

Справочники

- <https://www.w3schools.com/html/default.asp>
- <https://developer.mozilla.org/ru/docs/Web/HTML>
- <https://developer.mozilla.org/ru/docs/Web/CSS/Reference>
- <https://webref.ru/css>
- <https://www.w3schools.com/css/default.asp>

Блочные и строчные элементы

Чтобы создать дерево блоков, CSS сначала использует каскадирование и наследование, позволяющие назначить вычисленное значение для каждого css-свойства каждому элементу и текстовому узлу в исходном дереве. Затем для каждого элемента CSS генерирует ноль или более блоков в соответствии со значением свойства `display` этого элемента. Как правило, элемент генерирует один основной блок, который представляет самого себя и содержит свое содержимое.

Положение блоков на странице определяется следующими факторами:

- размером элемента (с учётом того, заданы они явно или нет);
- типом элемента (строчный или блочный);
- схемой позиционирования (нормальный поток, позиционированные или плавающие элементы);
- отношениями между элементами в DOM (родительский — дочерний элемент);
- внутренними размерами содержащихся изображений;
- внешней информацией (например, размеры окна браузера).

Блочные элементы и блочные контейнеры

Блочные элементы — элементы высшего уровня, которые форматируются визуально как блоки, располагаясь на странице в окне браузера вертикально.

Значения свойства `display`, такие как `block`, `list-item` и `table` делают элементы блочными. Блочные элементы генерируют основной блок, который содержит только блок элемента.

Элементы со значением `display: list-item` генерируют дополнительные блоки для маркеров, которые позиционируются относительно основного блока.

Блочные элементы могут размещаться непосредственно внутри элемента `<body>`. Они создают разрыв строки перед элементом и после него, образуя прямоугольную область, по ширине занимающую всю ширину веб-страницы или блока-родителя.

Примеры: `<address>`, `<article>`, `<aside>`, `<blockquote>`, `<dd>`, `<div>`, `<dl>`, `<dt>`, `<details>`, `<fieldset>`, `<figcaption>`, `<figure>`, `<footer>`, `<form>`, `<h1>`-`<h6>`, `<header>`, `<li>`, `<nav>`, `<ol>`, `<output>`, `<p>`, `<section>`, `<summary>`, `<table>`, `<ul>` и др.

Строчные элементы и строчные контейнеры

Встроенные (строчные) элементы генерируют внутрострочные контейнеры. Они не формируют новые блоки контента. Значения свойства `display`, такие как `inline` и `inline-table` делают элементы строчными.

Примеры: `<a>`, `<cite>`, `<code>`, `<dfn>`, `<del>`, `<em>`, `<i>`, `<img>`, `<ins>`, `<kbd>`, `<label>`, `<map>`, `<mark>`, `<s>`, `<samp>`, `<small>`, `<span>`, `<strong>`, `<q>` и др.

Строчно-блочные элементы

Такие элементы являются встроенными, но для них можно задавать ширину и высоту.

