

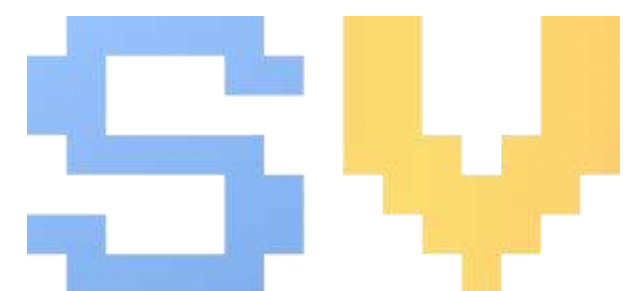
# CSS Fundamentals



SILICON VALLEY

# Что мы пройдем?

- Introduction to CSS
- Inline, Internal, and External CSS
- More CSS and Combining Selectors
- Class and ID Selectors
- Working With Colors
- Styling Hyperlinks and Pseudo-classes
- Using Chrome DevTools
- Positioning
- The CSS Box Model
- A Flexbox Overview
- A CSS Grid Overview



# Introduction to CSS

CSS (Cascading Style Sheets) - это язык для описания внешнего вида документов, написанных с использованием языка разметки, такого как HTML или XML. С помощью CSS вы можете определить стили для текста, цвета, размеров, отступов и многого другого, которые будут применяться к элементам вашего документа.

CSS позволяет вам отделить оформление документа от его содержания, что делает его более гибким и легко поддающимся изменению. Это означает, что вы можете изменять внешний вид сайта без изменения самого содержимого.

CSS также позволяет вам использовать один и тот же стиль для нескольких элементов на странице, что делает его более эффективным и легко поддающимся обслуживанию.

В целом, CSS позволяет вам создавать красивые и профессиональные веб-сайты, которые легко обслуживать и масштабировать. Если вы хотите научиться использовать CSS, рекомендуется изучить документацию и примеры кода, а также практиковать создание стилей для различных типов элементов и структур. Существует множество ресурсов и онлайн-курсов, которые могут помочь вам научиться использовать CSS. Важно понимать, что изучение CSS является необходимым шагом для любого, кто хочет создавать современные веб-сайты.



# Introduction to CSS

Ознакомьтесь со следующими материалами:

<https://htmlacademy.ru/> — онлайн-платформа, предлагающая интерактивные учебные пособия и упражнения по HTML и CSS.

<https://learn.javascript.ru/> — веб-сайт, предоставляющий исчерпывающее руководство по веб-разработке, включая HTML, CSS, JavaScript и другие веб-технологии.

<https://webdesign.ru.net/> — веб-сайт, предлагающий учебные пособия, статьи и советы по веб-разработке, включая HTML и CSS.

<https://www.htmlbook.ru/css/> — сайт с подробным справочным руководством по CSS, а также руководствами и примерами.

<https://developer.mozilla.org/ru/docs/Web/CSS> — русскоязычная версия сайта Mozilla Developer Network, где вы можете найти документацию и учебные пособия по CSS и другим веб-технологиям.

<https://css-tricks.com/> — веб-сайт с советами, рекомендациями и приемами работы с CSS, а также множеством руководств и статей.

<https://www.w3schools.com/css/> — веб-сайт, предлагающий учебные пособия, справочники и примеры по CSS и другим веб-технологиям.



# Inline, Internal, and External CSS

Inline CSS - это способ применения стилей к конкретному элементу HTML, используя атрибут "style" в теге элемента.

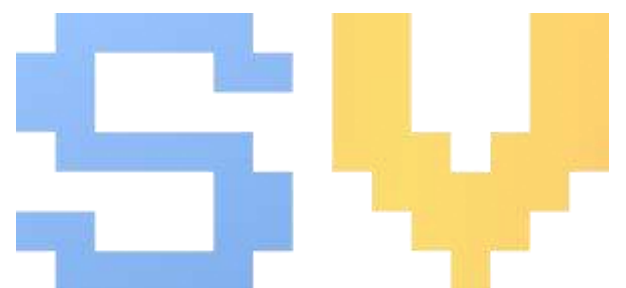
```
<p style="color: red;">Этот текст будет красным</p>
```

Internal CSS - это способ применения стилей, когда стили определяются внутри тега "style" на странице HTML.

```
<head>
<style>
  p {
    color: red;
  }
</style>
</head>
<body>
  <p>Этот текст будет красным</p>
</body>
```

