



## ОТЗИВЧИВ УЕБ ДИЗАЙН С FLEX ТЕХНОЛОГИЯ



### Модул 2: CSS

# СЪДЪРЖАНИЕ

---

Свойство display: flex .....	3
Свойства на flex контейнера .....	6
FLEX-DIRECTION .....	6
FLEX-WRAP .....	11
FLEX-FLOW .....	14
JUSTIFY-CONTENT .....	17
ALIGN-ITEMS.....	22
ALIGN-CONTENT.....	28
Свойства на елементите на flex контейнера .....	30
ORDER .....	30
FLEX-GROW .....	34
FLEX-SHRINK .....	37
FLEX-BASIS .....	40
ALIGN-SELF .....	43
FLEX .....	46

## Когато завършите тази лекция:

---

- *ще се запознаете как се създава flex контейнер;*
- *ще се научите как елементите в контейнера стават flex елементи;*
- *ще се научите да прилагате свойства към flex контейнера и елементите.*

## Свойство display: flex

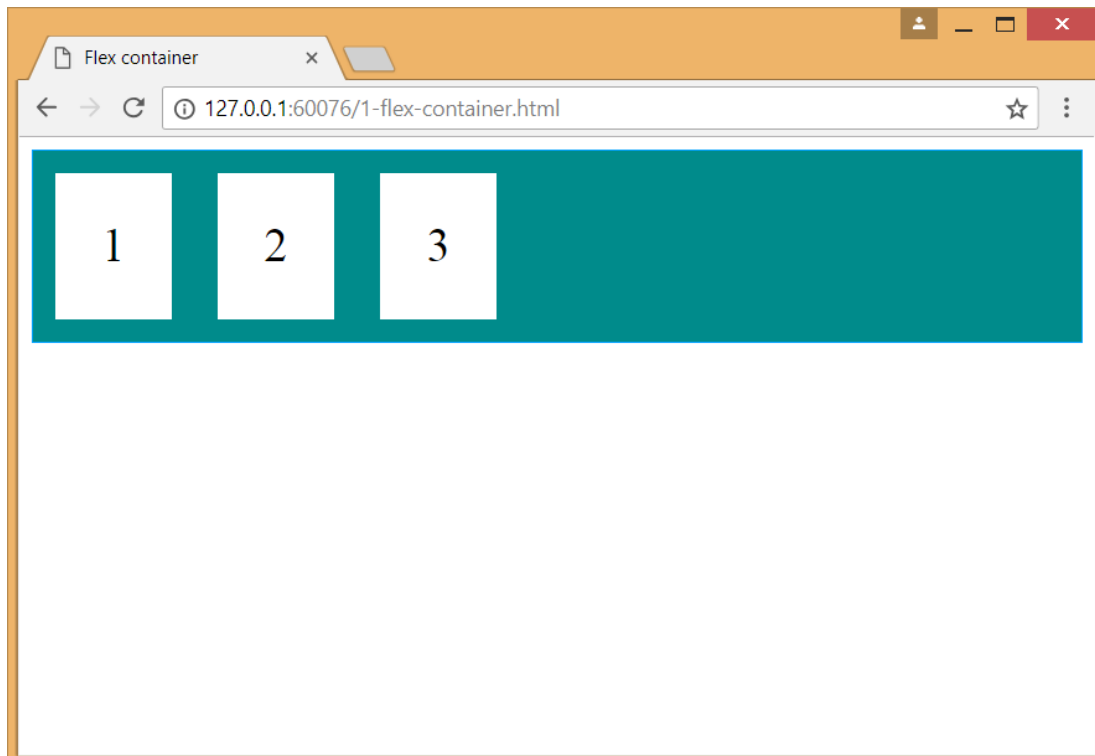
Всички елементи, които принадлежат на родителския със свойство **display: flex;**, стават flex елементи (фиг.1).

### ПРИМЕР:

HTML

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

```
.flex-container {  
    display: flex;  
    background-color: darkcyan;  
}  
.flex-container > div {  
    background-color: white;  
    margin: 15px;  
    padding: 30px;  
    font-size: 30px;  
}
```



*Фиг. 1 Елементи във flex контейнер*

# Свойства на flex контейнера

## FLEX-DIRECTION

Свойство **flex-direction** - дефинира посоката, в която да се подредят елементите в контейнера.

- **column** - Подрежда елементите вертикално отгоре надолу (фиг.2)
- **column-reverse** - Подрежда елементите вертикално отдолу нагоре
- **row** - Подрежда елементите хоризонтално отляво надясно (фиг.3)
- **row-reverse** - Подрежда елементите хоризонтално отдясно наляво

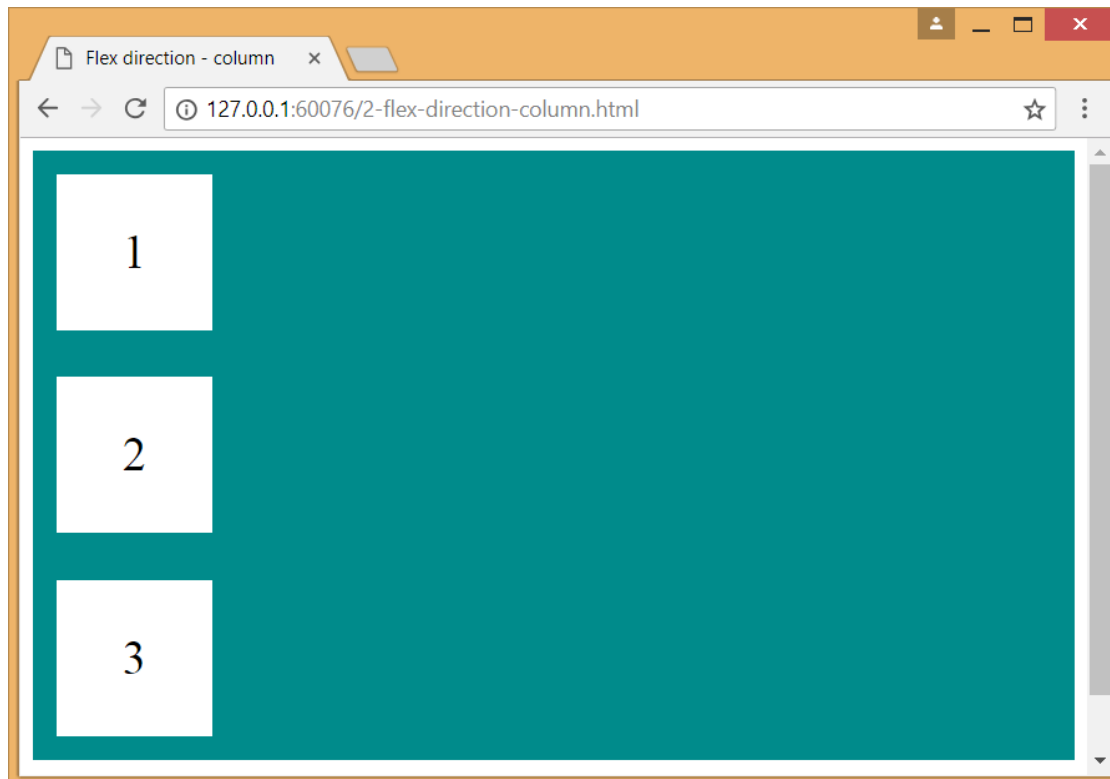
## ПРИМЕР ЗА ПОДРЕЖДАНЕ В КОЛОНА:

