CSS: IMAGES, FLEXBOX

AIT
Hi-tech team

НАШИ ПРАВИЛА

Включенная камера

Вопросы по поднятой руке

Не перебиваем друг друга

Все вопросы, не связанные с тематикой курса (орг-вопросы и т. д.), должны быть направлены куратору

Подготовьте свое рабочее окружение для возможной демонстрации экрана (закройте лишние соцсети и прочие приложения)

Повторим;)

Какие значения есть у свойства position

Какое значение свойства position отвечает за его фиксацию относительно экрана

Какое свойство помогает нам указать порядок позиционированных элементов по слоям

ЦЕЛЬ

Поработать с изображениями в качестве бэкграунда. Изучить концепцию Flexbox. Узнать какие свойства отвечают за выравнивание элементов.

ПЛАН ЗАНЯТИЯ

1. Работа с изображениями
2. Определение Flexbox
3. Главная и поперечная оси
4. Свойства

Параметры фона

```
background-size: no-repeat;
background-position: center;
background-size: cover;
width: 400px;
```

Фоновый цвет

Свойство `background-color` определяет цвет фона для любого элемента в CSS. Свойство принимает любой допустимый цвет.

```
.box {
  background-color: #BCBCBC;
}
.box {
  background-color: grey;
}
.box {
  background-color: rgba(188,188,188);
}
```

Фоновое изображение

Свойство **`background-image`** позволяет отображать изображение в качестве фона элемента. В приведённом ниже примере у нас есть два блока — в одном фоновое изображение больше, чем размеры блока, а в другом - маленькое изображение звезды.

Этот пример демонстрирует две особенности фоновых изображений. По умолчанию большое изображение не масштабируется до размера блока, поэтому мы видим только его небольшой угол, в то время как маленькое изображение повторяется, чтобы заполнить весь блок. В нашем случае фактически было использовано изображение одной маленькой звезды.

Фоновое изображение

HTML

```
  <div class="first_box">
</div>
  <div class="second_box">
</div>
```

CSS

```
.first_box {

  background-image:
url(balloons.jpg);

}

.second_box {

  background-image: url(star.png);

}
```

Примечание: Если кроме фонового изображения вы добавили фоновый цвет, то изображение будет отображаться над цветом.

Свойство background-repeat

Свойство **`background-repeat`** используется для управления повторениями фонового изображения. Доступные значения:

- `no-repeat` — останавливает повторение фонового изображения во всех направлениях.
- `repeat-x` — повторение фонового изображения по горизонтали.
- `repeat-y` — повторение фонового изображения по вертикали.
- `repeat` — повторение фонового изображения в обоих направлениях. Установлено по умолчанию.

Свойство background-repeat

Используем свойство `background-repeat` для работы с изображением звёздочкой из предыдущего примера

CSS

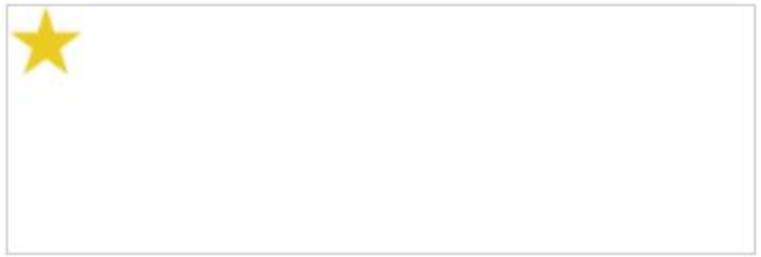
```
.second_box {

  background-image:
url(star.png);

  background-repeat: no-repeat;

}
```

Итог



Изменение размеров фонового изображения

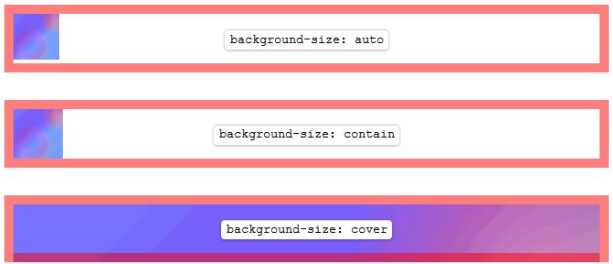
Для более корректного отображения больших картинок для фона мы можем использовать использовать свойство `background-size,` которое может принимать значения длины или в процентах, чтобы размер изображения соответствовал размеру фона.

```
background-size: 30%;
```

Изменение размеров фонового изображения

Для свойства `background-size` можно использовать не только показатели ширины, но и ключевые слова:

- `cover` — браузер сделает изображение достаточно большим, чтобы оно полностью заполнило блок, сохраняя при этом соотношение сторон. В этом случае часть изображения, скорее всего, окажется за пределами блока.
- `contain` — браузер сделает изображение нужного размера, чтобы поместиться в блоке. В этом случае могут появиться пробелы с обеих сторон или сверху и снизу изображения, если соотношение сторон изображения отличается от соотношения сторон блока.