External CSS - это способ применения стилей, когда стили определяются в отдельном файле CSS, который подключается к странице HTML с помощью тега "link". Этот способ позволяет удобно управлять стилями на нескольких страницах сайта.

```
<head>
<link rel="stylesheet" type="text/css" href="styles.css">
</head>
<body>
  <p>Этот текст будет красным</p>
</body>
в файле styles.css:
p {
  color: red;
}
```





# Inline, Internal, and External CSS

Создайте HTML-документ, который отображает информацию о вашем любимом городе. Документ должен содержать следующую информацию:

Заголовок с названием города

Описание города, включая его историю, ключевые достопримечательности, культуру и т.д.

Список лучших мест для посещения в городе

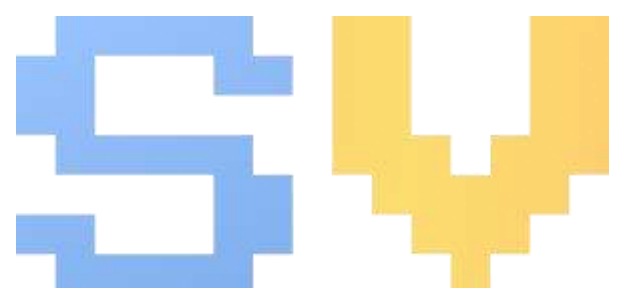
Информация о том, когда лучше всего посетить город

Изображения города

Контактная информация для посетителей, включая ссылки на официальный сайт города и туристические компании.

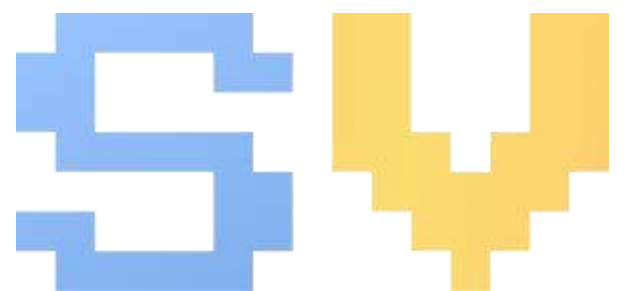
Таблица с лучшими заведениями и их адресами

Используйте внешний файл CSS для стилизации вашей страницы и поменяйте цвет текста.



# More CSS

- color - устанавливает цвет текста.
- font-size - устанавливает размер шрифта.
- font-family - устанавливает семейство шрифтов.
- font-weight - устанавливает толщину шрифта.
- text-align - устанавливает выравнивание текста.
- text-decoration - устанавливает декорацию текста (например, подчеркивание или зачеркивание).
- line-height - устанавливает высоту строки



# More CSS

Дополните свою веб-страницу, которая отображает информацию о вашем любимом городе.

Поменяйте шрифт своего текста на Roboto, и важные даты или имена в тексте выделите жирным.





# Combining Selectors

Комбинирование селекторов в CSS - это способ применения стилей к нескольким элементам на одной странице. Это может быть полезно, если вам нужно применить одинаковые стили к нескольким элементам.

Существует несколько способов комбинировать селекторы:

- Использование запятой: используя запятую, можно определить несколько селекторов, которые будут применяться к одним и тем же стилям.
- Использование пробела: с помощью пробелов между тегами можно задать вложенность внутри CSS
- Использование простого имени класса: можно использовать классы для сочетания элементов, к которым нужно применить одинаковые стили.

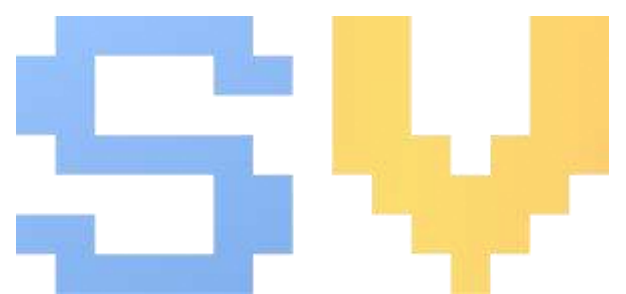
Важно заметить, что при комбинировании селекторов, порядок селекторов имеет значение. Порядок селекторов влияет на то, какие стили будут применены к элементу. Если стиль определен в нескольких местах для одного и того же элемента, то будет использован последний определенный стиль.



