

# Верстка на основе flex-контейнеров (Flexbox)

---


# Почему Flexbox?

---

Долгое время единственными надёжными инструментами CSS-верстки были такие способы как float (обтекание) и position (позиционирование).

С их помощью сложно или невозможно достичь следующих простых требований к макету:

- ❑ Вертикального выравнивания блока внутри родителя.
- ❑ Оформление всех детей контейнера так, чтобы они распределили между собой доступную ширину/высоту, независимо от того, сколько ширины/высоты доступно.
- ❑ Сделать все колонки в макете одинаковой высоты, даже если наполнение в них различно.

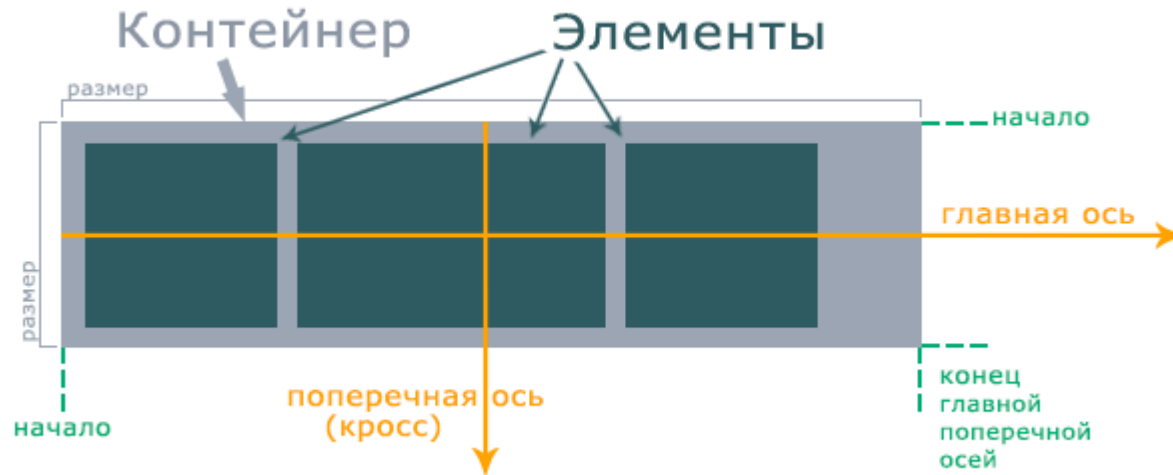


Модель верстки **Flexbox**, создана, чтобы убрать недостатки при создании самых разных HTML-конструкций, в том числе адаптированных под разную ширину и высоту, и сделать верстку логичной и простой

# Flexbox

Flexbox предназначен для создания гибких макетов.

Flexbox определяет набор CSS свойств для контейнера (flex-контейнер) и его дочерних элементов (flex-блоков).