HTML

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

```
.flex-container {  
    display: flex;  
    flex-direction: column;  
    background-color: darkcyan;  
}  
  
.flex-container > div {  
    background-color: white;  
    width: 100px;  
    margin: 15px;  
    text-align: center;  
    line-height: 100px;  
    font-size: 30px;  
}
```





Фиг.2 Подредане на елементи във flex контейнер вертикално

## ПРИМЕР ЗА ПОДРЕЖДАНЕ НА РЕД:

### HTML

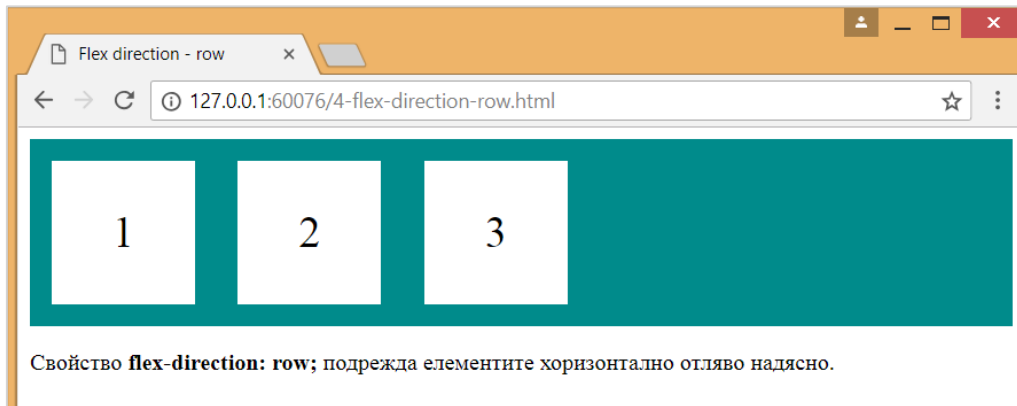
```
<div class="flex-container">
  <div> 1 </div>
  <div> 2 </div>
  <div> 3 </div>
</div>
```

### CSS

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

.flex-container > div {
```

```
background-color: white;
width: 100px;
margin: 15px;
text-align: center;
line-height: 100px;
font-size: 30px;
}
```



*Фиг.3 Подредане на елементи във flex контейнер хоризонтално*

## FLEX-WRAP

Свойство **flex-wrap** - определя как елементите да бъдат обхванати

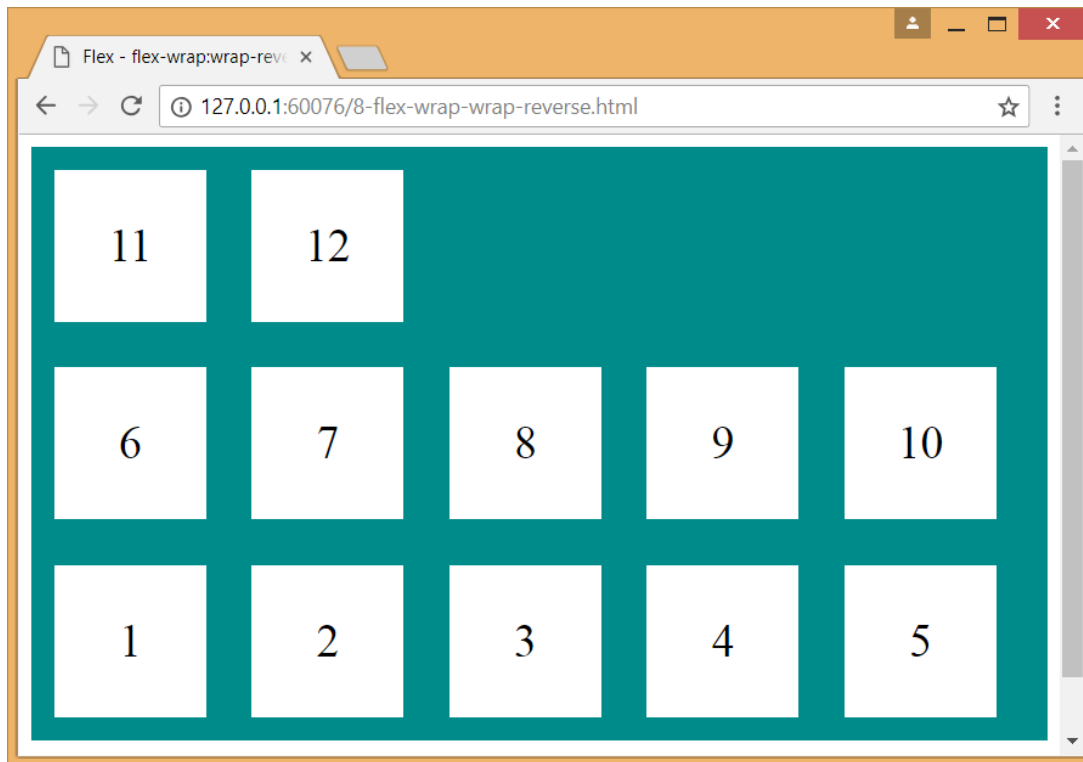
- **wrap** - обхваща елементи в рамка
- **nowrap** - определя елементи да не бъдат обхванати в рамка
- **wrap-reverse** - обхваща елементи в рамка в обратен ред (фиг.4)

### ПРИМЕР:

#### HTML

```
<div class="flex-container">  
  <div> 1 </div>  
  <div> 2 </div>  
  <div> 3 </div>  
  <div> 4 </div>  
  .....  
  <div> 12 </div>  
</div>
```

```
.flex-container {  
    display: flex;  
    flex-wrap: wrap-reverse;  
    background-color: darkcyan;  
}  
  