# Combining Selectors

Дополните свою веб-страницу, которая отображает информацию о вашем любимом городе.

Используйте комбинации селекторов, чтобы поменять нескольким элементам цвет текста.



# Class and ID Selectors

В CSS, класс и ID селекторы используются для выбора определенных элементов на странице для применения стилей.

Класс селектор используется с точкой (.) перед именем класса. Например:

```
.highlight {  
  background-color: yellow;  
}
```

`<p class="highlight">Этот текст будет иметь желтый фон</p>`

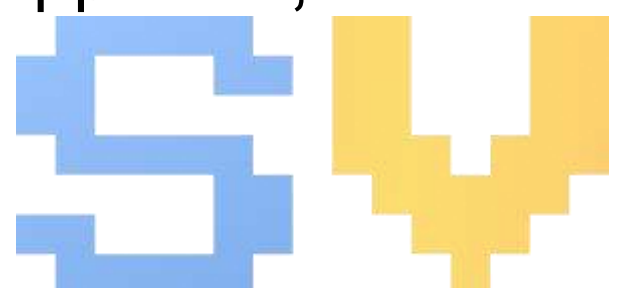
`<span class="highlight">Этот текст тоже будет иметь желтый фон</span>`

ID селектор используется с решеткой (#) перед именем ID. Например:

```
#main-header {  
  font-size: 2em;  
}
```

`<h1 id="main-header">Этот заголовок будет иметь размер шрифта 2х раз больше обычного</h1>`

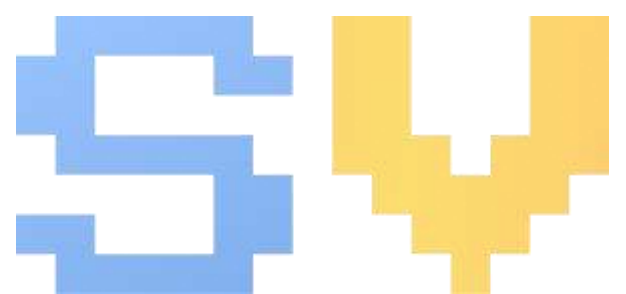
ID селектор имеет более высокую специфичность, чем класс селектор, поэтому если стиль определен как для ID, так и для класса для одного и того же элемента, то будет использован стиль, определенный для ID.



# Class and ID Selectors

Дополните свою веб-страницу, которая отображает информацию о вашем любимом городе.

Добавьте классы, чтобы отредактировать текст.



# Working With Colors

В CSS, можно указывать цвета с помощью различных систем задания цвета, таких как:

- Ключевые слова: некоторые цвета можно указать, используя ключевые слова, такие как "red", "green", "blue" и т.д.
- RGB (Red, Green, Blue): цвет можно задать, указав значения для красного, зеленого и синего цветов. Например:

```
color: rgb(255, 0, 0); /* red */
```

- HEX (hexadecimal): цвет можно задать, используя 6-значный шестнадцатеричный код цвета. Например:

```
color: #ff0000; /* red */
```

- HSL (Hue, Saturation, Lightness): цвет можно задать, указав значения для оттенка, насыщенности и яркости. Например:

```
color: hsl(0, 100%, 50%); /* red */
```

Кроме того, CSS позволяет использовать прозрачность для цветов. Это может быть сделано, используя префикс "rgba" для RGB или добавление четвертого значения для HSL, которое определяет уровень прозрачности.

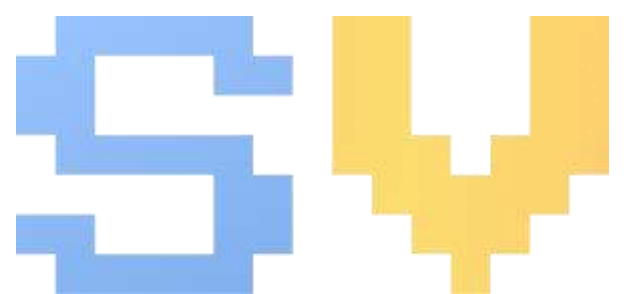
```
color: rgba(255, 0, 0, 0.5); /* red with 50% opacity */
```



# Working With Colors

Дополните свою веб-страницу, которая отображает информацию о вашем любимом городе.