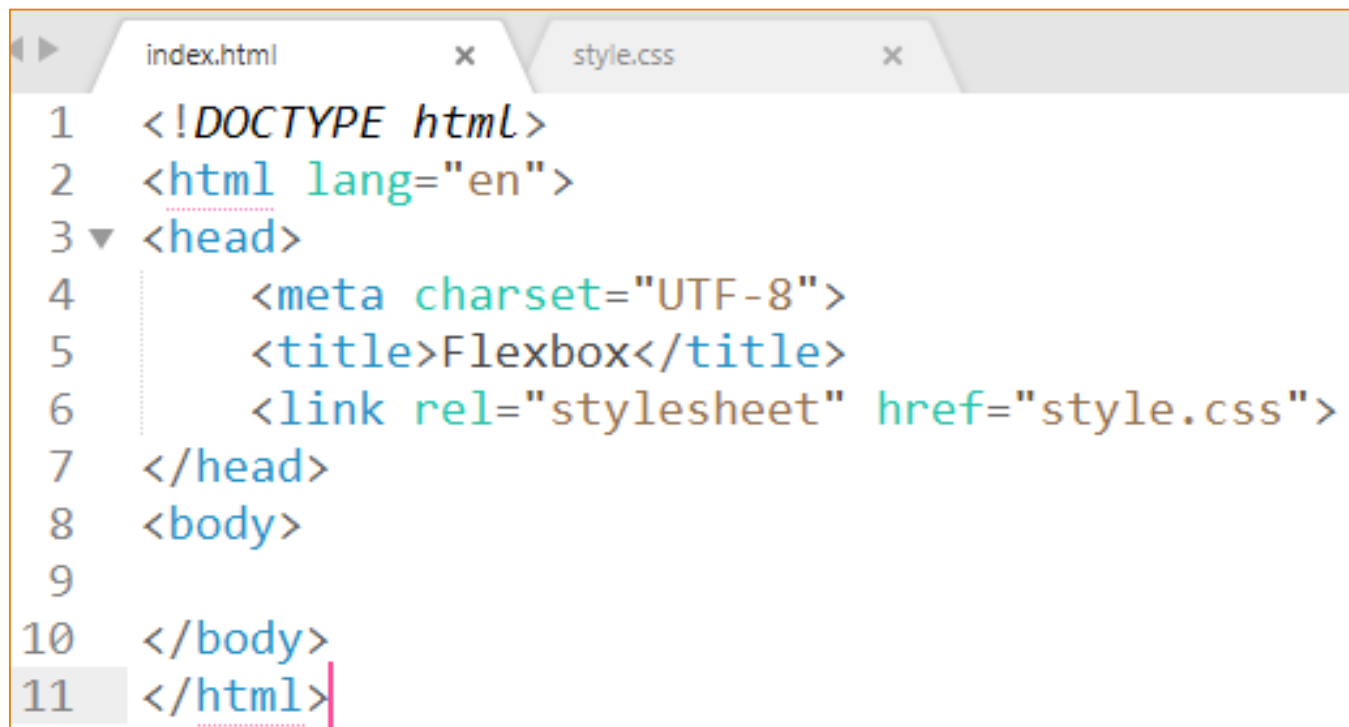


FlexBox состоит из  
**Контейнера** и его **Дочерних**  
**элементов.**

*При изучении новой темы выполните практическое задание на использование верстки Flexbox.*

---

Создайте папку с 2 файлами index.html и style.css. Подключите файл стилей style.css к файлу index.html



```
1 <!DOCTYPE html>
2 <html lang="en">
3   <head>
4     <meta charset="UTF-8">
5     <title>Flexbox</title>
6     <link rel="stylesheet" href="style.css">
7   </head>
8   <body>
9
10  </body>
11 </html>
```

