



# CSS

Cascading Style  
Sheets  
(Каскадные  
таблицы стилей.)



# ПЛАН

**1**

Подключение css стилей.

**2**

Синтаксис.

**3**

Базовые селекторы.

**4**

Базовые стили.

**5**

Работа с текстом.



# CSS

CSS (Cascading Style Sheets) - это язык стилей, который используется для оформления веб-страниц. Он определяет, как элементы HTML должны отображаться на экране, включая их цвета, шрифты, размеры, расположение и другие аспекты внешнего вида.



# ПОДКЛЮЧЕНИЕ CSS

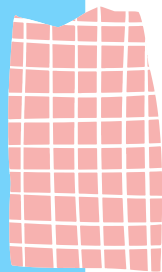
1 Inline стили:  
`style="color: blue; font-size: 16px;`

2 Внутренние стили:  
`<style></style>`

3 Внешние стили:  
`<link></link>`

# INLINE СТИЛИ:

Inline стили встроены непосредственно в HTML элементы и задаются с помощью атрибута style.

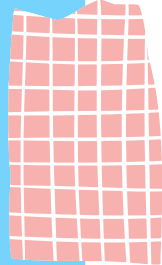


---

Inline стили являются самыми простыми для использования, но имеют несколько ограничений. Они применяются только к конкретному элементу, что делает их неудобными для повторного использования и обслуживания больших проектов.

# ВНУТРЕННИЕ СТИЛИ:

Внутренние стили задаются внутри секции `<style>` внутри блока `<head>` в HTML документе. Они применяются ко всем элементам, находящимся внутри HTML файла. Пример:

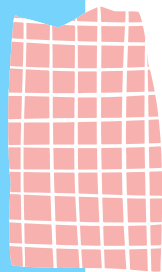


---

Внутренние стили полезны, когда у вас есть небольшой набор стилей, который применяется только к определенной странице. Однако они все равно не обладают гибкостью внешних стилей и могут усложнить обслуживание, если нужно применить их ко многим файлам.

# ВНЕШНИЕ СТИЛИ:

Внешние стили описываются в отдельных CSS файлах и подключаются к HTML документам с помощью элемента `<link>`.



Внешние стили предоставляют наибольшую гибкость и удобство в использовании. Они позволяют разделять структуру (HTML) и оформление (CSS), а также легко обновлять стили для всего сайта, изменяя только один файл CSS.

Рекомендуется использовать внешние стили (`<link>`), поскольку они являются наиболее эффективным и распространенным способом подключения стилей.

# СЕЛЕКТОРЫ





Селекторы в CSS используются для выбора и стилизации определенных элементов HTML. Они определяют, какие элементы будут применяться к определенным стилям.

#### 1) Селектор элемента

```
5 p {  
6   /* Стили для всех <p> элементов */  
7 }
```

#### 2) Селектор класса

```
10 .my-class {  
11   /* Стили для всех элементов с классом "my-class" */  
12 }
```

#### 3) Селектор идентификатора

```
5 #my-id {  
6   /* Стили для элемента с идентификатором "my-id" */  
7 }
```

#### 4) Селектор потомка

```
9  
10 div p {  
11   /* Стили для всех <p> элементов,  
12      которые являются потомками <div> элементов */  
13 }
```

#### 5) Селектор дочернего элемента

```
div > p {  
   /* Стили для всех <p> элементов,  
      которые являются прямыми дочерними элементами <div> элементов */  
}
```

#### 6) Селектор атрибута

```
input[type="text"] {  
   /* Стили для всех <input> элементов с атрибутом type="text" */  
}
```

## 7) Селектор псевдокласса

```
/* Селектор псевдокласса */  
a: hover {  
    /* Стили для ссылок, когда они находятся под указателем мыши */  
}
```

## 8) Селектор псевдоэлемента

```
/* Селектор псевдоэлемента */  
p::first-line {  
    /* Стили для первой строки внутри каждого <p> элемента */  
}
```