.flex-container > div {  
    background-color: white;  
    width: 100px;  
    margin: 15px;  
    text-align: center;  
    line-height: 100px;  
    font-size: 30px;  
}
```



Фиг.4 Подредане на елементи във flex контейнер в рамка в обратен ред

## FLEX-FLOW

Свойство **flex-flow** - е обобщаващо свойство, което задава стойности от свойства flex-direction и flex-wrap (фиг.5).

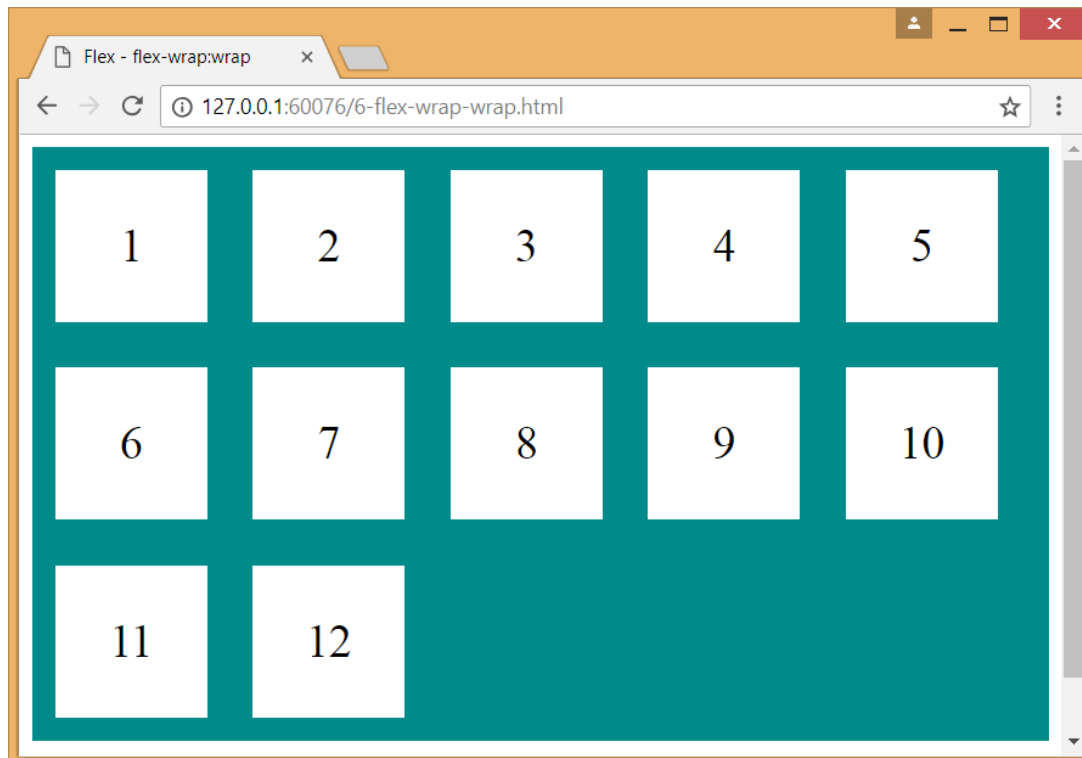
### ПРИМЕР:

HTML

```
<div class="flex-container">
  <div> 1 </div>
  <div> 2 </div>
  <div> 3 </div>
  <div> 4 </div>
  .....
  <div> 12 </div>
</div>
```

```
.flex-container {  
    display: flex;  
    flex-flow: row wrap;  
    background-color: darkcyan;  
}  
  
.flex-container > div {  
    background-color: white;  
    width: 100px;  
    margin: 15px;  
    text-align: center;  
    line-height: 100px;  
    font-size: 30px;  
}
```





Фиг.5 Подреждане със свойство *flex-flow*

## JUSTIFY-CONTENT

Свойство **justify-content** - служи за хоризонтално подравняване на flex елементите.

- **center** - центрира flex елементите (фиг.6)
- **flex-start** - подравнява елементите в началото на контейнера
- **flex-end** - подравнява елементите в края на контейнера
- **space-around** - представя елементите с разстояние около тях
- **space-between** - представя елементите с разстояние между тях (фиг.7)

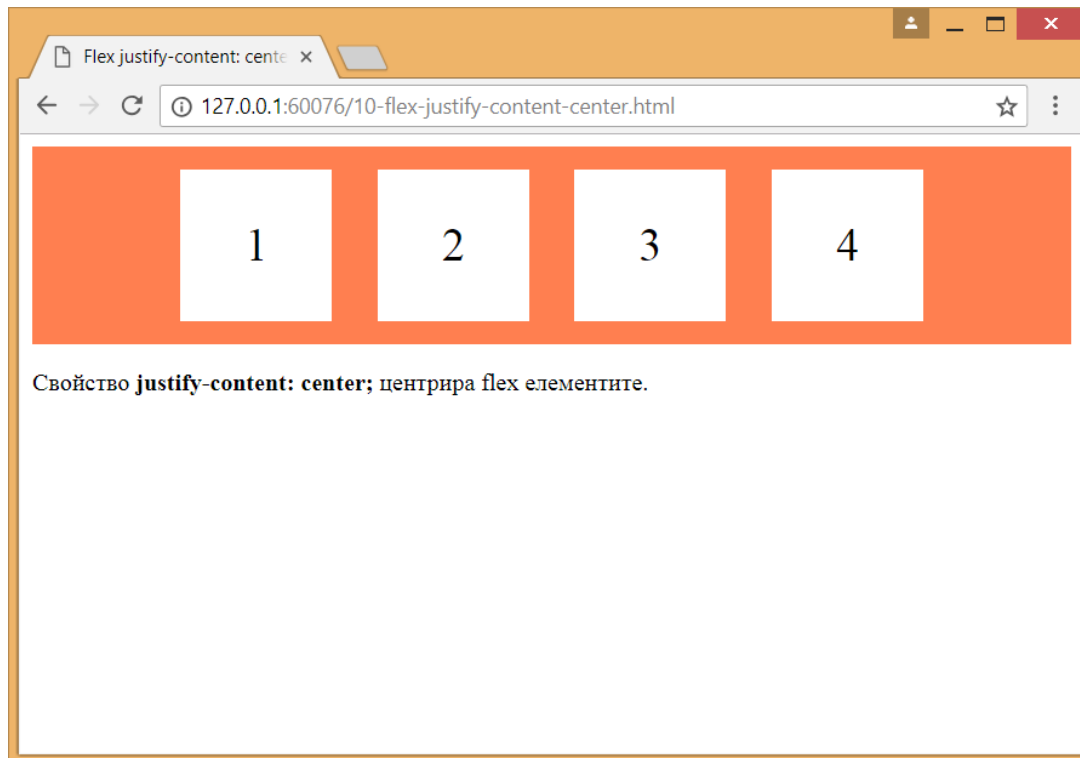
### ПРИМЕР:

#### HTML

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

## CSS