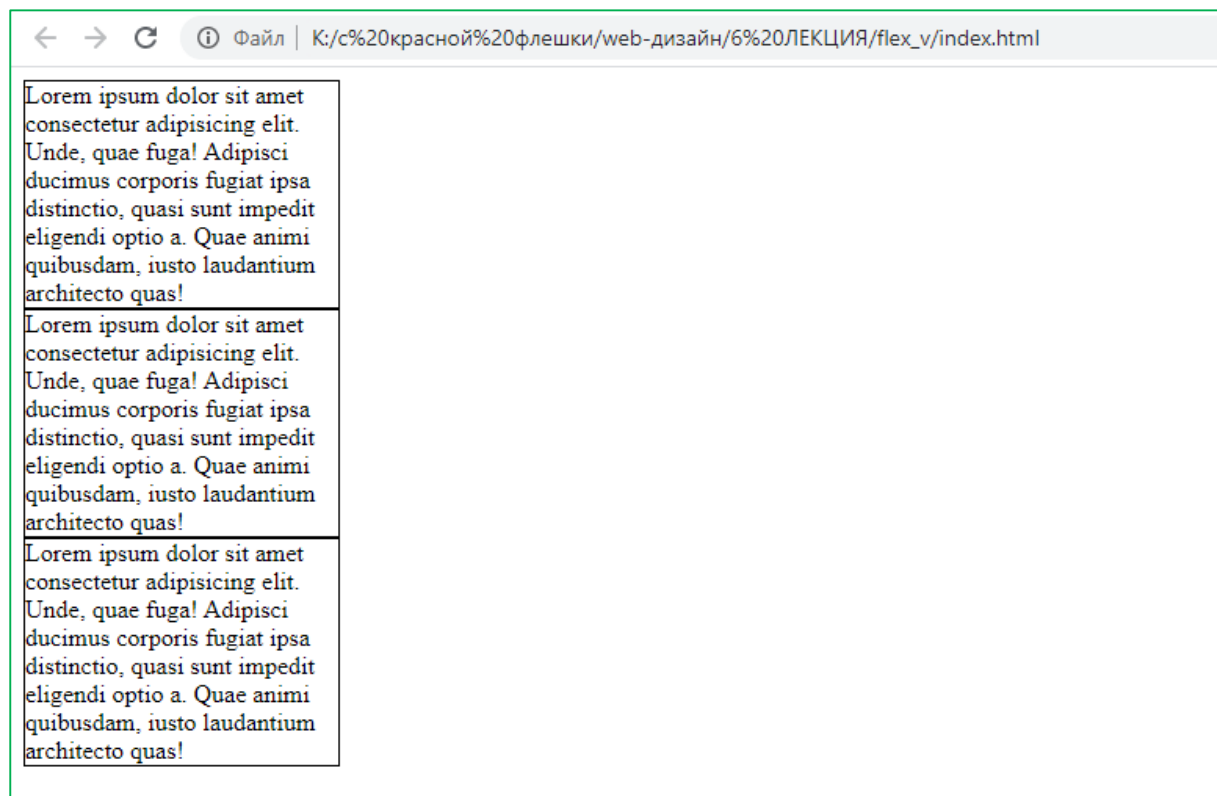
Создайте блок-обертку (`div class= "wrapper"`) и в нем несколько блоков с произвольным текстом.

Чтобы быстро создать произвольный текст в блоках, наберите слово Lorem и нажмите Tab

```
8 ▼ <body>
9 ▼   <div class="wrapper">
10 ▼     <div class="item">
11       Lorem ipsum dolor sit amet consectetur adipisicing elit.
12       Unde, quae fuga! Adipisci ducimus corporis fugiat ipsa distinctio,
13       quasi sunt impedit eligendi optio a. Quae animi quibusdam,
14       iusto laudantium architecto quas!
15     </div>
16 ▼     <div class="item">
17       Lorem ipsum dolor sit amet consectetur adipisicing elit.
18       Unde, quae fuga! Adipisci ducimus corporis fugiat ipsa distinctio,
19       quasi sunt impedit eligendi optio a. Quae animi quibusdam,
20       iusto laudantium architecto quas!
21     </div>
22 ▼     <div class="item">
23       Lorem ipsum dolor sit amet consectetur adipisicing elit.
24       Unde, quae fuga! Adipisci ducimus corporis fugiat ipsa distinctio,
25       quasi sunt impedit eligendi optio a. Quae animi quibusdam,
26       iusto laudantium architecto quas!
27     </div>
28   </div>
29 </body>
```

Чтобы видеть результат, добавьте в файл style.css границу и ширину для блоков

```
1 .item{
2     width: 200px;
3     border: 1px solid black;
4 }
```