Позиционирование фонового изображения

Свойство `background-position` позволяет вам изменять позицию, в которой фоновое изображение появляется в блоке. При этом используется система координат, в которой левый верхний угол блока равен (0,0), а сам блок располагается вдоль горизонтальной (x) и вертикальной (y) осей.

```
background-position: top;
```

```
background-position: left;
```

```
background-position: center;
```

```
background-position: 25% 75%;
```

Примечание: По умолчанию значение `background-position` равно (0,0).

Несколько фоновых изображений

Возможно создавать несколько фоновых изображений — просто разделив значения свойства `background-image` запятыми.

Когда вы сделаете это, произойдёт наложение фоновых изображений друг на друга. Фоновые изображения будут наложены слоями, где каждое новое фоновое изображение, перечисленное в коде, будет накладываться поверх ранее указанного изображения.

```
background-image: url(image1.png), url(image2.png), url(image3.png),
url(image1.png);
```

Сокращённое свойство background

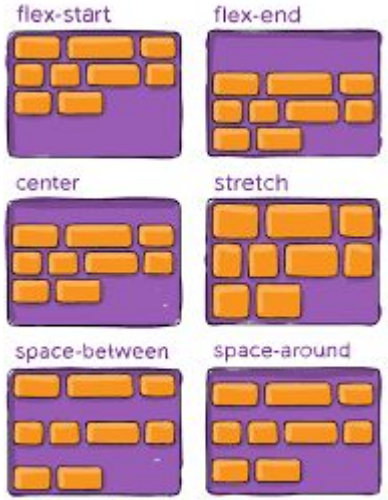
Свойство background объединяет в себе все свойства, которые используются для определения фона страницы.

Свойства, которые можно задать (по порядку): `background-color`, `background-image`, `background-repeat`, `background-position`.

Flexbox

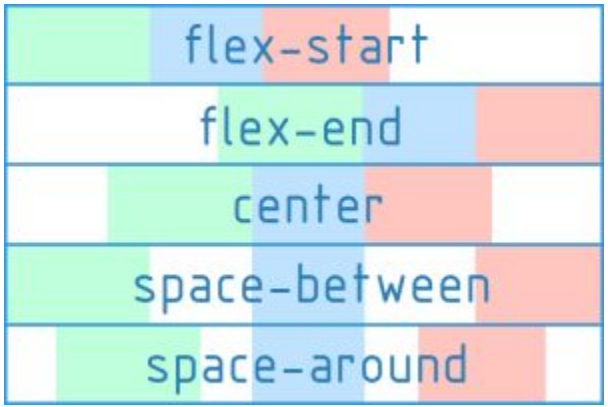
Flexbox

Flexbox - предоставляет инструменты для быстрого создания сложных, гибких макетов

