

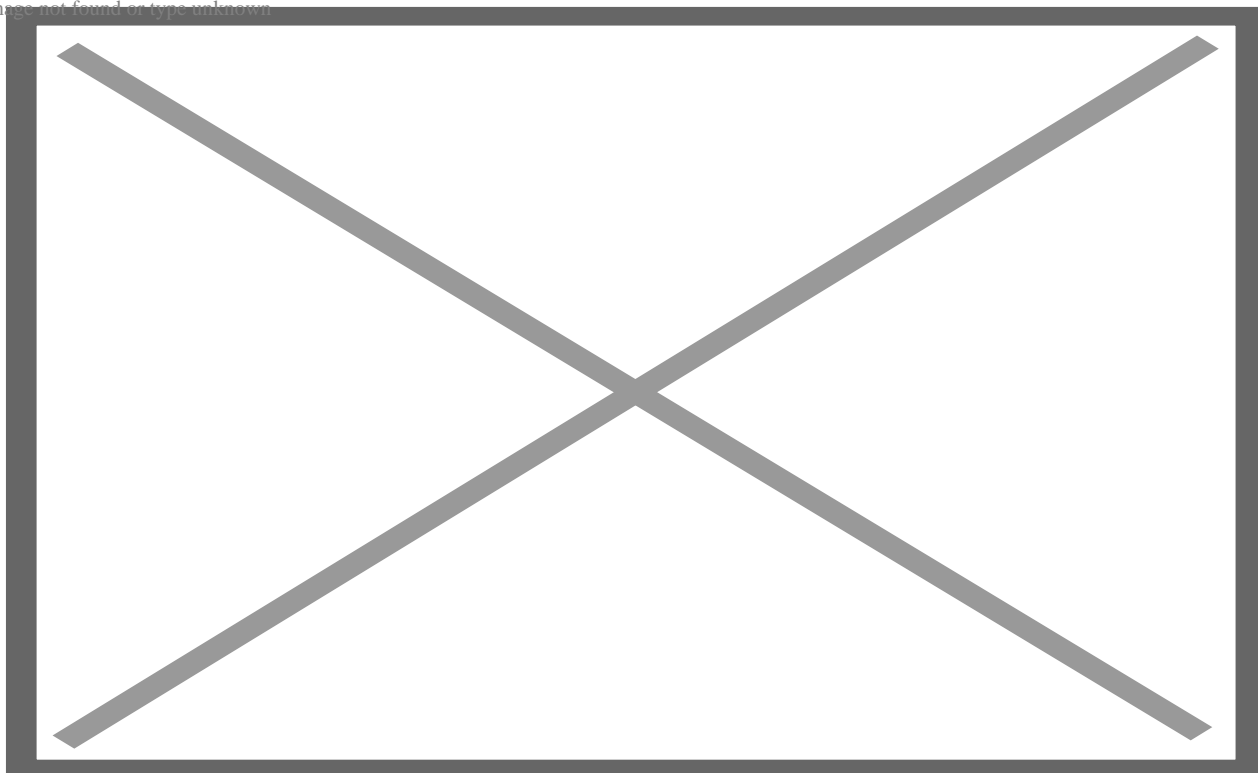


## CSS Flexbox: Полное руководство

### Описание

Если вы когда-нибудь пытались создать макет с помощью CSS, вы знаете, что это может быть нелегкой задачей – добиться правильного расположения всех элементов. Именно здесь на помощь приходит свойство CSS flex.

Image not found or type unknown



Представленный в CSS3, модуль разметки flexbox обеспечивает более эффективный способ расположения элементов на странице. Но что такое flexbox и как его

использовать, чтобы сделать веб-дизайн более отзывчивым и динамичным? В этом посте мы рассмотрим основы свойства CSS flex и изучим несколько примеров, демонстрирующих его использование.

## Что такое свойство CSS Flex?

Свойство flex – это универсальная команда, позволяющая разработчикам создавать гибкие и адаптивные макеты для веб-страниц. Это свойство позволяет изменять размеры и расположение элементов веб-страницы относительно друг друга, а не указывать точные размеры.