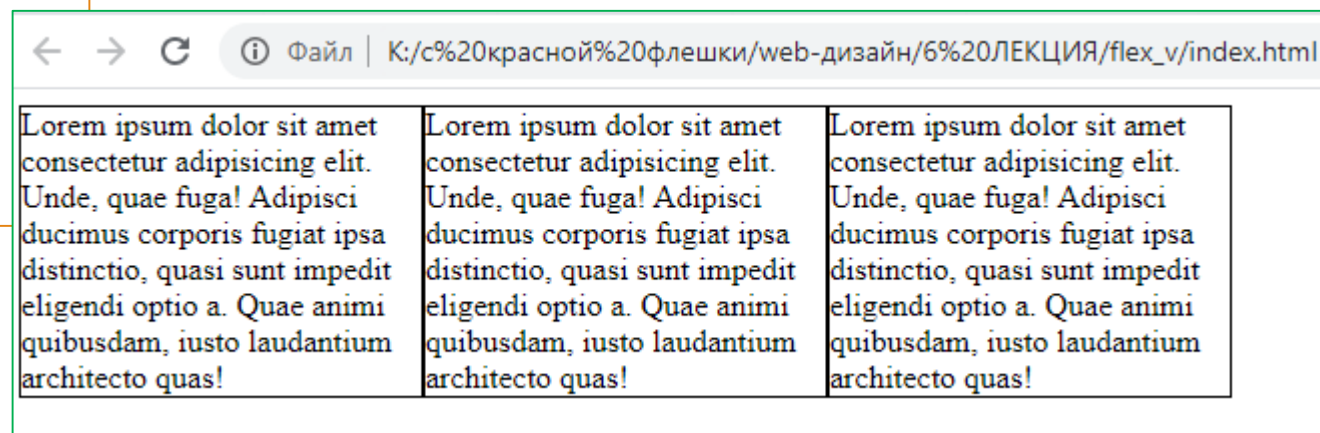
**Чтобы начать использовать модель Flexbox, необходимо задать родительский flex-контейнер.**

В нашем примере это будет `div class= "wrapper"`.



В файле style.css  
установите у селектора `.wrapper` свойству `display` значение `flex`

```
1 .wrapper{
2     display: flex;
3 }
4 .item{
5     width: 200px;
6     border: 1px solid black;
7 }
```



В результате на сайте все блоки должны встать в одну линию.



---

# Flex-свойства для родительского контейнера

# 1. Выравнивание блоков по основной оси (горизонтали)

---

Свойство **justify-content**

## Значения свойства **justify-content**:

**justify-content: center;** - выравнивание элементов по центру

**justify-content: flex-start;** - выравнивание элементов в начале контейнера (по умолчанию)

**justify-content: flex-end;** - выравнивание элементов в конце контейнера

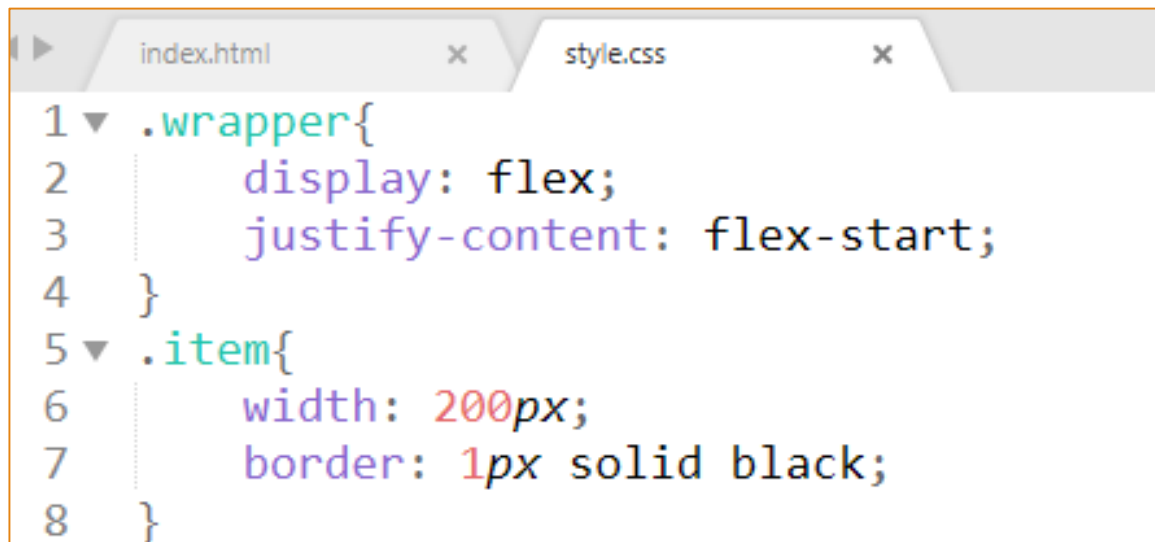
**justify-content: space-around;** - отступы и по бокам, и между элементами