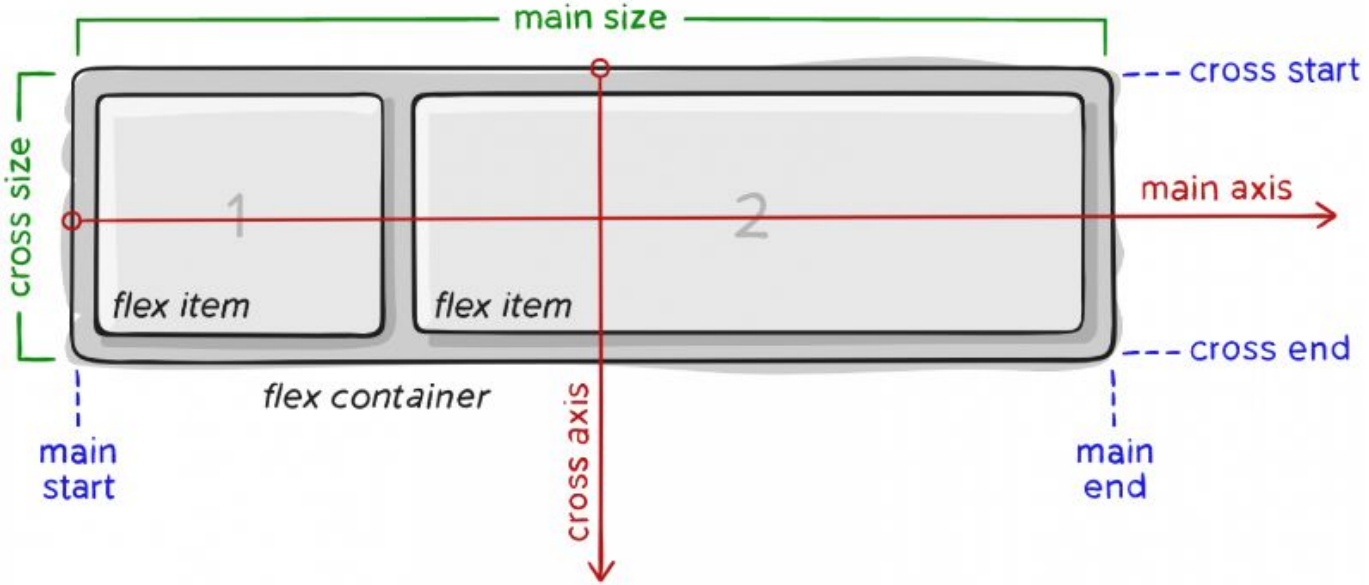


display:flex

Элемент становится контейнером для гибких (flex) элементов. Это позволяет управлять распределением пространства в контейнере и создавать адаптивные макеты.



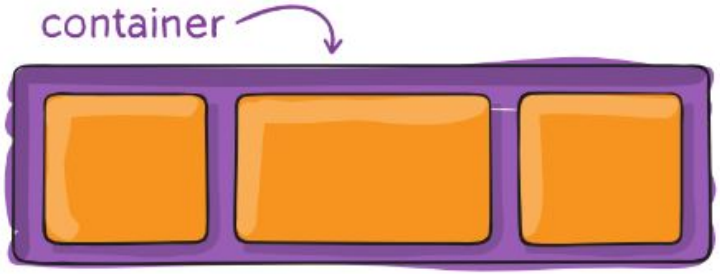
Если «обычная» компоновка основана как на блочном, так и на inline направлениях, то flex layout основана на «направлениях flex-flow»



Элементы будут расположены либо в направлении главной оси (main axis от main-start до main-end) или в направлении поперечной оси (cross axis от cross-start до cross-end).

- **main axis** — главная ось flex контейнера — это основная ось, вдоль которой располагаются flex элементы. Будьте внимательны, эта ось не обязательно горизонтальная; это зависит от flex-direction свойства (см. ниже).
- **main-start | main-end** — flex элементы помещаются в контейнер, начиная с main-start и заканчивая main-end.
- **main size** — ширина или высота flex элемента, в зависимости от того, что находится в основном измерении. Определяется основным размером flex элементов т.е. свойством 'width' или 'height', в зависимости от того, что находится в основном измерении.
- **cross axis** — ось перпендикулярная главной оси, называется поперечной осью. Её направление зависит от направления главной оси.
- **cross-start | cross-end** — flex строки заполняются элементами и помещаются в контейнер, начиная от cross-start flex контейнера по направлению к cross-end.
- **cross size** — ширина или высота flex элемента. В зависимости от css свойства flex-direction, это ширина или высота элемента. Это всегда поперечный размер flex элементов.

Свойства для Родителя (flex контейнер)



display

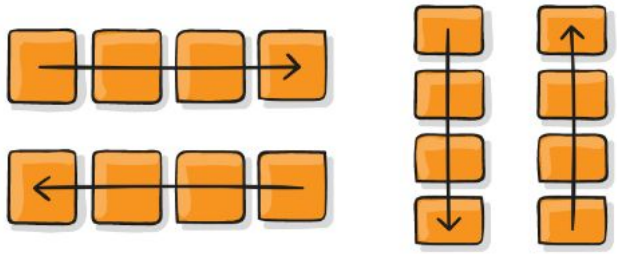
Определяет flex контейнер. Включает flex контекст для всех потомков первого уровня.

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

Свойства для Родителя (flex контейнер)

flex-direction

Устанавливает основную ось, таким образом определяя направление flex элементов, помещаемых в flex контейнер.



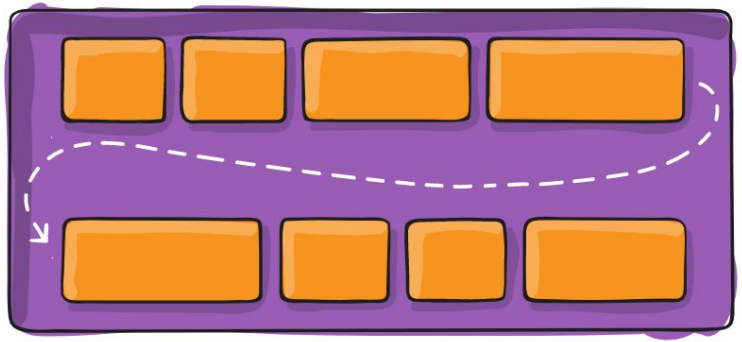
- **row** (по умолчанию): слева направо в ltr; справа налево в rtl
- **row-reverse** справа налево ltr; слева направо в rtl
- **column**: так же, как и row но сверху вниз
- **column-reverse:** то же самое, row-reverse но снизу вверх

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

Свойства для Родителя (flex контейнер)

flex-wrap

По умолчанию гибкие элементы будут пытаться уместиться на одной строке. Вы можете изменить это и позволить элементам переходить на новую строку по мере необходимости с помощью этого свойства.