```
.flex-container {  
    display: flex;  
    justify-content: center;  
    background-color: coral;  
}  
  
.flex-container > div {  
    background-color: white;  
    width: 100px;  
    margin: 15px;  
    text-align: center;  
    line-height: 100px;  
    font-size: 30px;  
}
```



Фиг.6 Центриране на flex элементите

## ПРИМЕР:

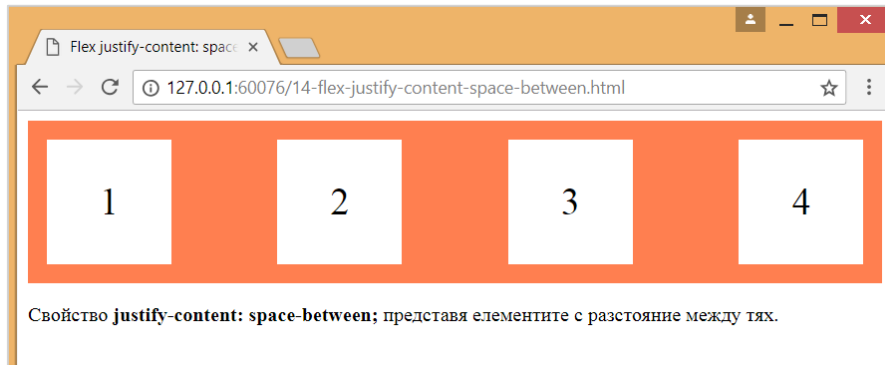
### HTML

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

### CSS

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

```
.flex-container > div {  
    background-color: white;  
    width: 100px;  
    margin: 15px;  
    text-align: center;  
    line-height: 100px;  
    font-size: 30px;  
}
```



Фиг.7 Представяне flex елементите с разстояние около тях

## ALIGN-ITEMS

Свойство **align-items** - служи за вертикално подравняване на flex елементите

- **center** - подравнява елементите в контейнера вертикално в центъра (фиг.8)
- **flex-start** - подравнява елементите в контейнера вертикално отгоре
- **flex-end** - подравнява елементите в контейнера вертикално отдолу
- **stretch** - подравнява елементите, така че да изпълват контейнера вертикално (фиг.9)
- **baseline** - подравнява елементите в контейнера вертикално по линия

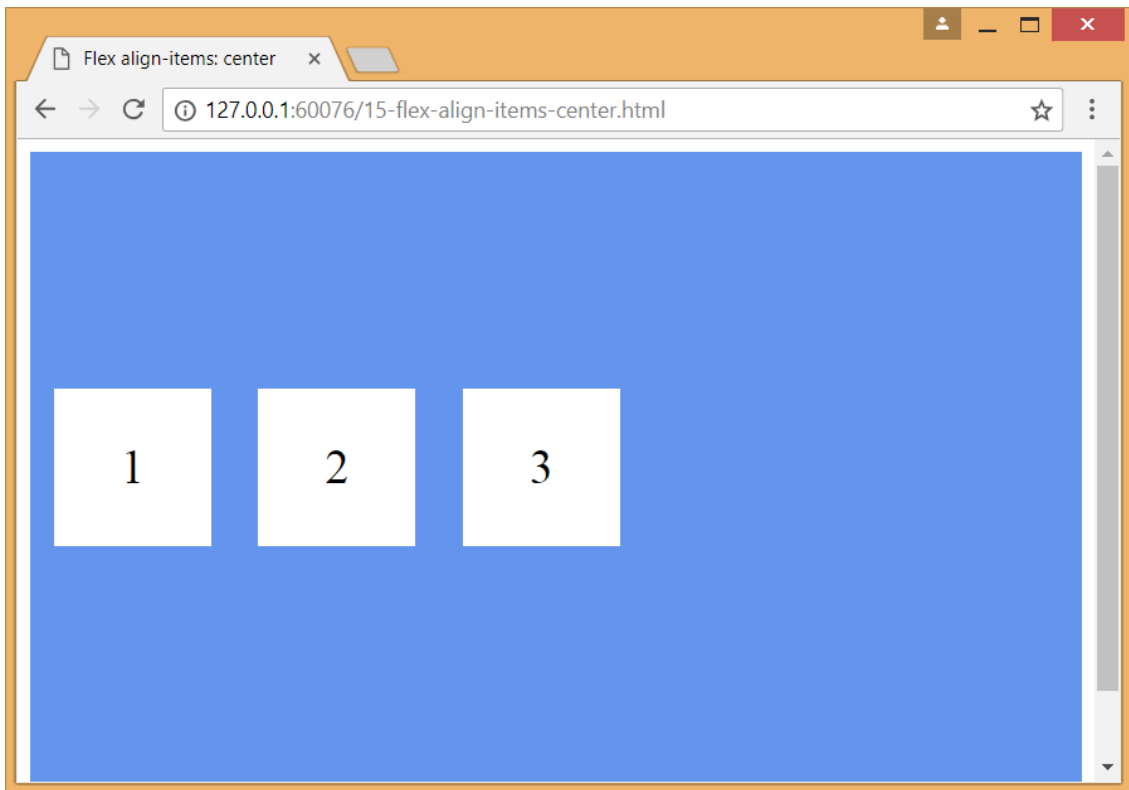
### ПРИМЕР:

HTML

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