**justify-content: space-between;** - крайние элементы прижимаются к краям, все остальное место делится поровну отступами между ними

# 1. Выравнивание блоков по горизонтали

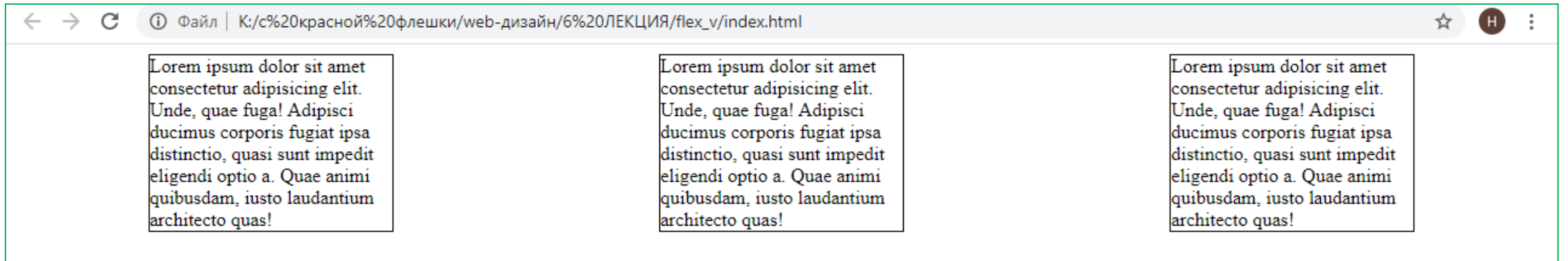
---

Для родителя (**div class= “wrapper”**) добавьте свойство **justify-content**.  
Поэкспериментируйте с его значениями.



```
index.html x style.css x
1 ▼ .wrapper{
2     display: flex;
3     justify-content: flex-start;
4 }
5 ▼ .item{
6     width: 200px;
7     border: 1px solid black;
8 }
```

Установите значение свойства **justify-content**, чтобы получить результат как на рисунке:



Чтобы изучать свойства flex дальше, добавьте произвольного текста в центральный блок.

## 2. Выравнивание блоков по поперечной оси (вертикали)

---

Свойство `align-items`

Значения свойства `align-items`:

`align-items: center`; - выравнивание элементов по центру

`align-items: flex-start`; - выравнивание элементов по верху

`align-items: flex-end`; - выравнивание элементов по низу

`align-items: stretch`; - выравнивание на всю высоту (по умолчанию)

`align-items: baseline`; - выравнивание элементов по базовой линии

Для родителя добавьте свойство **align-items**.  
Поэкспериментируйте с его значениями.

---

Установите значение свойства **align-items**, чтобы  
получить результат как на рисунке:

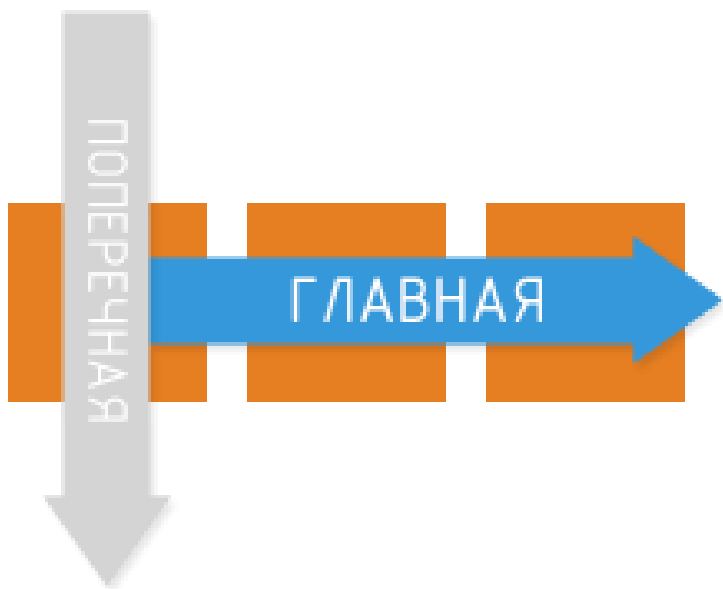
Lorem ipsum dolor sit amet  
consectetur adipisicing elit.  
Unde, quae fuga! Adipisci  
ducimus corporis fugiat ipsa  
distinctio, quasi sunt impedit  
eligendi optio a. Quae animi  
quibusdam, iusto laudantium  
architecto quas!