## 9) Селектор группы

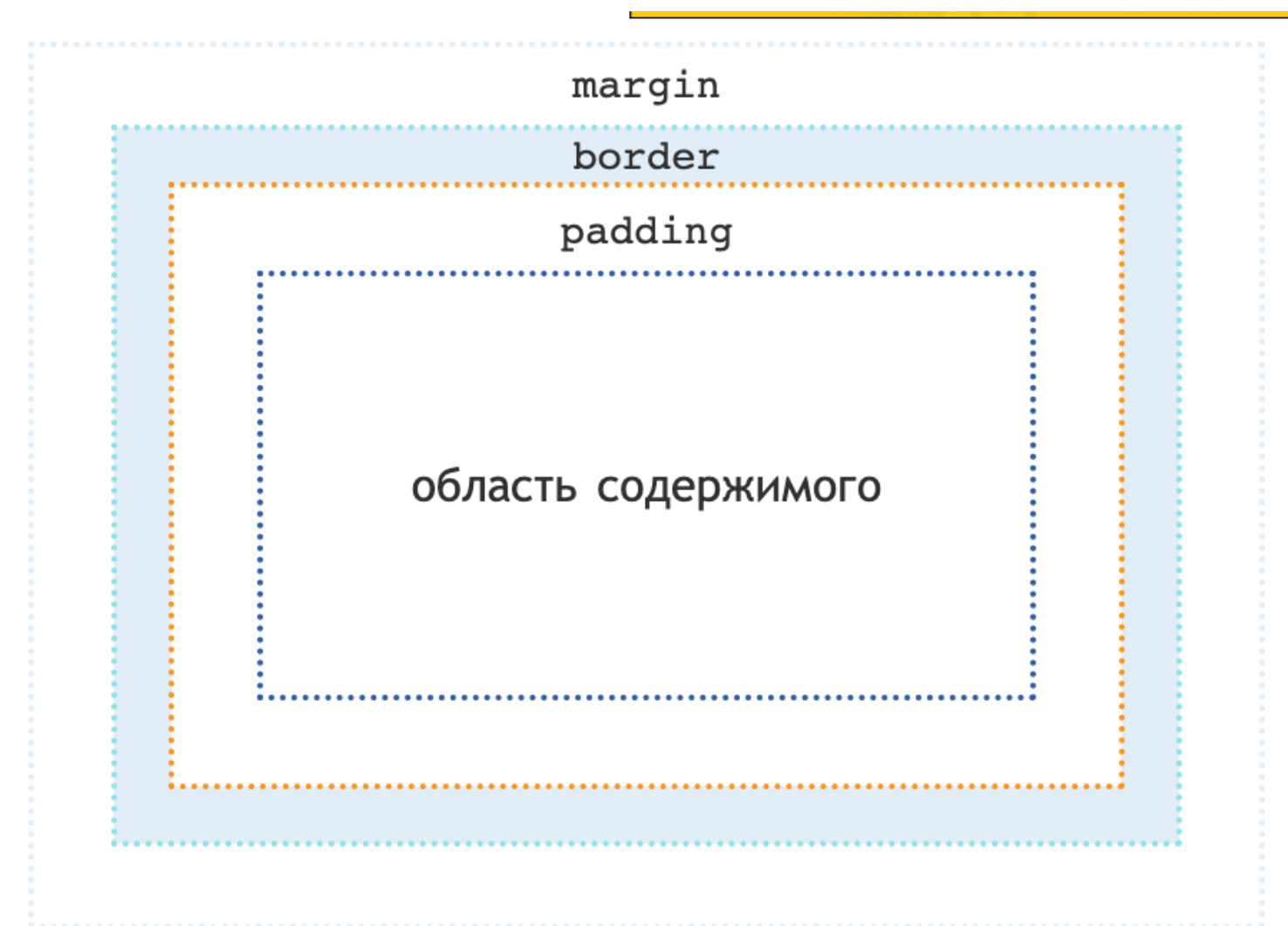
```
/* Селектор группы */  
h1, h2, h3 {  
    /* Стили для всех <h1>, <h2> и <h3> элементов */  
}
```