```
.flex-container {  
  display: flex;  
  height: 400px;  
  align-items: center;  
  background-color: cornflowerblue;  
}  
.flex-container > div {  
  background-color: white;  
  width: 100px;  
  margin: 15px;  
  text-align: center;  
  line-height: 100px;  
  font-size: 30px;  
}
```





Фиг.8 Подравняване на flex елементите в контейнера вертикално в центъра

## ПРИМЕР:

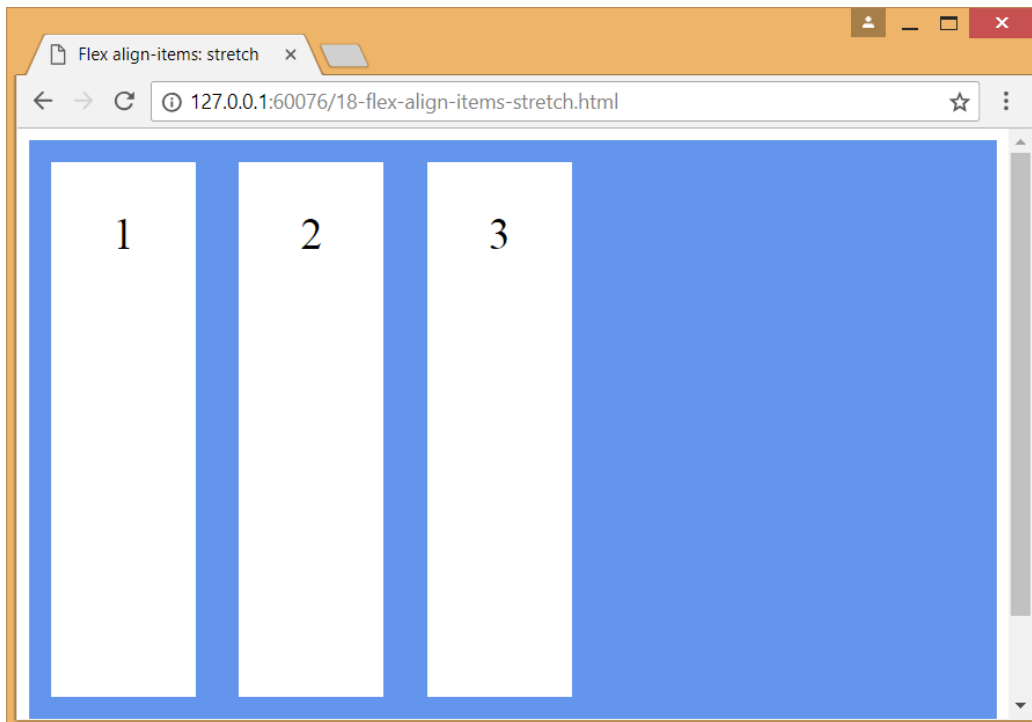
### HTML

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

### CSS

```
.flex-container {
  display: flex;
  height: 400px;
  align-items: stretch;
  background-color: cornflowerblue;
```

```
}  
.flex-container > div {  
  background-color: white;  
  width: 100px;  
  margin: 15px;  
  text-align: center;  
  line-height: 100px;  
  font-size: 30px;  
}
```



Фиг.9 Подравнява flex елементите, така че да изпълват контейнера вертикално

## ALIGN-CONTENT

Свойство **align-content** - служи за подравняване на flex елементите по линии.

- **space-between** - подравнява flex елементите по линии с разстояние между тях
- **space-around** - подравнява flex елементите по линии с разстояние около тях
- **stretch** - подравнява flex елементите по линии с разстояние около тях и височина изпълваща контейнера.
- **center** - подравнява flex елементите по линии вертикално в центъра на контейнера.
- **flex-start** - подравнява flex елементите по линии вертикално отгоре на контейнера (фиг.10).
- **flex-end** - подравнява flex елементите по линии вертикално отдолу на контейнера.

### ПРИМЕР:

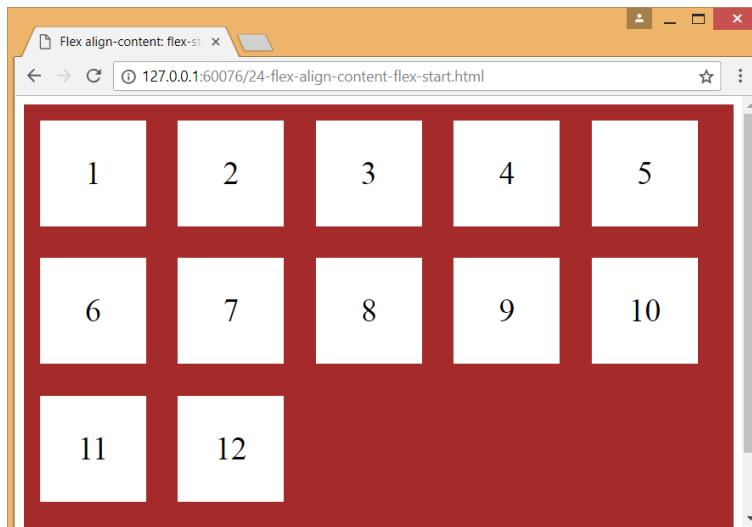
HTML

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

```
<div> 12 </div>  
</div>
```

## CSS

```
.flex-container {  
  display: flex;  
  height: 400px;  
  flex-wrap: wrap;  
  align-content: flex-start;  
  background-color: brown;  
}  
.flex-container > div {  
  background-color: white;  
  width: 100px;  
  margin: 15px;  
  text-align: center;  
  line-height: 100px;  
  font-size: 30px;  
}
```



Фиг.10. Подравняване flex елементите по линии вертикално отгоре на контейнера

## Свойства на елементите на flex контейнера

### ORDER

Свойство **order** подрежда flex елементите на flex контейнера (фиг. 11).

## ПРИМЕР:

### HTML

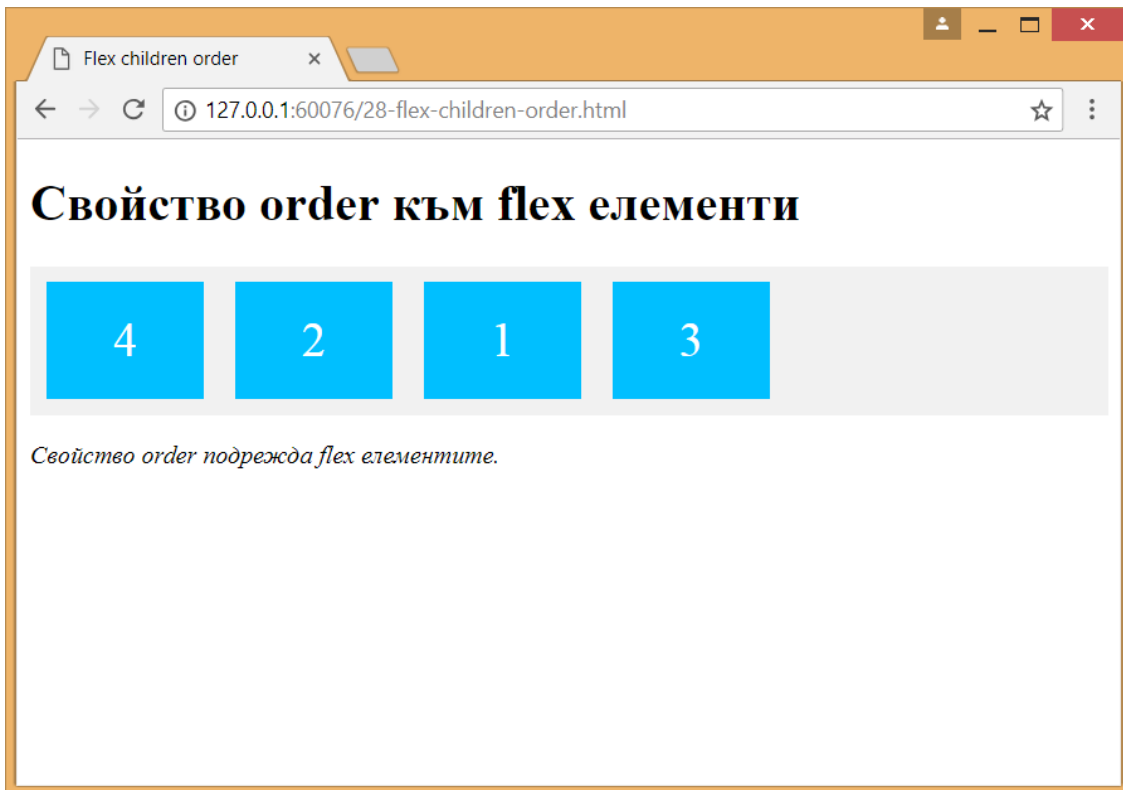
```
<div class="flex-container">
  <div style="order: 3"> 1 </div>
  <div style="order: 2"> 2 </div>
  <div style="order: 4"> 3 </div>
  <div style="order: 1"> 4 </div>
</div>
```

### CSS