Используйте разные способы написания цвета, чтобы каждый сектор имел разный цвет фона.





# Styling hyperlinks and Pseudo-classes

В CSS, псевдо-классы используются для выбора элементов на основе их состояния или контекста. Псевдо-классы добавляются к селектору элемента с помощью двоеточия (:) и имеют специальное назначение.

**:hover** - используется для выбора элемента, когда курсор наводится на него

```
a:hover {  
    color: red;  
}
```

**:active** - используется для выбора элемента, когда он нажат

```
button:active {  
    background-color: blue;  
}
```

**:visited** - используется для выбора посещенных ссылок

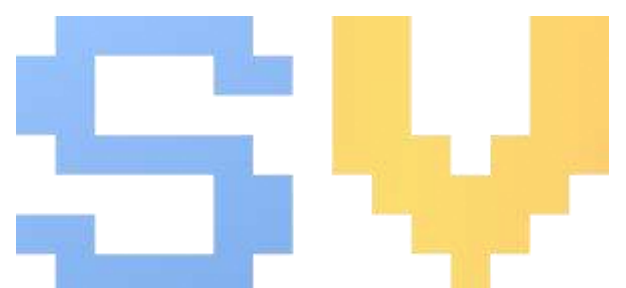
```
a:visited {  
    color: purple;  
}
```

**:first-child** - используется для выбора первого дочернего элемента внутри родителя

```
p:first-child {  
    font-weight: bold;  
}
```

**:nth-child(n)** - используется для выбора определенного дочернего элемента внутри родителя, где n - порядковый номер дочернего элемента.

```
li:nth-child(3) {  
    background-color: yellow;  
}
```



# Styling hyperlinks and Pseudo-classes

Дополните свою веб-страницу, которая отображает информацию о вашем любимом городе.

+ Добавьте псевдо-классы, чтобы при наведении на ссылки их цвет менялся на другой.

+ Добавьте псевдо-классы, чтобы каждый второй элемент таблицы имел серый фон.



# Using Chrome DevTools

Chrome DevTools - это инструменты разработчика для браузера Google Chrome, которые позволяют инспектировать и отлаживать веб-страницы. Они позволяют вам просматривать и изменять HTML и CSS код страницы в реальном времени.

Чтобы открыть DevTools в Chrome, вы можете использовать горячие клавиши (F12 или Ctrl + Shift + I для Windows / Linux, Cmd + Shift + I для Mac) или выбрать пункт меню "Инструменты разработчика" в меню браузера.

Одним из мощных функций DevTools является вкладка "Elements", которая позволяет просматривать и изменять HTML и CSS код страницы. Вы можете выбрать элемент на странице и просмотреть или изменить соответствующий HTML и CSS код. Также можно использовать функцию "Add attribute" чтобы добавить атрибуты к элементам.

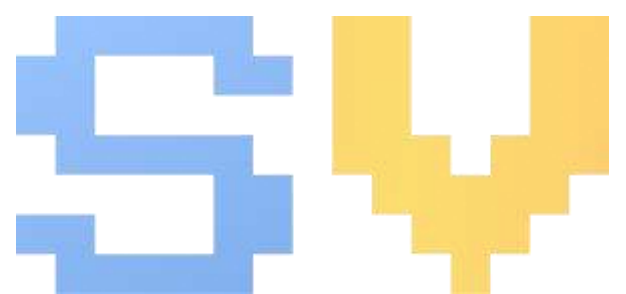
Вкладка "Styles" позволяет просматривать и изменять стили выбранного элемента. Вы можете просматривать и изменять стили, заданные в CSS файлах, а также добавлять новые стили непосредственно в DevTools. Вы также можете использовать функцию "force element state" чтобы принудительно применить состояние псевдо-класса к элементу.

Вкладка "Computed" показывает как браузер вычисляет стили для выбранного элемента, включая наследованные стили и примененные псевдо-классы.



# Using Chrome DevTools

Установите Chrome DevTools и откройте свой проект о любимом городе, сравните с вашим кодом. Исследуйте другие сайты.