## 10) Универсальный селектор

```
/* Универсальный селектор: */  
* {  
    /* Стили для всех элементов на странице */  
}
```

# БЛОЧНАЯ МОДЕЛЬ

Каждый блок имеет область содержимого, в которой находится текст, дочерние элементы, изображение и т.п., и необязательные окружающие ее padding, border и margin. Размер каждой области определяется соответствующими свойствами и может быть нулевым, или, в случае margin, отрицательным.



Поля, рамка и отступы могут быть разбиты на верхний, правый, нижний и левый сегменты, каждый из которых независимо управляется своим соответствующим свойством.

Фон области содержимого, полей и рамки блока определяется свойствами фона. Область рамки может быть дополнительно окрашена с помощью свойства `border`.

Отступы элемента всегда прозрачны, что позволяет показывать фон родительского элемента.

Так как поля и отступы элемента не являются обязательными, по умолчанию их значение равно нулю. Тем не менее, некоторые браузеры добавляют этим свойствам положительные значения по умолчанию на основе своих таблиц стилей. Очистить стили браузеров для всех элементов можно при помощи универсального селектора:

```
* {  
  margin: 0;  
  padding: 0;  
}
```

# MARGIN

Отступы окружают край рамки элемента, обеспечивая расстояние между соседними блоками. Свойства отступов определяют их толщину. Применяются ко всем элементам, кроме внутренних элементов таблицы. Сокращенное свойство `margin` задает отступы для всех четырех сторон, а его подсвойства задают отступ только для соответствующей стороны.

```
div {  
    margin: 10px; /* Установка одинаковых отступов для всех сторон */  
}  
  
div {  
    margin: 10px 20px; /* Установка вертикальных отступов  
    (10px сверху и снизу) и горизонтальных отступов (20px слева и справа) */  
}  
  
div {  
    margin: 10px 20px 30px; /* Установка вертикальных отступов  
    (10px сверху и 30px снизу) и левого и правого отступов (20px) */  
}  
  
div {  
    margin: 10px 20px 30px 40px; /* Установка отступов для всех сторон  
    (10px сверху, 20px справа, 30px снизу, 40px слева) */  
}
```

- СХЛОПЫВАНИЕ
- ВЕРТИКАЛЬНЫХ
- ОТСТУПОВ

Смежные вертикальные отступы двух или более элементов уровня блока `margin` объединяются (перекрываются). При этом ширина общего отступа равна ширине большего из исходных. Исключение составляют отступы корневого элемента, которые не схлопываются.

Для предотвращения проблемы схлопывания рекомендуется задавать для всех элементов только верхний или нижний `margin`.

Для предотвращения проблемы схлопывания рекомендуется задавать для всех элементов только верхний или нижний `margin`.

# PADDING

Свойство padding в CSS используется для задания внутренних отступов (пространства) вокруг содержимого элемента. Оно определяет расстояние между границей элемента и его содержимым.

```
div {  
    padding: 10px; /* Установка одинаковых отступов для всех сторон */  
}  
  
div {  
    padding: 10px 20px; /* Установка вертикальных отступов  
    (10px сверху и снизу) и горизонтальных отступов (20px слева и справа) */  
}  
  
div {  
    padding: 10px 20px 30px; /* Установка вертикальных отступов  
    (10px сверху и 30px снизу) и левого и правого отступов (20px) */  
}  
  
div {  
    padding: 10px 20px 30px 40px; /* Установка отступов для всех сторон  
    (10px сверху, 20px справа, 30px снизу, 40px слева) */  
}
```



Некоторые нюансы, связанные с padding:

1. Отступы добавляются к размерам элемента и могут влиять на его положение в документе. Например, если у элемента задано width: 200px и padding: 10px, то его фактическая ширина будет равна 220px (200px + 10px слева + 10px справа).
2. Значения отступов могут быть заданы в различных единицах измерения, таких как пиксели (px), проценты (%), относительные (em, rem и т.д.), а также ключевые слова (initial, inherit, auto и т.д.).
3. padding не создает видимых границ вокруг элемента, а лишь определяет внутреннее пространство. Если вам нужны видимые границы, используйте свойство border.
4. Комбинирование padding с другими свойствами, такими как margin, width, height, позволяет более точно управлять расположением и внешним видом элементов на странице.
5. Отступы могут влиять на расчет размеров элементов при использовании определенных свойств, например, box-sizing. По умолчанию, padding добавляется к размерам элемента, но при использовании box-sizing: border-box, отступы включаются в общие размеры элемента.

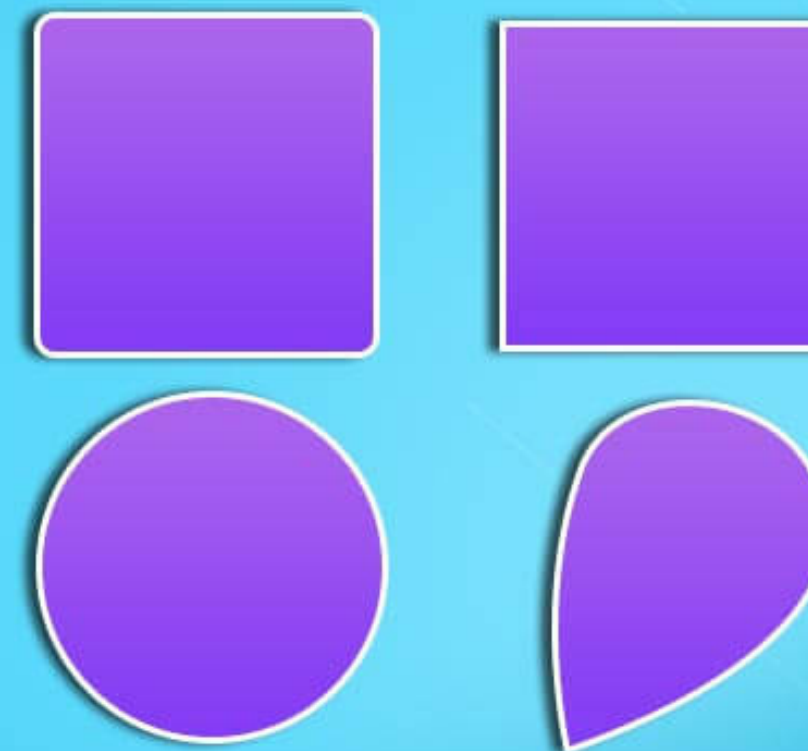


- 
- 
- 
- 

# BORDER

Свойство border в CSS используется для установки стиля, ширины и цвета границы элемента. Оно позволяет задать внешнюю рамку вокруг элемента.

## CSS Border Radius



Для настройки стиля, ширины и цвета границы отдельно, вы можете использовать следующие свойства:

1. **border-width:** Задаёт ширину границы элемента. Значение может быть задано в пикселях (px), процентах (%), относительных (em, rem)
2. **border-style:** Задаёт стиль границы элемента. Возможные значения включают none (без границы), solid (сплошная граница), dashed (пунктирная граница), dotted (точечная граница), double (двойная граница) и другие.
3. **border-color:** Задаёт цвет границы элемента. Значение может быть задано в виде имени цвета (например, red, blue, green), кода цвета (#RRGGBB или #RGB), ключевых слов (transparent, initial, inherit) или RGB-значений (rgb()) или rgba()).

```
div {  
  border-width: 2px; /* Ширина границы 2 пикселя для всех сторон */  
}  
  
div {  
  border-style: dashed; /* Пунктирная граница для всех сторон */  
}  
  
div {  
  border-color: red; /* Красный цвет границы для всех сторон */  
}  
  
div {  
  border: 1px solid black; /* Установка границы толщиной 1 пиксель  
и сплошным стилем черного цвета */  
}
```