```
.flex-container {
  display: flex;
  align-items: stretch;
  background-color: #f1f1f1;
```



```
}  
.flex-container > div {  
  background-color: deepskyblue;  
  color: white;  
  width: 100px;  
  margin: 10px;  
  text-align: center;  
  line-height: 75px;  
  font-size: 30px;  
}
```



Фиг.11 Прилагане на свойство order към елементите на контейнера

## FLEX-GROW

Свойство **flex-grow** задава flex елементите да запълват останалото пространство в контейнера (фиг.12). Стойността по подразбиране е 0.

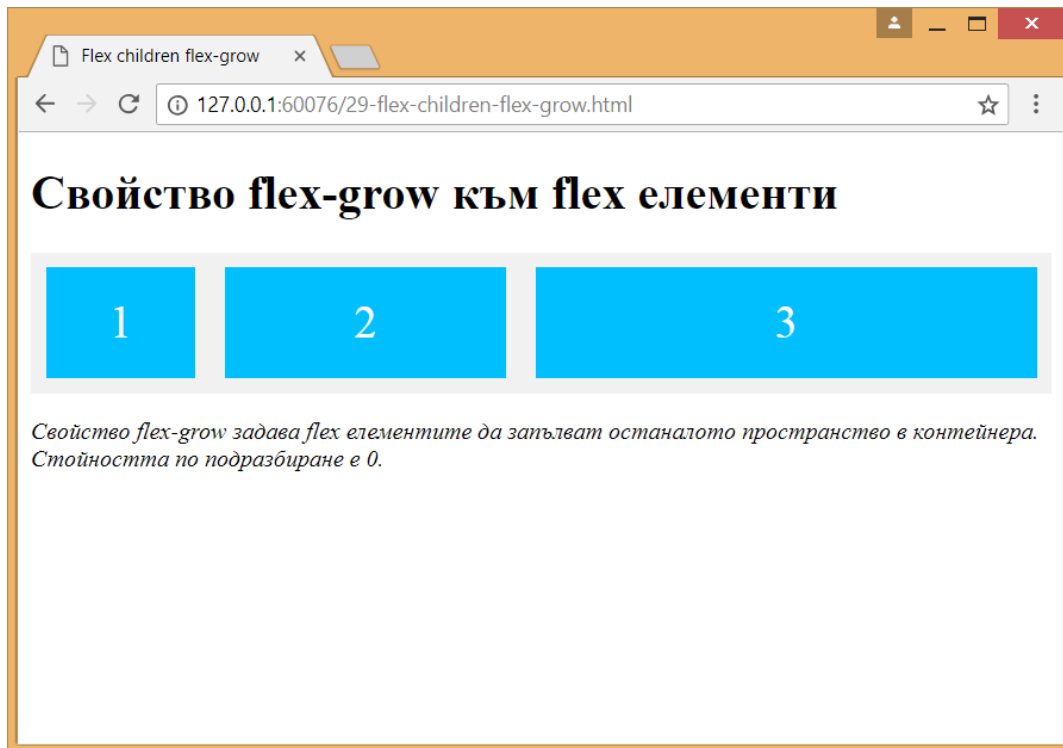
### ПРИМЕР:

#### HTML

```
<div class="flex-container">  
  <div style="flex-grow: 0"> 1 </div>  
  <div style="flex-grow: 3"> 2 </div>  
  <div style="flex-grow: 8"> 3 </div>  
</div>
```

#### CSS

