



#ШПАРГАЛОЧКИ

# СОЗДАНИЕ САЙТОВ FRONT-END РАЗРАБОТКА

## Начальный уровень

Материалы подготовлены отделом методической разработки

Больше полезных материалов и общения в нашем комьюнити  
в Telegram: [https://t.me/hw\\_school](https://t.me/hw_school)





# Flexbox: создаем боковое меню сайта об экологии



# object-fit

**object-fit** - CSS-свойство, позволяющее выбрать, как картинка должна быть изменена, чтобы поместиться в тег `<img>`, если его размеры отличаются от размеров изображения.

Например, чтобы отобразить картинку размером 500x350 в теге `<img>` шириной 300px и высотой 200px, ее придется сжать или обрезать.

**object-fit** принимает следующие значения:

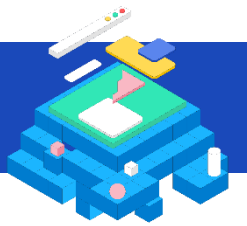
**fill** - картинка подгоняется под размер, может быть сжата или растянута, заполняет все поле

**contain** - картинка будет вписана в `<img>`, пропорции сохранятся (может заполнить не все поле)

**cover** - картинка заполнит все поле, пропорции сохранятся (при необходимости обрежется)

**none** - без изменений

**scale-down** - совпадает с `none` или `contain` (со значением, дающим меньший размер)



# object-position

**object-position** - второе свойство для работы с содержимым тега `<img>`. Если после применения `object-fit` видимая часть картинки обрезалась и показывается не та часть, которая нужна, `object-position` поможет исправить ситуацию.

Это свойство похоже на `background-position` и тоже принимает часть изображения, которую нужно показать (**left**, **right**, **top**, **bottom**, **center**) или сдвиг по X и Y, например в пикселях или процентах:

**object-position: left bottom;**

**object-position: 20% 50%;**



# Flexbox

**flexbox** - это значение свойства **display**, которое размещает **дочерние элементы** выбранного блока (его называют **flex-контейнером**) относительно двух осей - основной (направленной слева направо) и перекрестной (направленной сверху вниз).

По умолчанию главной осью является именно горизонтальная и элементы располагаются вдоль нее.