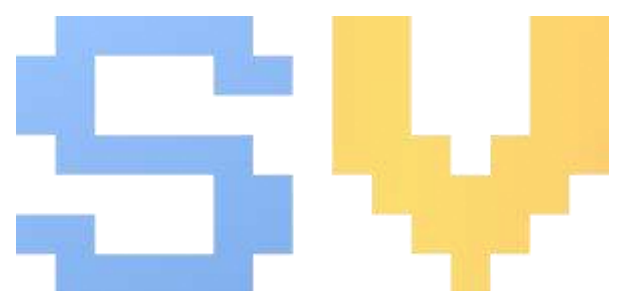


# Positioning

Позиционирование - это один из ключевых аспектов CSS, который позволяет контролировать местоположение элементов на веб-странице. В CSS есть три типа позиционирования:

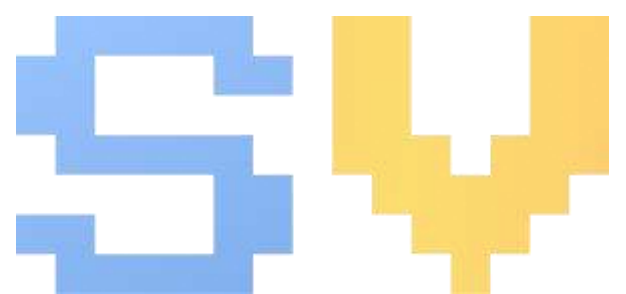
- `static` (по умолчанию) - элемент располагается в соответствии с его нормальной последовательностью в HTML коде
- `relative` - элемент может быть смещен относительно своего исходного местоположения с помощью свойств `top`, `bottom`, `left`, `right`
- `absolute` - элемент абсолютно позиционируется относительно ближайшего родителя с заданным `position` (если родитель не задан, то относительно корня страницы) с помощью свойств `top`, `bottom`, `left`, `right`.
- `fixed` - аналогично `absolute`, но элемент остаётся на своём месте при прокрутке страницы.

Важно понимать, что элементы с `position: relative` или `absolute` изменяют свою позицию в документе, и могут перекрывать другие элементы. Чтобы избежать этого, можно использовать свойство `z-index`, которое позволяет указать, какой элемент должен быть отображен поверх других.



# Positioning

Создайте два контейнера, которые не расположены друг в друге. Задайте им разный цвет и размеры. Поменяйте их позиции относительно друг друга так, чтобы один находился внутри другого.





# The CSS Box Model

CSS Box Model - это модель, которая определяет, как браузер отображает элементы HTML на странице. Каждый элемент HTML представляется в виде контейнера, который состоит из следующих частей:

- Содержимое (content) - это само содержимое элемента, например, текст внутри тега `<p>`
- Поля (padding) - пространство между содержимым элемента и его границей
- Границы (border) - линия вокруг элемента
- Отступы (margin) - пространство между элементом и другими элементами на странице

В CSS есть свойства, с помощью которых можно настраивать каждую часть Box Model:

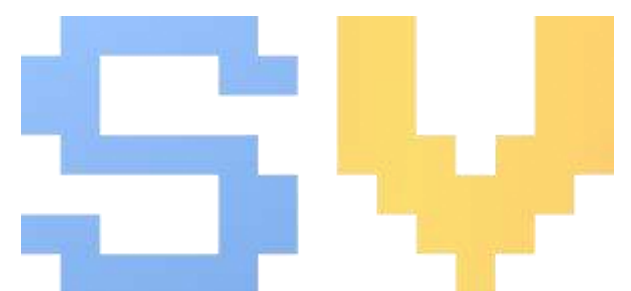
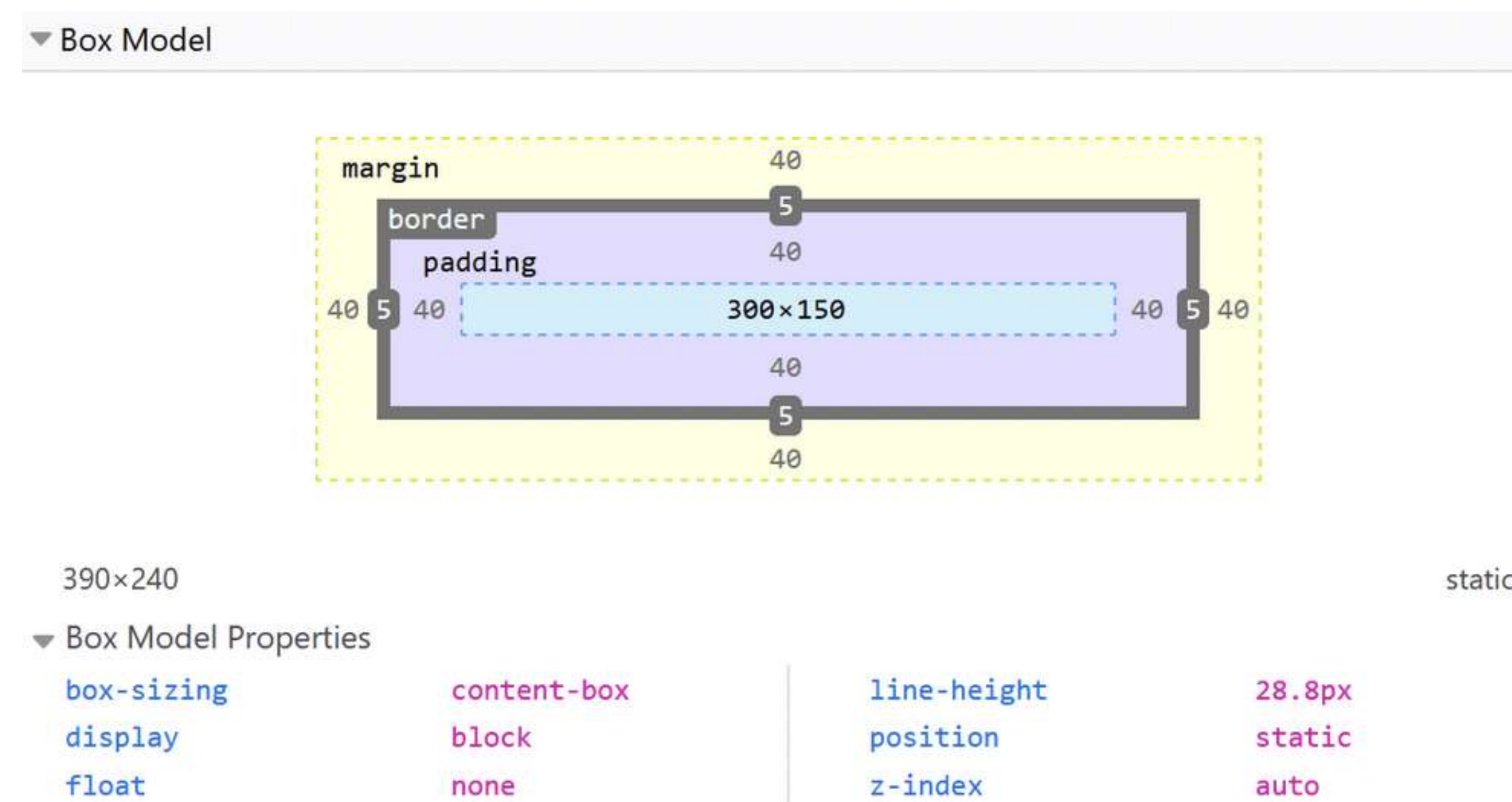
- content - width и height
- padding - padding-top, padding-right, padding-bottom, padding-left, padding
- border - border-top, border-right, border-bottom, border-left, border-width, border-color, border-style
- margin - margin-top, margin-right, margin-bottom, margin-left, margin

По умолчанию браузеры рассчитывают ширину и высоту элемента как сумму content, padding и border. Но в CSS есть свойство `box-sizing`, которое можно использовать для изменения этого поведения и рассчитывать ширину и высоту элемента только из content.



# The CSS Box Model

Используйте Chrome DevTools и исследуйте на примере любых сайтов каким образом устроены контейнеры.



# A Flexbox Overview

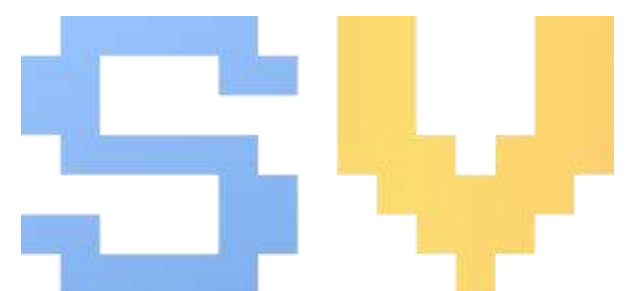
Flexbox (Flexible Box Layout) - это модуль CSS, который позволяет располагать элементы в контейнере гибким образом. Он предоставляет ряд свойств, которые можно использовать для контроля направления, порядка, выравнивания и масштабирования элементов в контейнере.