Lorem ipsum dolor sit amet  
consectetur adipisicing elit.  
Unde, quae fuga! Adipisci  
ducimus corporis fugiat ipsa  
distinctio, quasi sunt impedit  
eligendi optio a. Quae animi  
quibusdam, iusto laudantium  
architecto quas! Lorem ipsum  
dolor sit amet consectetur  
adipisicing elit. Unde, quae  
fuga! Adipisci ducimus  
corporis fugiat ipsa distinctio,  
quasi sunt impedit eligendi  
optio a. Quae animi  
quibusdam, iusto laudantium  
architecto quas!

Lorem ipsum dolor sit amet  
consectetur adipisicing elit.  
Unde, quae fuga! Adipisci  
ducimus corporis fugiat ipsa  
distinctio, quasi sunt impedit  
eligendi optio a. Quae animi  
quibusdam, iusto laudantium  
architecto quas!

### 3. Свойство Flex-direction

Свойство **flex-direction** указывает направление главной оси.



flex-direction: **row**



flex-direction: **column**

# 3. Свойство Flex-direction

---

Значения свойства **flex-direction**:

**flex-direction: row**; - элементы располагаются слева направо в строку (по умолчанию)

**flex-direction: row-reverse**; - элементы располагаются справа налево в строку

**flex-direction: column**; - элементы располагаются сверху вниз в столбец

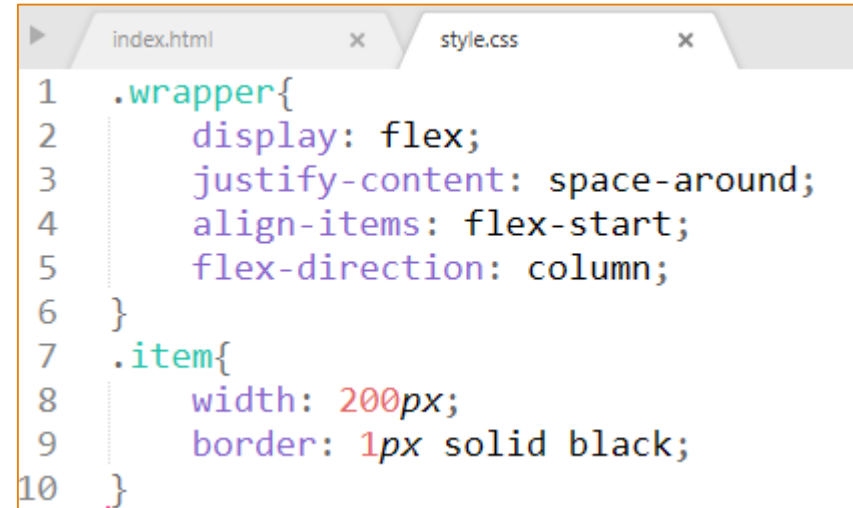
**flex-direction: column-reverse**; - элементы располагаются снизу вверх в столбец