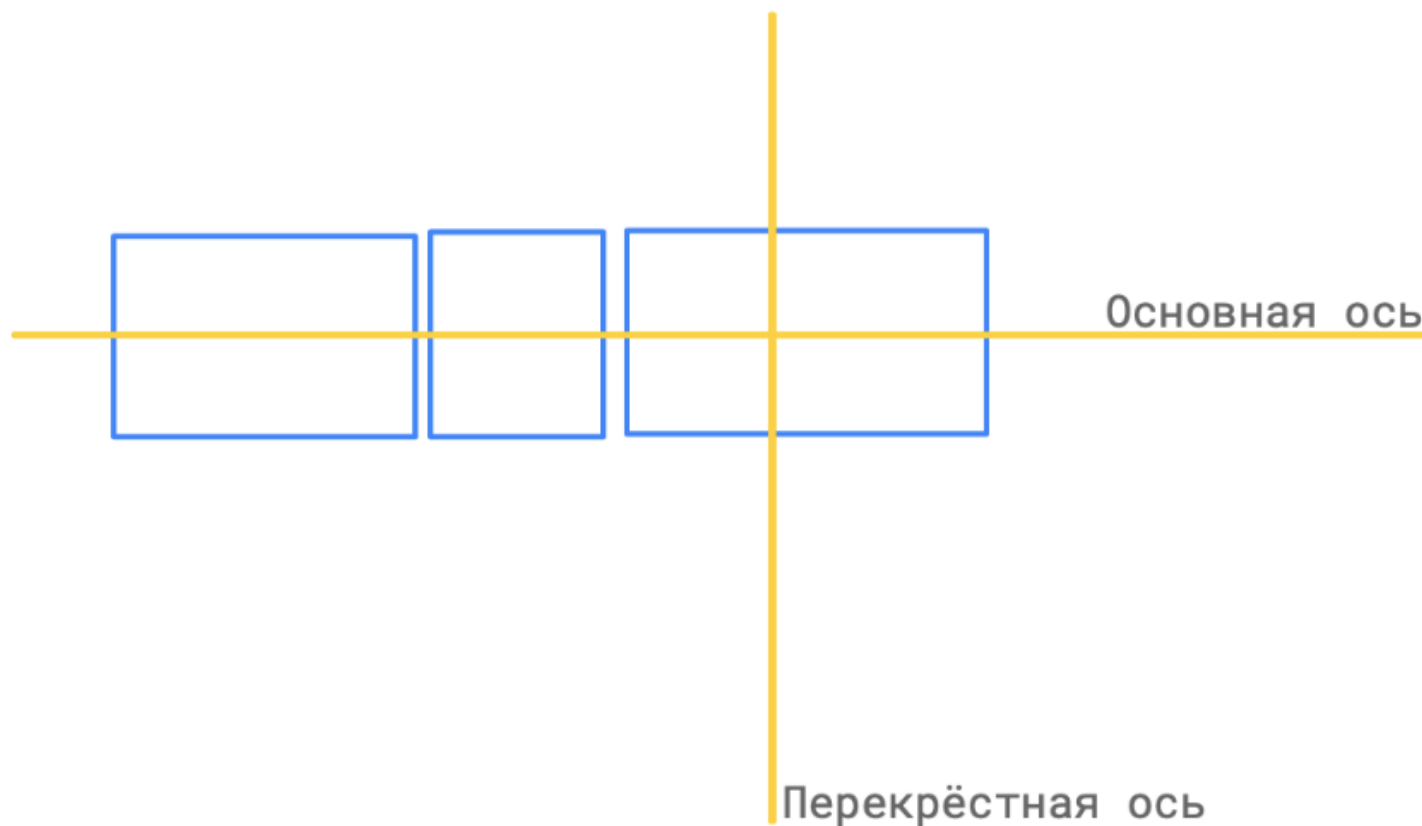
```
.menu {
```

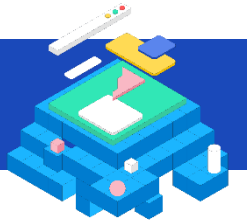
```
    display: flex; /* теперь меню - flex-контейнер */
```

```
}
```



# Flexbox, основная и перекрестная оси





# flex-direction

**flex-direction** - свойство, задающее направление главной оси. Указывается у flex-контейнера и имеет 4 значения:

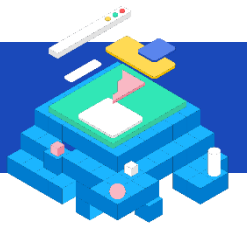
**row** - главная ось направлена слева направо

**column** - главная ось направлена сверху вниз

**row-reverse** - главная ось направлена справа налево

**column-reverse** - главная ось направлена снизу вверх

```
.menu {  
  
  display: flex; /* теперь меню - flex-контейнер */  
  
  flex-direction: column; /* теперь дочерние элементы выстроены сверху вниз */  
  
}
```



# align-items

**align-items** - свойство, помогающее выровнять элементы относительно **перекрестной оси**. Также указывается у flex-контейнера и принимает следующие значения:

**flex-start** - элементы прижаты к началу перекрестной оси

**flex-end** - элементы прижаты к концу перекрестной оси

**center** - элементы расположены по центру перекрестной оси

**baseline** - элементы выравниваются по базовой линии текста внутри них

**stretch** - элементы растягиваются по всей длине перекрестной линии





# justify-content

**justify-content** - свойство, задающее распределение элементов вдоль главной оси контейнера. Важно, что прежде чем задавать это свойство, элементам нужно задать необходимый размер и отступы.

**justify-content** задается flex-контейнеру и принимает следующие значения:

**flex-start** - элементы прижимаются к началу главной оси

**flex-end** - элементы прижимаются к концу главной оси

**center** - элементы встают рядом в центре главной оси

**space-between** - крайние элементы прижаты к краям контейнера, остальные распределяются равномерно

**space-around** - элементы равномерно распределяются вдоль главной оси, расстояние между крайними элементами и краем контейнера равно половине расстояния между соседними элементами

**space-evenly** - элементы равномерно распределяются вдоль главной оси, расстояние между соседними элементами и краями одинаковое