Flexbox используется для создания различных типов макетов, таких как горизонтальное и вертикальное выравнивание, масштабирование элементов и адаптивная вёрстка.

Чтобы начать использовать Flexbox, вы должны назначить контейнеру свойство `display: flex`, а затем можете использовать свойства такие как `flex-direction`, `justify-content`, `align-items`, `align-content` для управления расположением элементов в контейнере.

```
.container {  
  display: flex;  
  flex-direction: row;  
  justify-content: space-between;  
  align-items: center;  
}
```

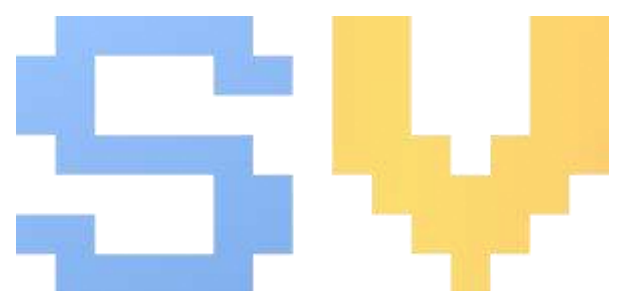
Flexbox также позволяет управлять размерами и порядком элементов с помощью свойств таких как `flex-wrap`, `flex-grow`, `flex-shrink`, `order`.



# A Flexbox Overview

При использовании Flexbox есть несколько способов для расположения и выравнивания элементов внутри контейнера.

- `justify-content` - Это свойство управляет выравниванием элементов вдоль главной оси (по умолчанию горизонтальной). Возможные значения: `flex-start`, `flex-end`, `center`, `space-between`, `space-around`.
- `align-items` - Это свойство управляет выравниванием элементов вдоль поперечной оси (по умолчанию вертикальной). Возможные значения: `flex-start`, `flex-end`, `center`, `baseline`, `stretch`.
- `align-content` - Это свойство управляет выравниванием множества строк элементов вдоль главной оси. Возможные значения: `flex-start`, `flex-end`, `center`, `space-between`, `space-around`, `stretch`.
- `margin` - Это свойство используется для установки полей вокруг элемента. Можно установить отдельное значение для каждой стороны (`top`, `right`, `bottom`, `left`) или одно общее значение для всех сторон сразу.
- `padding` - Это свойство используется для установки внутреннего отступа элемента. Можно установить отдельное значение для каждой стороны (`top`, `right`, `bottom`, `left`) или одно общее значение для всех сторон сразу.