```
.flex-container {  
  display: flex;  
  align-items: stretch;  
  background-color: #f1f1f1;  
}  
.flex-container > div {  
  background-color: deepskyblue;  
  color: white;  
  width: 100px;  
  margin: 10px;  
  text-align: center;  
  line-height: 75px;  
  font-size: 30px;  
}
```



Фиг.12. Прилагане на свойство flex-grow към елементите на контейнера

## FLEX-SHRINK

Свойство **flex-shrink** задава как flex елементите да свиват съответно останалите (фиг.13). Стойността по подразбиране е 1.

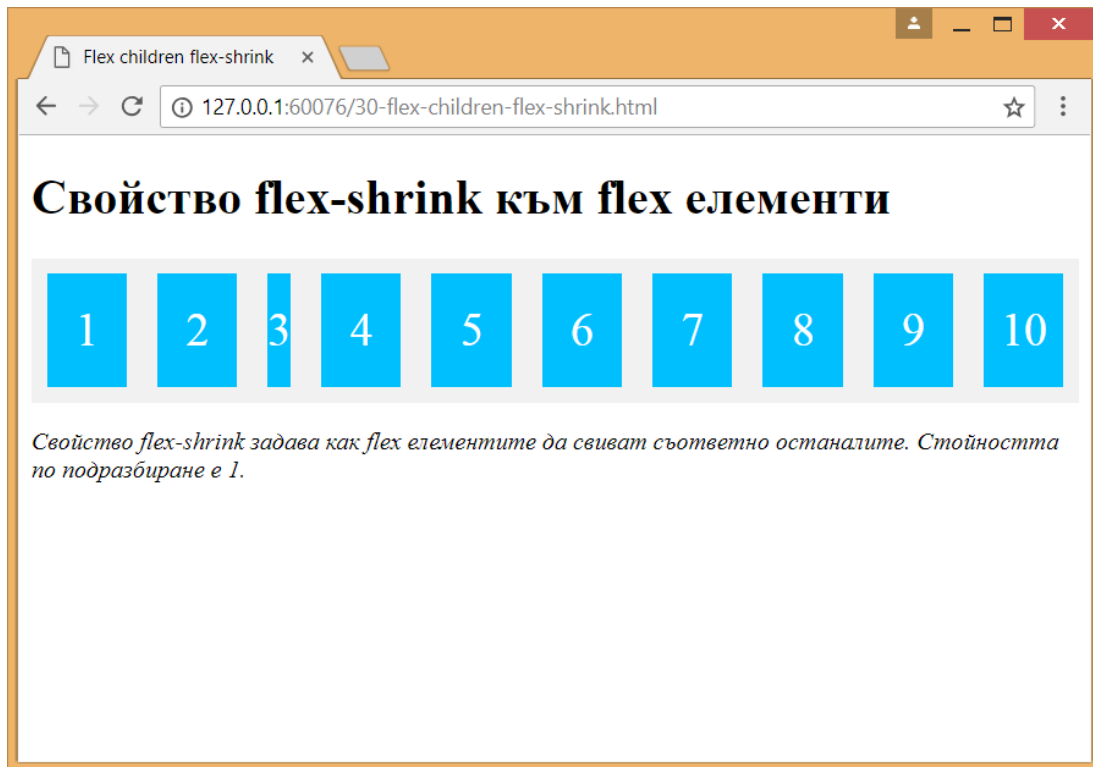
### ПРИМЕР:

HTML

```
<div class="flex-container">
  <div> 1 </div>
  <div> 2 </div>
  <div style=" flex-shrink: 6"> 3 </div>
  <div> 4 </div>
  .....
  <div> 12 </div>
</div>
```

## CSS

```
.flex-container {  
  display: flex;  
  align-items: stretch;  
  background-color: #f1f1f1;  
}  
.flex-container > div {  
  background-color: deepskyblue;  
  color: white;  
  width: 100px;  
  margin: 10px;  
  text-align: center;  
  line-height: 75px;  
  font-size: 30px;  
}
```



Фиг.13 Прилагане на свойство flex-shrink към елементите на контейнера



## FLEX-BASIS

Свойство **flex-basis** определя ширината на flex елементите (фиг.14).

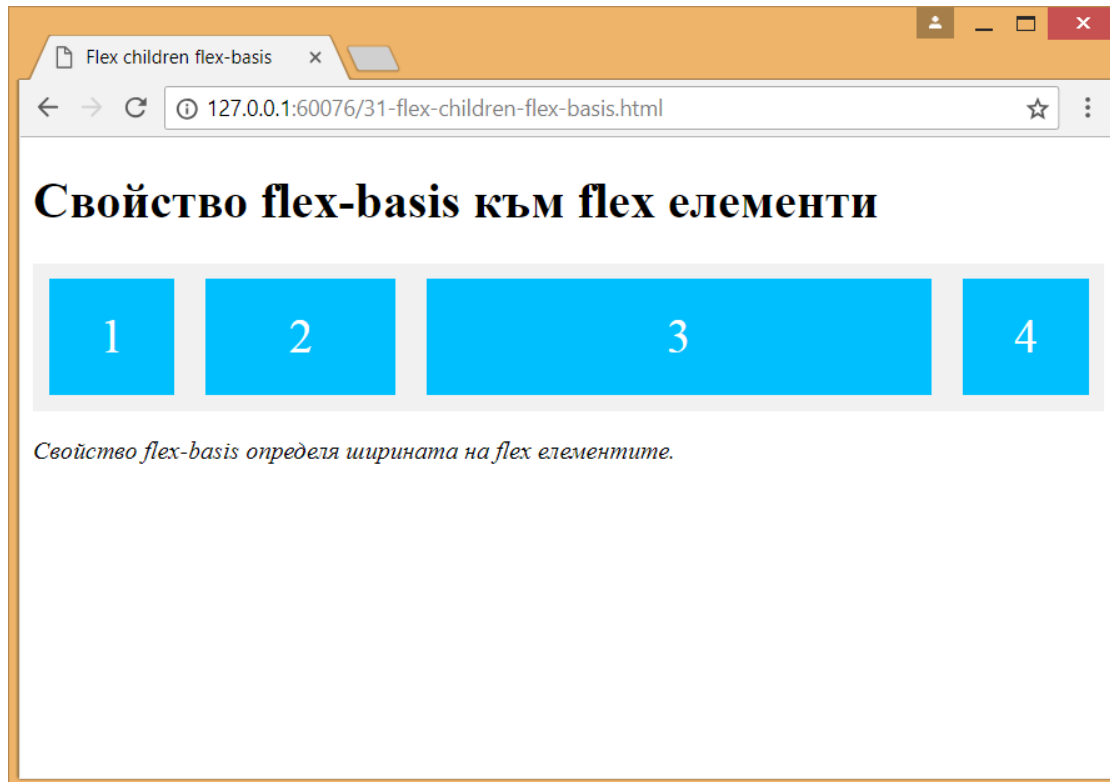
### ПРИМЕР:

#### HTML

```
<div class="flex-container">
  <div> 1 </div>
  <div style="flex-basis: 150px"> 2 </div>
  <div style="flex-basis: 400px"> 3 </div>
  <div> 4 </div>
</div>
```

## CSS