- **nowrap** (по умолчанию): все flex элементы будут в одной строке
- **wrap**: flex-элементы будут перенесены на несколько строк сверху вниз.
- **wrap-reverse**: flex-элементы будут перенесены на несколько строк снизу вверх.

```
.container{
  flex-wrap: nowrap | wrap | wrap-reverse;
}
```

Свойства для Родителя (flex контейнер)

flex-flow

Это сокращение для **flex-direction** и **flex-wrap** свойств, которые вместе определяют основные и поперечные оси flex контейнера. Значением по умолчанию является row nowrap.

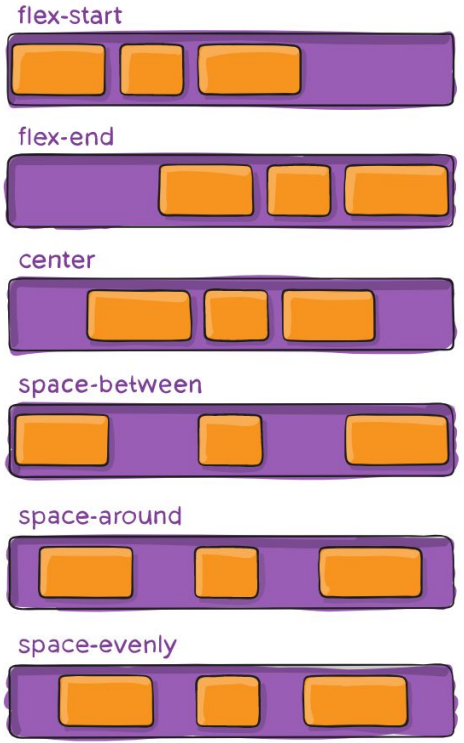
```
flex-flow: <‘flex-direction’> || <‘flex-wrap’>
```

Свойства для Родителя

justify-content

Это свойство определяет выравнивание вдоль главной оси.

- **flex-start** (по умолчанию): элементы сдвинуты в начало flex-direction.
- **flex-end:** элементы сдвинуты ближе к концу flex направления.
- **center:** элементы центрированы вдоль линии
- **space-between:** элементы равномерно распределены по линии; первый элемент находится в начале строки, последний элемент в конце строки
- **space-around:** элементы равномерно распределены по линии с одинаковым пространством вокруг них.
- **space-evenly:** элементы распределяются таким образом, чтобы расстояние между любыми двумя элементами (и расстояние до краев) было одинаковым.



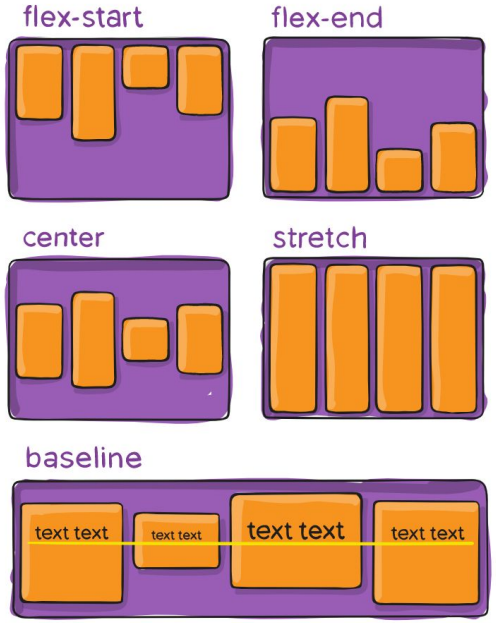

```
.container {
  justify-content: flex-start | flex-end | center | space-between | space-around
}
```

Свойства для Родителя

align-items

Это свойство определяет поведение по умолчанию того, как flex элементы располагаются вдоль поперечной оси на текущей линии.

- **stretch** (по умолчанию): растягивать, чтобы заполнить контейнер (все еще соблюдаются min-width / max-width)
- **flex-start:** элементы размещаются в начале поперечной оси.
- **flex-end:** элементы располагаются в конце поперечной оси. Разница опять-таки тонкая и заключается в соблюдении flex-direction или writing-mode правил.
- **center**: элементы центрированы по поперечной оси
- **baseline**: элементы выровнены, по их базовой линии




```
.container {
  align-items: stretch | flex-start | flex-end | center | baseline
}
```

Ваша новая IT-профессия – Ваш новый уровень жизни

Программирование с нуля в
немецкой школе AIT TR GmbH

AIT
Hi-tech team