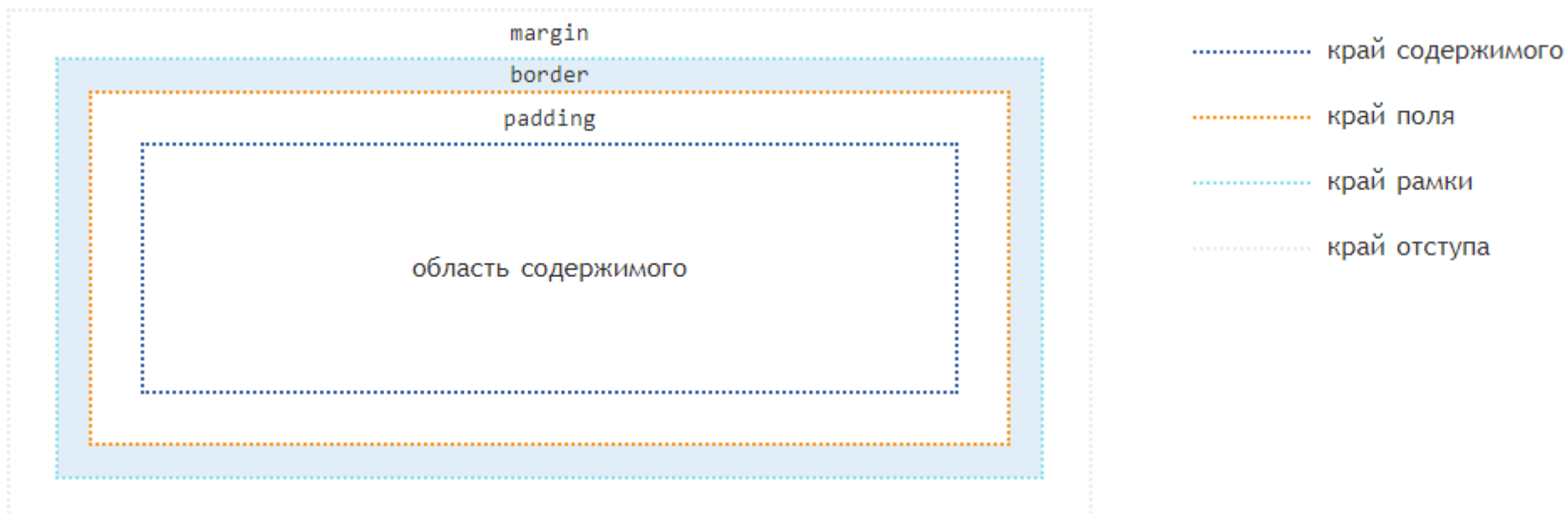
```
{display: inline-block;}
```

Примеры: `<audio>`, `<button>`, `<canvas>`, `<input>`, `<meter>`, `<progress>`, `<select>`, `<textarea>`, `<video>` и др.

CSS блочная модель

Модуль CSS Box Model описывает свойства `padding` и `margin`, которые создают поля внутри и отступы снаружи CSS блока. Размеры блока также могут быть увеличены за счет рамки.

Каждый блок имеет прямоугольную область содержимого в центре, поля вокруг содержимого, рамку вокруг полей и отступ за пределами рамки. Размеры этих областей определяют свойства `padding` и его подсвойства — `padding-left`, `padding-top` и т.д., `border` и его подсвойства, `margin` и его подсвойства.



Физические свойства отступов: свойства `margin-top`, `margin-right`, `margin-bottom`, `margin-left`

Значения	Описание
длина	Размер отступа задается в единицах длины, например, px, in, em. Значение по умолчанию 0.
%	Вычисляется относительно ширины блока контейнера. Изменяются, если изменяется ширина родительского элемента.
auto	Для элементов уровня строки, плавающих (float) значения <code>margin-left</code> или <code>margin-right</code> вычисляются в 0. Если для элементов уровня блока задано <code>margin-left: auto</code> или <code>margin-right: auto</code> — соответствующее поле расширяется до края содержащего блока, если оба — их значения становятся равными, что горизонтально центрирует элемент относительно краев содержащего блока.
initial	Устанавливает значение свойства в значение по умолчанию.
inherit	Наследует значение свойства от родительского элемента.

```
margin-top: 20px;  
margin-right: 1em;  
margin-bottom: 5%;  
margin-left: auto;  
margin-top: inherit;  
margin-right: initial;
```

Краткая запись отступов: свойство `margin`

Свойство `margin` является сокращенным свойством для установки `margin-top`, `margin-right`, `margin-bottom` и `margin-left` в одном объявлении.

Если существует только одно значение, оно применяется ко всем сторонам.

Если два — верхний и нижний отступы устанавливаются на первое значение, а правый и левый — устанавливаются на второе.

Если имеется три значения — верхний отступ устанавливается на первое значение, левый и правый — на второе, а нижний — на третье.

Если есть четыре значения — они применяются сверху, справа, снизу и слева соответственно.

```
p {margin: 20px auto; }
```


Поля элемента

Область полей представляет собой пространство между краем области содержимого и рамкой элемента. Свойства полей определяют толщину их области. Применяются ко всем элементам, кроме внутренних элементов таблицы (за исключением ячеек таблицы). Сокращенное свойство `padding` задает поля для всех четырех сторон, а подсвойства устанавливают только их соответствующие стороны. Краткая запись полей работает аналогично отступам.