Свойство flex фактически является сокращением для нескольких других свойств flex:

- flex-grow
- flex-shrink
- flex-basis

Они определяют длину элемента и то, будет ли он увеличиваться или уменьшаться в зависимости от других элементов в контейнере. Свойство flex особенно полезно для отзывчивого дизайна, поскольку оно позволяет масштабировать и переставлять элементы в зависимости от размера экрана. Например, ряд изображений на экране настольного компьютера можно легко перестроить в колонку на мобильном устройстве, просто изменив свойство flex-direction со строки на колонку.

### ДОБАВИТЬ ИЗОБРАЖЕНИЕ/ВИЗУАЛ ЗДЕСЬ

Еще одним ключевым преимуществом flex является его способность упростить и сократить количество кода, необходимого для создания отзывчивых макетов. С помощью flex можно создавать сложные многоколоночные дизайны, используя меньше строк кода, чем традиционные техники, такие как плавающие элементы и позиционирование.

## Синтаксис CSS Flex

Свойство flex работает путем установки родительского элемента в **display: flex**, что дает вам доступ к ряду других гибких свойств макета. Два наиболее популярных свойства – **justify-content** и **align-items**. **Justify-content** определяет, как элементы внутри flex-контейнера располагаются вдоль главной оси, а **align-items**

управляет их положением вдоль поперечной оси. Давайте рассмотрим несколько примеров этого синтаксиса. Вот основной синтаксис ключевых слов для свойства CSS flex:

```
flex: auto;
```

```
flex: initial;
```

```
flex: none;
```

Вы также можете определить свойство flex с помощью процентов, эм и пикселей. Существует также несколько глобальных значений, которые вы можете использовать с этим свойством:

- Inherit
- Unset
- Revert

Давайте рассмотрим подробнее, что делает каждое из этих значений.

## Значения свойства CSS Flex

### Auto

Это значение устанавливает flex-basis в auto, что означает, что размер элемента будет определяться его содержимым и может увеличиваться или уменьшаться по мере необходимости.

#### Пример:

В этом примере элементу будет задан размер по умолчанию, основанный на его содержимом, и он сможет увеличиваться или уменьшаться по мере необходимости.

```
.item {  
    flex: 1 1 auto;  
  
}
```

### Initial

---

Это значение устанавливает свойства гибкого элемента в значение по умолчанию или начальное значение.

**Пример:**

```
.item {  
    flex: initial;  
  
}
```

В этом примере элемент будет иметь flex-basis 0, и ему будет разрешено уменьшаться или увеличиваться на одинаковый процент.

**None**

Это значение устанавливает гибкий контейнер элемента в 0, что означает, что он не будет ни увеличиваться, ни уменьшаться при наличии дополнительного пространства или его недостатке.

**Пример:**

```
item {  
    flex: none;  
  
}
```

В данном примере элемент будет иметь фиксированный, негибкий размер.

**Inherit**

Это значение устанавливает, что гибкие свойства элемента наследуются от его родительского свойства.

**Пример:**

```
item {  
    flex: inherit;  
  
}
```

В этом примере элемент будет иметь те же гибкие свойства, что и его родительское свойство.

## Unset

Это значение устанавливает гибкие свойства элемента в значение по умолчанию, если они не были установлены ранее; в противном случае устанавливается унаследованное значение.

### Пример:

```
.item {  
    flex: unset;  
  
}
```

В этом примере элемент наследует свойства flex или возвращается к значению по умолчанию, если оно не задано явно.

## Revert

Это значение устанавливает гибкие свойства элемента в значения по умолчанию, как если бы значение никогда не было установлено.

### Пример:

```
item {  
    flex: revert;  
  
}
```

В этом примере свойства flexbox элемента вернутся к своим первоначальным значениям по умолчанию.

## Свойства CSS Flex родительских и дочерних элементов

Ниже приведены различные свойства CSS flex для родительских и дочерних элементов. Давайте рассмотрим, что делает каждое из них, а затем посмотрим пример их использования в действии.