- 1. **groove**: Создает эффект "впадины" на границе элемента. Он выглядит так, как будто граница втягивается внутрь элемента. Часто используется для создания трехмерных эффектов.
- 2. **inset**: Создает эффект "внутренней" тени на границе элемента. Он выглядит так, будто граница углубляется внутрь элемента. Часто используется для создания кнопок или выделения активного элемента.
- 3. **outset**: Создает эффект "внешней" тени на границе элемента. Он выглядит так, будто граница выступает из элемента. Часто используется для создания кнопок или трехмерных эффектов.
- 4. **ridge**: Создает эффект "выступающей" границы элемента. Он выглядит так, будто граница поднимается над уровнем элемента. Часто используется для создания трехмерных эффектов.

border-style	
(border-top-style, border-right-style, border-bottom-style, border-left-style)	
Значения:	
none	Значение по умолчанию, означает отсутствие рамки. Также убирает рамку элемента из группы элементов с установленным значением данного свойства.
hidden	Эквивалентно none .
dotted	<div>dotted</div>
dashed	<div>dashed</div>
solid	<div>solid</div>
double	<div>double</div>
groove	<div>groove</div>
ridge	<div>ridge</div>
inset	<div>inset</div>
outset	<div>outset</div>
{1,4}	<div>Одновременное перечисление четырех разных стилей для рамок элемента, только для свойства border-style : {border-style: solid dotted none dotted;}</div>
initial	Устанавливает значение свойства в значение по умолчанию.
inherit	Наследует значение свойства от родительского элемента.

- 
- 
- 
- 

# ТЕКСТ

Различные свойства CSS обеспечивают контроль над преобразованием регистра, обработкой пробелов, правилами переноса и переносом текста и строк, интервалами и отступами.

Так же обеспечивают настройку внешнего вида текстового контента. Они позволяют управлять размером, шрифтом, выравниванием и другими аспектами текста.

- 1.color: Задает цвет текста.
- 2.font-family: Задает шрифт текста.
- 3.font-size: Задает размер шрифта.
- 4.font-weight: Задает насыщенность шрифта (жирный или обычный).
- 5.font-style: Задает стиль шрифта (наклонный, обычный или курсивный).
- 6.text-align: Выравнивает текст по горизонтали.
- 7.line-height: Задает высоту строки текста.
- 8.text-decoration: Задает декорации текста (подчеркивание, зачеркивание и т.д.).
- 9.text-transform: Преобразует регистр текста (в верхний, нижний или заглавный).
- 10.text-shadow: Добавляет тень к тексту.
- 11.letter-spacing: Задает интервал между буквами.
- 12.word-spacing: Задает интервал между словами.
- 13.text-indent: Задает отступ первой строки текста.
- 14.white-space: Управляет обработкой пробелов и переносов строки внутри текста.
- 15.overflow-wrap: Определяет, как текст обрабатывается, если он не помещается в заданную ширину.
- 16.text-overflow: Определяет, как текст обрезаются, если он не помещается в заданную область.
- 17.vertical-align: Выравнивает текст по вертикали относительно базовой линии.
- 18.text-align-last: Определяет выравнивание последней строки в блоке текста, содержащего несколько строк.
- 19.direction: Задает направление текста (слева направо или справа налево).

Это лишь некоторые из наиболее часто используемых базовых свойств текста в CSS.

Существует множество других свойств, позволяющих более тонко настроить внешний вид текстового контента.



**THANK  
YOU!**

Have a  
great day  
ahead.