex

Qiymatlari: `flex-grow` `flex-shrink` `flex-basis`

**flex-grow** – flex konteyner bo'sh maydonni avlod elementlari orasida taqsimlashi koeffitsienti

Qiymatlari: *son*

**flex-shrink** - flex elementni kichrayish koeffitsienti

Qiymatlari: *son*

**flex-basis** - bo'sh joy ajratilgunga qadar flex elementning boshlang'ich bazaviy o'lchamini aniqlaydi.

Qiymatlari: *uzunlik* | `auto` | `content`



# Tekishlash (alignment)

**justify-content** – asosiy o'q bo'yicha flex elementlardan qolgan bo'sh joyga nisbatan tekishlash

Qiymatlari: flex-start | flex-end | center | space-between | space-around

Qo'llanilishi: flex konteynerga

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

justify-content: flex-start; (default)



justify-content: flex-end;



justify-content: center;



justify-content: space-between;



justify-content: space-around;



# Tekishlash (alignment)

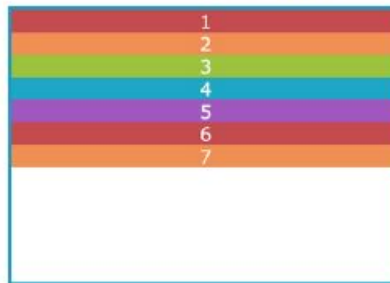
**justify-content** – asosiy o'q bo'yicha tekishlash

Qiymatlari: flex-start | flex-end | center | space-between | space-around

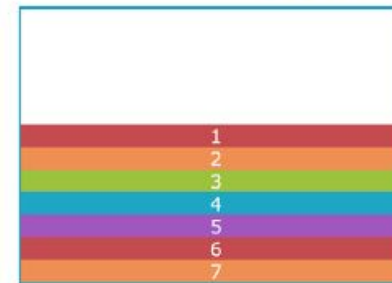
Qo'llanilishi: flex konteynerga

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

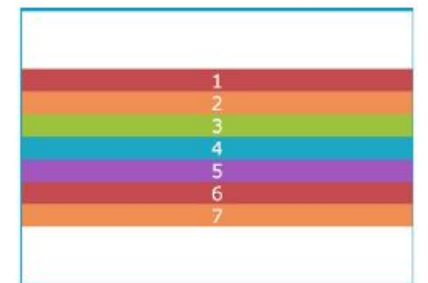
`justify-content: flex-start;` (default)



`justify-content: flex-end;`



`justify-content: center;`



`justify-content: space-between;`



`justify-content: space-around;`



# Tekishlash (alignment)

**align-items** – asosiy o'qqa perpendikulyar o'q bo'yicha tekishlash

Qiymatlari: flex-start | flex-end | center | baseline | stretch

Qo'llanilishi: flex konteynerga

```
#container {  
  display: flex;  
  flex-direction: row;  
  height: 200px;  
  align-items: flex-start;  
}
```

align-items: flex-start;



align-items: flex-end;



align-items: center;



align-items: stretch; (default)



align-items: baseline;



# Tekishlash (alignment)

---

**align-self** – alohida olingan flex elementni flex konteyner uzunligi bo'yicha tekishlash

Qiymatlari: auto | flex-start | flex-end | center | baseline | stretch

Qo'llanilishi: flex elementlarga

```
.box4 {  
  align-self: flex-end;  
}
```



# Tekishlash (alignment)

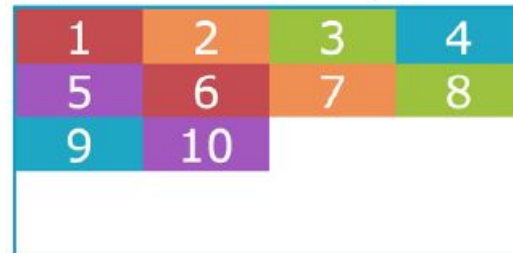
**align-content** – perpendikulyar o'q bo'yicha qo'shimcha bo'sh joy qolganda qatorlarni tekishlash

Qiymatlari: flex-start | flex-end | center | space-around | space-between | stretch

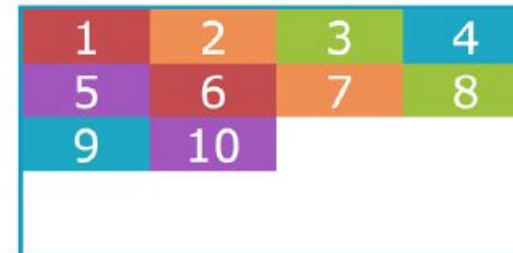
Qo'llanilishi: ko'p qatorli flex konteynerlarga

```
#container {  
  display: flex;  
  flex-direction: row;  
  flex-wrap: wrap;  
  height: 350px;  
  align-items: flex-start;  
}  
  
.box {  
  width: 25%;  
}
```

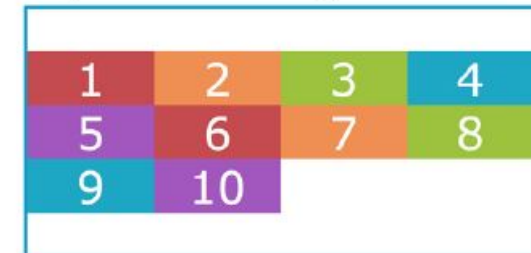
align-content: flex-start;



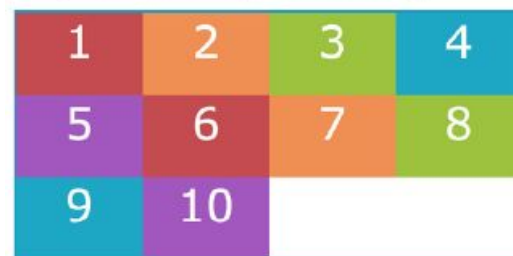
align-content: flex-end;



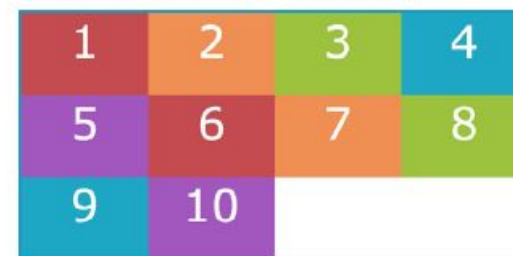
align-content: center;



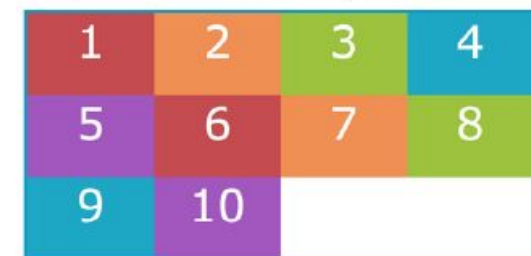
align-content: space-between;



align-content: space-around;



align-content: stretch; (default)



# Savollar?

---