### Свойства родительского элемента CSS Flex

#### Display

Это свойство определяет тип контейнера, который вы хотите создать. По умолчанию используется значение block. Чтобы включить flexbox, установите свойство display для элемента контейнера в flex или inline-flex, в зависимости от потребностей макета.

#### Align-items

Это свойство контролирует, как элементы внутри flex-контейнера выравниваются по поперечной оси (оси, перпендикулярной направлению главной оси). Оно позволяет выравнивать элементы в начале, в центре, в конце или растягивать их по поперечной оси.

#### Flex-Flow

Это свойство является сокращением двух других свойств, flex-direction и flex-wrap. Оно определяет направление, в котором будут двигаться гибкие элементы, а также то, должны ли они заворачиваться или нет. Значение по умолчанию – row nowrap, что означает, что элементы идут горизонтально в одну линию без обертывания.

#### Flex-Wrap

Это свойство определяет, должны ли гибкие элементы обертываться или нет, когда они превышают ширину гибкого контейнера. Вы можете установить значение nowrap (по умолчанию), wrap или wrap-reverse, в зависимости от ваших потребностей в верстке.

## **Justify-Content**

Это свойство управляет выравниванием элементов внутри гибкого контейнера по главной оси (оси, по которой движется flex-направление). Оно позволяет выравнивать элементы в начале, в центре, в конце, между, вокруг или равномерно по главной оси.

## **Свойства CSS Flex Child**

### **Order**

Это свойство позволяет определить порядок элементов flex, контролируя порядок расположения независимо от исходного порядка в HTML-разметке.

### **Flex-Grow**

Это свойство задает, насколько должен вырасти гибкий элемент относительно других гибких элементов, если вдоль главной оси есть неиспользованное пространство. Чем больше значение, тем больше будет пропорционально увеличиваться элемент.

### **Flex-Shrink**

Это свойство устанавливает, насколько сильно должен уменьшаться гибкий элемент относительно других гибких элементов, если вдоль главной оси недостаточно места. The larger the value, the smaller the item will proportionally shrink.

### **Flex-Basis**

Это свойство задает начальный размер гибкого элемента вдоль главной оси до распределения оставшегося пространства. Оно может быть установлено на конкретное значение (например, пиксели) или на ключевое слово (например, auto,

которое использует размер элемента по умолчанию). Теперь, когда мы рассмотрели наиболее популярные свойства CSS flex и их потенциальные значения, давайте посмотрим, как они применяются в действии на примере.

## Пример гибкого свойства CSS

Допустим, у нас есть контейнер `<div>` с тремя дочерними элементами `<div>`, каждый из которых представляет элемент. Если мы добавим свойство **display: flex** для контейнера, он включит flexbox-макет для своих дочерних элементов. Присвоение свойства **flex: 1** дочерним элементам позволяет им расти и занимать одинаковое пространство внутри контейнера. Чтобы элементы располагались равномерно, можно добавить в таблицу стилей свойство **justify-content: space-between**. Наконец, добавьте некоторые другие свойства стиля, такие как padding, цвет фона и границы, для более сложного визуального представления.

Вот ваш конечный код:

```
// HTML ???

<div class="container">
  <div class="item">Item 1</div>
  <div class="item">Item 2</div>
  <div class="item">Item 3</div>

</div>

// CSS ???

.container {
  display: flex;
  justify-content: space-between;
}

.item {
  flex: 1;
  padding: 10px;
  background-color: lightblue;
  border: 1px solid blue;
}
```

Результат в браузере:



Item 1

Item 2

Item 3

В объяснении выше мы упомянули, что это создало “макет flexbox”. Давайте подробнее поговорим о том, что это такое, в следующем разделе.