Чтобы лучше видеть результат работы свойства **flex-direction** абзацы лучше пронумеровать.

```
<div class="wrapper">
  <div class="item">
    1. Lorem ipsum dolor sit amet consectetur adipisicing elit.
    Unde, quae fuga! Adipisci ducimus corporis fugiat ipsa distinctio,
    quasi sunt impedit eligendi optio a. Quae animi quibusdam,
    iusto laudantium architecto quas!
  </div>
  <div class="item">
    2. Lorem ipsum dolor sit amet consectetur adipisicing elit.
    Unde, quae fuga! Adipisci ducimus corporis fugiat ipsa distinctio,
    quasi sunt impedit eligendi optio a. Quae animi quibusdam,
    iusto laudantium architecto quas!
    Lorem ipsum dolor sit amet consectetur adipisicing elit.
    Unde, quae fuga! Adipisci ducimus corporis fugiat ipsa distinctio,
    quasi sunt impedit eligendi optio a. Quae animi quibusdam,
    iusto laudantium architecto quas!
  </div>
  <div class="item">
    3. Lorem ipsum dolor sit amet consectetur adipisicing elit.
    Unde, quae fuga! Adipisci ducimus corporis fugiat ipsa distinctio,
    quasi sunt impedit eligendi optio a. Quae animi quibusdam,
    iusto laudantium architecto quas!
  </div>
</div>
```

Для родителя добавьте свойство **flex-direction**, изменяя его значения.

A screenshot of a code editor with two tabs: 'index.html' and 'style.css'. The 'style.css' tab is active, showing CSS code for a flex container. The code is as follows:

```
1 .wrapper{
2     display: flex;
3     justify-content: space-around;
4     align-items: flex-start;
5     flex-direction: column;
6 }
7 .item{
8     width: 200px;
9     border: 1px solid black;
10 }
```

Чтобы изучать свойства flex дальше, установите свойство **flex-direction**, так чтобы элементы встали в одну линию.

## 4. Свойство Flex-wrap

---

Свойство **flex-wrap** указывает, должны ли элементы Flex переноситься на другую строку или нет.

Добавьте в контейнер **wrapper** еще несколько блоков **item** и попробуйте различные значения свойства **flex-wrap**:

**flex-wrap: wrap;** - элементы переносятся на новую строку

**flex-wrap: nowrap;** - элементы не переносятся на новую строку  
(по умолчанию)

## 5. Свойство Flex-flow

---

Свойство **flex-flow** является сокращенным свойством для задания свойств **flex-direction** и **flex-wrap**

Например, **flex-flow: row wrap;**

---

# Flex-свойства для элементов

# 1. Свойство Flex-grow

---

Свойство **flex-grow** указывает, насколько отдельный элемент будет больше соседних элементов (по умолчанию 0).

*Пример 1:*

- Если все flex-блоки внутри flex-контейнера имеют **flex-grow:1**, то они будут одинакового размера
- Если один из них имеет **flex-grow:2**, то он будет в 2 раза больше, чем все остальные