```
.flex-container {  
  display: flex;  
  align-items: stretch;  
  background-color: #f1f1f1;  
}  
.flex-container > div {  
  background-color: deepskyblue;  
  color: white;  
  width: 100px;  
  margin: 10px;  
  text-align: center;  
  line-height: 75px;  
  font-size: 30px;  
}
```



Фиг.14 Прилагане на свойство flex-basis към елементите на контейнера

## ALIGN-SELF

Свойство **align-self** определяет подравняването на flex елементите в контейнера (фиг.15).

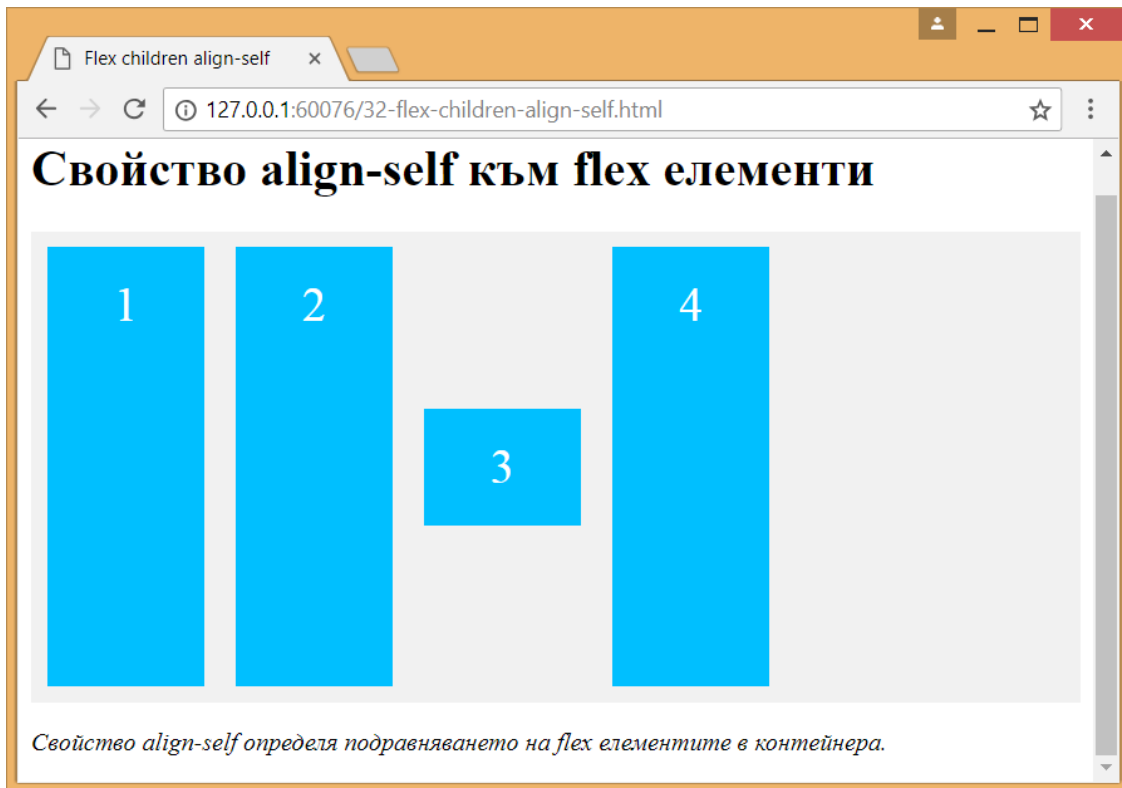
ПРИМЕР:

HTML

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

## CSS

```
.flex-container {  
  display: flex;  
  height: 300px;  
  background-color: #f1f1f1;  
}  
.flex-container > div {  
  background-color: deepskyblue;  
  color: white;  
  width: 100px;  
  margin: 10px;  
  text-align: center;  
  line-height: 75px;  
  font-size: 30px;  
}
```



Фиг.15 Прилагане на свойство `align-self` към елементите на контейнера

## FLEX

Свойство **flex** към flex елементите комбинира свойства flex-grow, flex-shrink и flex-basis (фиг.16).

### ПРИМЕР:

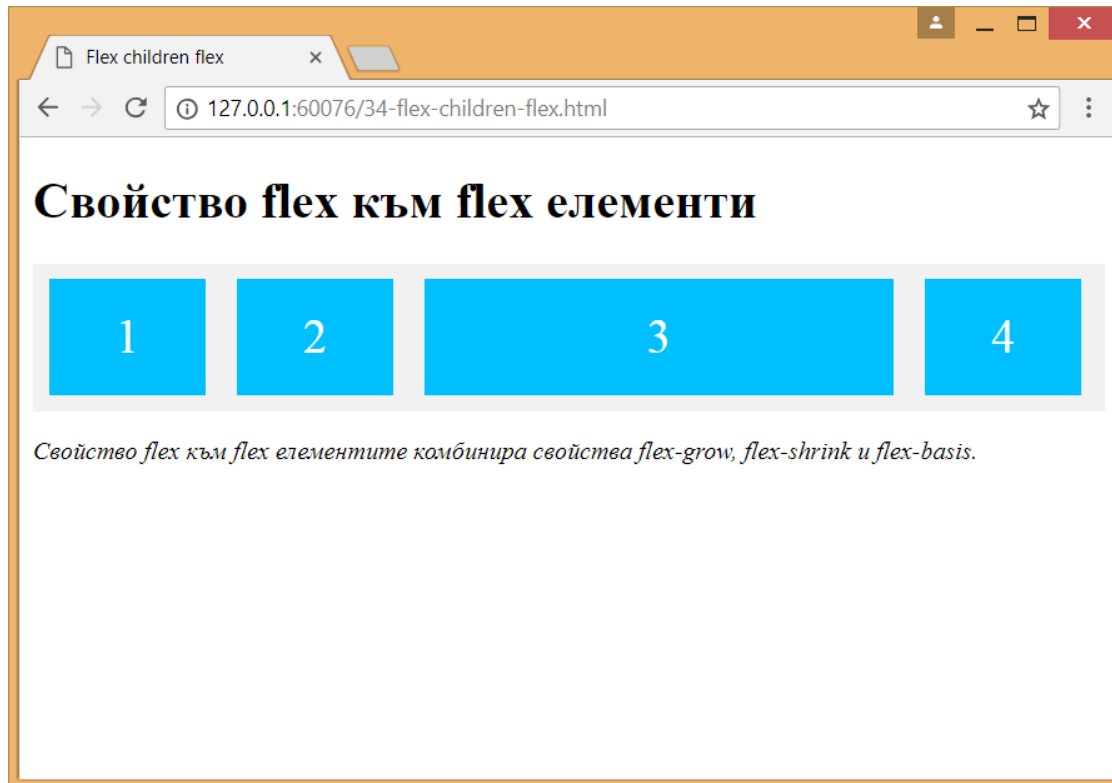
#### HTML

```
<div class="flex-container">  
  <div> 1 </div>  
  <div> 2 </div>  
  <div style="flex: 0 0 300px"> 3 </div>  
  <div> 4 </div>  
</div>
```

## CSS

```
.flex-container {  
  display: flex;  
  align-items: stretch;  
  background-color: #f1f1f1;  
}  
.flex-container > div {  
  background-color: deepskyblue;  
  color: white;  
  width: 100px;  
  margin: 10px;  
  text-align: center;  
  line-height: 75px;  
  font-size: 30px;  
}
```





Фиг.16. Прилагане на свойство flex към елементите на контейнера