## CSS Flexbox, объяснение

CSS flexbox – это модуль верстки, который создает сложные многоколоночные дизайны, используя очень мало кода. В отличие от сетки CSS, которая является двухмерной системой компоновки, flexbox – это одномерная система, которая фокусируется на расположении элементов вдоль одной оси (горизонтально или вертикально). Одним из ключевых преимуществ CSS flexbox является то, что он позволяет упростить код и сделать макеты более масштабируемыми. Вы можете создавать макеты, которые адаптируются к различным размерам экранов и устройствам, гарантируя, что ваш сайт будет выглядеть великолепно независимо от того, как к нему обращаются пользователи. Это может быть особенно важно для мобильных пользователей, которые могут заходить на ваш сайт с различных устройств и экранов.

### Как использовать CSS Flexbox

Для начала установите для контейнера `display: flex`. Теперь вы можете использовать различные свойства, такие как `flex-wrap` и `flex-basis`, чтобы определить, как должны быть расположены дочерние элементы. Это позволит вам

легко управлять позиционированием и размерами элементов на вашей странице, не прибегая к сложным вычислениям или корректировкам. Давайте рассмотрим пример.

## Пример CSS Flexbox

Предположим, вы хотите создать отзывчивый двухколоночный макет для главной страницы вашего сайта. Одна колонка будет содержать заголовок и вводный текст, а другая – контактную форму. Для начала вам нужно создать контейнер, который будет содержать две колонки. Это можно сделать, добавив следующий CSS:

```
container {  
    display: flex;  
    flex-wrap: wrap;  
}
```

Свойство **display: flex** задает вашему контейнеру вид элемента flexbox. Свойство **flex-wrap** указывает контейнеру, что его дочерние элементы должны быть развернуты на следующую строку, если на первой строке для них недостаточно места. Теперь создайте два дочерних элемента, по одному для каждой колонки. Это можно сделать с помощью следующего кода:

```
.column {  
    flex-basis: 50%;  
    padding: 20px;  
    box-sizing: border-box;  
}
```

Здесь свойство **flex-basis** устанавливает ширину каждой колонки равной 50% от ширины контейнера, поэтому они будут одинакового размера. Свойства **padding** и **box-sizing** используются для добавления подкладок к столбцам без изменения их общей ширины. Теперь, когда вы настроили контейнер и колонки, вы можете начать добавлять содержимое. Вот пример того, как может выглядеть ваш HTML:

```
// HTML ???  
  
<div class="container">  
  <div class="column">
```

```
<h2>Welcome to our website!</h2>
<p>Thank you for visiting us. Lorem ipsum dolor sit amet, consectetur adipiscing
</div>
<div class="column">
  <h3>Contact Us</h3>
  <form>
    <label for="name">Name:</label>
    <input type="text" id="name" name="name">
    <label for="email">Email:</label>
    <input type="email" id="email" name="email">
    <label for="message">Message:</label>
    <textarea id="message" name="message"></textarea>
    <input type="submit" value="Send">
  </form>
</div>
</div>

// CSS ???

.container {
  display: flex;
  flex-wrap: wrap;
}

.column {
  flex-basis: 50%;
  padding: 20px;
  box-sizing: border-box;
}
}
```

Результат в браузере:

Welcome to our website!

Thank you for visiting us. Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua.

Contact Us

Name:  Email:  Message:

Вот и все. С помощью всего нескольких строк CSS и простого HTML вы можете создать отзывчивый двухколоночный макет с использованием CSS flexbox. Вы можете дополнительно настроить свой макет с помощью вышеупомянутых свойств для точной настройки поведения контейнера flexbox и его дочерних элементов.

## **Использование свойства CSS Flex**

Свойство CSS flex – это мощный инструмент для создания универсальных и отзывчивых макетов в ваших веб-проектах. Благодаря обширному набору свойств и опций, модуль flexbox позволяет дизайнерам и разработчикам создавать сложные и динамичные макеты, которые могут адаптироваться к различным размерам экрана, устройствам и взаимодействию с пользователем. Независимо от того, создаете ли вы простую целевую страницу или продвинутое веб-приложение, освоение основ свойства CSS flex – это важный шаг на пути к созданию более эффективных и действенных веб-проектов.

### **Дата Создания**

09.06.2023