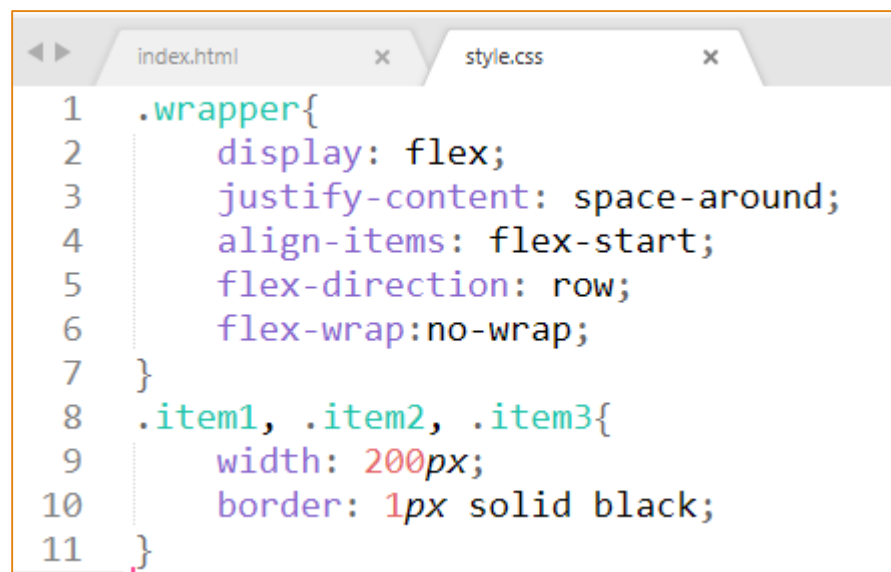
*Пример 2:*

- Если все flex-блоки внутри flex-контейнера имеют **flex-grow:3**, то они будут одинакового размера
- Если один из них имеет **flex-grow:12**, то он будет в 4 раза больше, чем все остальные

Оставьте в контейнере только 3 блока, задав им разные классы.

```
<div class="wrapper">
  <div class="item1">
    1. Lorem ipsum dolor sit amet consectetur adipisicing elit.
    Unde, quae fuga! Adipisci ducimus corporis fugiat ipsa distinctio,
    quasi sunt impedit eligendi optio a. Quae animi quibusdam,
    iusto laudantium architecto quas!
  </div>
  <div class="item2">
    2. Lorem ipsum dolor sit amet consectetur adipisicing elit.
    Unde, quae fuga! Adipisci ducimus corporis fugiat ipsa distinctio,
    quasi sunt impedit eligendi optio a. Quae animi quibusdam,
    iusto laudantium architecto quas!
  </div>
  <div class="item3">
    3. Lorem ipsum dolor sit amet consectetur adipisicing elit.
    Unde, quae fuga! Adipisci ducimus corporis fugiat ipsa distinctio,
    quasi sunt impedit eligendi optio a. Quae animi quibusdam,
    iusto laudantium architecto quas!
  </div>
</div>
```

Исправьте стили



```
index.html x style.css x
1 .wrapper{
2   display: flex;
3   justify-content: space-around;
4   align-items: flex-start;
5   flex-direction: row;
6   flex-wrap: no-wrap;
7 }
8 .item1, .item2, .item3{
9   width: 200px;
10  border: 1px solid black;
11 }
```

Добавьте для элементов контейнера различные значения свойства `flex-grow`.

---

1 пример

```
12 .item3{
13     flex-grow: 3;
14 }
```

2 пример

```
12 .item1{
13     flex-grow: 1;
14 }
15 .item2{
16     flex-grow: 2;
17 }
18 .item3{
19     flex-grow: 2;
20 }
```

3 пример

```
12 .item1{
13     flex-grow: 8;
14 }
15 .item2{
16     flex-grow: 1;
17 }
18 .item3{
19     flex-grow: 2;
20 }
```

## 2. Свойство Flex-shrink

---

Определяет, насколько **блок** будет уменьшаться относительно соседних элементов внутри flex-контейнера в случае недостатка свободного места.  
По умолчанию равен 1.

Принцип действия у данного свойства такой же, как у свойства flex-grow.



### 3. Свойство Flex-basis

---

Базовый размер отдельного элемента, заменяет свойство width

### 4. Свойство Flex

Свойство **flex** является сокращенным свойством для задания свойств **flex-grow**, **flex-shrink** и **flex-basis**

---

## 5. Свойство *Align-self*

Выравнивание отдельно взятого flex-блока по поперечной оси.  
Значения такие же как у *align-items*.

## 6. Свойство Order

---

Порядок следования отдельно взятого flex-блока внутри flex-контейнера. Задается целым числом и по умолчанию равно 0.

```
12  .item1{
13      flex-grow: 2;
14      order:1;
15  }
16  .item2{
17      order:2;
18  }
19  .item3{
20      order:0;
21  }
```

## 7. margin: auto – выравнивание блока по вертикали по центру

---

```
8  .item1, .item2, .item3{
9      flex-basis: 200px;
10     border: 1px solid black;
11 }
12 .item1{
13     flex-grow: 2;
14     order:1;
15     margin: auto;
16 }
17 .item2{
18     order:2;
19 }
20 .item3{
21     order:0;
22 }
```