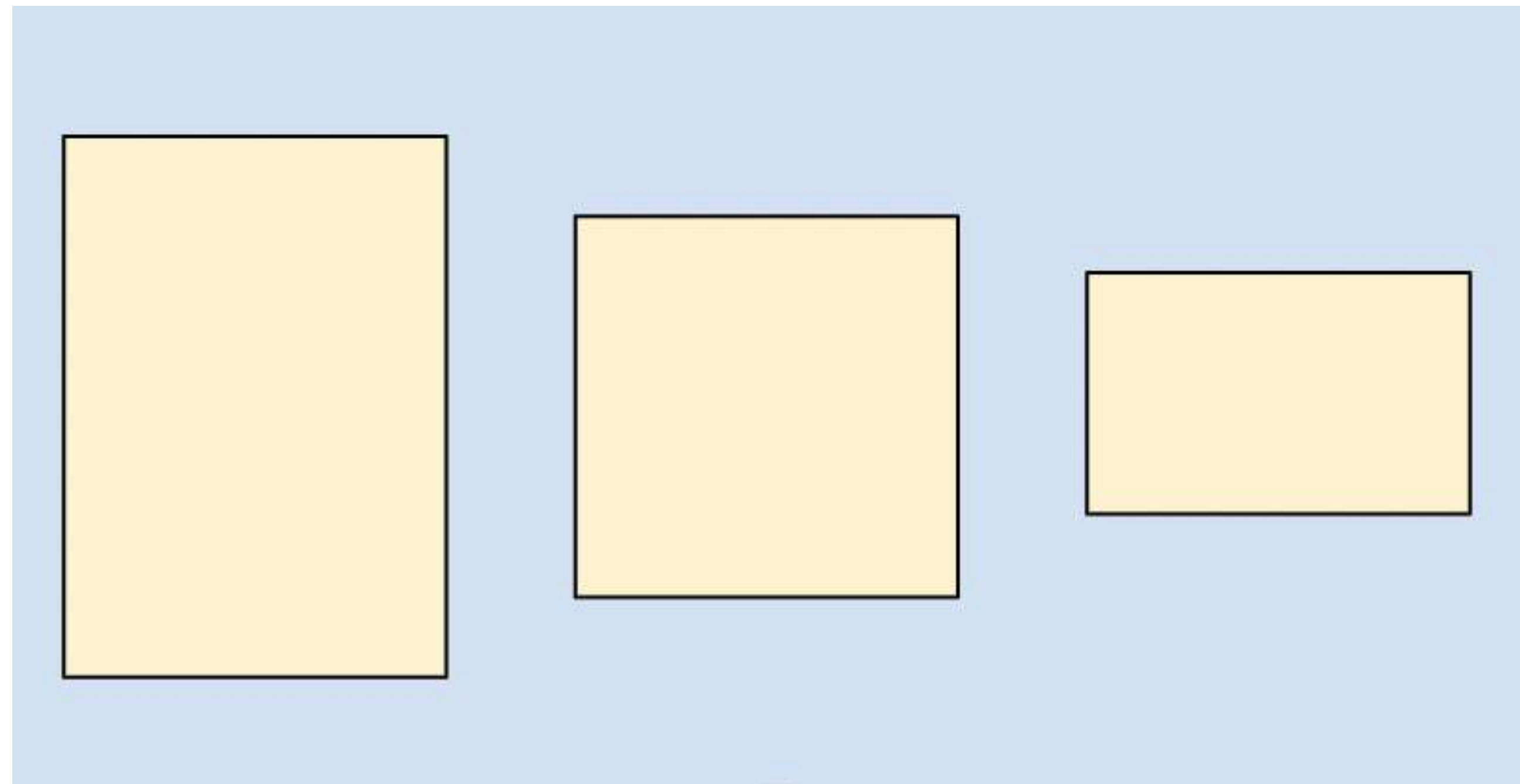




# A Flexbox Overview

Создайте три контейнера разных цветов и распложите их в колонну. Между ними задайте расстояние в 20 пикселей.

Создайте четыре контейнера разной высоты внутри другого. Задайте стили так, чтобы они все находились на середине.



# A CSS Grid Overview

CSS Grid - это модуль CSS, который предоставляет мощные инструменты для создания сеточных макетов. Он позволяет размещать элементы на сетке с использованием строк и столбцов, а также контролировать размеры и положения элементов.

Чтобы начать использовать CSS Grid, вы должны назначить контейнеру свойство `display: grid`, а затем можете использовать свойства такие как `grid-template-rows`, `grid-template-columns`, `grid-row-gap`, `grid-column-gap` для создания сетки.

```
.container {  
  display: grid;  
  grid-template-rows: 50px 50px;  
  grid-template-columns: 100px 200px;  
  grid-row-gap: 10px;  
  grid-column-gap: 20px;  
}
```

Кроме того, вы можете использовать свойства такие как ``grid-row``, ``grid-column``, ``grid-row-start``, ``grid-row-end``, ``grid-column-start``, ``grid-column-end`` для размещения элементов на сетке. CSS Grid также поддерживает адаптивность и может использоваться совместно с Flexbox для решения сложных макетных задач.





# A CSS Grid Overview

В CSS Grid, вы можете определить размеры строк и столбцов с помощью свойств `grid-template-rows` и `grid-template-columns`.

`grid-template-rows` используется для определения высоты строк в гриде, а `grid-template-columns` - для ширины столбцов.

Вы можете задать размеры строк и столбцов в пикселях, процентах или ключевом слове `fr` (fraction), которое означает "доля свободного места".



# A CSS Grid Overview

Создайте контейнер с двумя рядами: первый имеет 3 одинаковых контейнера и второй два других контейнера тоже одинакового размера. Задайте расстояние между контейнерами.

Создайте контейнер с двумя рядами: первый имеет 3 одинаковых контейнера и второй два других контейнера тоже одинакового размера. Задайте разные высоты этим двум рядам.