длина	Поля элемента задаются при помощи единиц длины, например, px, pt, cm. Значение по умолчанию 0.
%	Вычисляются относительно ширины родительского элемента, могут меняться при изменении ширины элемента. Поля сверху и снизу равны полям слева и справа, т.е. верхние и нижние поля тоже вычисляются относительно ширины элемента.

```
padding-top: 0.5em;  
padding-right: 0;  
padding-bottom: 2cm;  
padding-left: 10%;  
padding-top: inherit;  
padding-bottom: initial;
```

Ширина и высота

Свойство `width` определяет ширину содержимого блока. Это свойство не применяется к незамещаемым строчным элементам `display: inline`; Ширина содержимого встроенных блоков определяется шириной отображаемого содержимого внутри них. Встроенные блоки сливаются в линейные блоки. Отрицательные значения не допускаются. Свойства `min-width` и `max-width` позволяют ограничивать ширину содержимого до определенного диапазона. Значения не могут быть отрицательными. Для `min-width` значение по умолчанию 0, для `max-width` — `none`.

Свойство `height` определяет высоту содержимого блока. Это свойство не применяется к незамещаемым строчным элементам. Значения длины не могут быть отрицательными.

Иногда полезно ограничить высоту элементов определенным диапазоном. Свойства `min-height` и `max-height` предлагают эту функциональность.

```
width: 100px;  
width: 10em;  
width: 50%;  
width: auto;  
width: inherit;
```

```
min-width: 100px;  
min-width: 10em;  
min-width: 50%;  
min-width: inherit;  
max-width: 500px;  
max-width: 20em;  
max-width: 80%;  
max-width: none;  
max-width: inherit;
```

```
height: 100px;  
height: 10em;  
height: 50%;  
height: auto;  
height: inherit;
```

```
min-height: 100px;  
min-height: 2em;  
min-height: 50%;  
min-height: inherit;  
max-height: 500px;  
max-height: 20em;  
max-height: 80%;  
max-height: none;  
max-height: inherit;
```

CSS-рамка

CSS-рамка элемента представляет собой одну или несколько линий, окружающих содержимое элемента и его поля `padding`. Рамка задаётся с помощью краткого свойства `border`. Стилль рамки задается с помощью трех свойств: стиль, цвет и ширина.

По умолчанию рамки всегда отрисовываются поверх фона элемента, фон распространяется до внешнего края элемента. Стилль рамки определяет ее отображение, без этого свойства рамки не будут видны вообще. Для элемента можно задавать рамку для всех сторон одновременно с помощью свойства `border-style` или для каждой стороны отдельно с помощью уточняющих свойств `border-top-style` и т.д. **Не наследуется.**

```
p {border-style: solid; }  
p {border-top-style: solid;}
```

CSS-памка

`border-style: none;`

`border-style: hidden;`

`border-style: solid;`

`border-style: dotted;`

`border-style: dashed;`

`border-style: double;`

`border-style: groove;`

`border-style: ridge;`

`border-style: inset;`

`border-style: outset;`

`border-style: solid dotted dashed double;`

CSS-рамка

border-color (border-top-color, border-right-color, border-bottom-color, border-left-color)	
transparent	Устанавливает прозрачный цвет для рамки. При этом ширина рамки остается. Можно использовать для смены цвета рамки при наведении курсора мыши на элемент, чтобы избежать смещение элемента. <code>p {border-color: transparent; }</code>
цвет	Цвет рамок задается при помощи значений свойства color. <code>p {border-color: #cacd58;}</code>
border-width (border-top-width, border-right-width, border-bottom-width, border-left-width)	
thin / medium / thick	Ключевые слова, устанавливают ширину рамки относительно друг друга. Первое значение уже, чем второе, второе — тоньше третьего. Значение по умолчанию — medium
Размер в px, em	<code>p {border-width: 5px;}</code>

Свойство `border` позволяет объединить в себе следующие свойства: `border-width`, `border-style`, `border-color`, например:

```
p {border: 2px solid grey;}
```

CSS3-рамка

CSS3-рамка дополняет возможности форматирования границ элементов с помощью свойств, позволяющих закруглить углы элемента, а также использовать изображения для оформления границ элемента.

Закругление углов с помощью `border-radius`

Свойство позволяет закруглить углы строчных и блочных элементов. Кривая для каждого угла определяется с помощью одного или двух радиусов, определяющих его форму — круга или эллипса.

Свойство `border-radius` позволяет закруглить все `border-top-left-radius`, `border-top-right-radius`, `border-bottom-right-radius`, `border-bottom-left-radius` углы одновременно, а с помощью этих свойств можно закруглить каждый угол отдельно.

Если задать два значения для свойства `border-radius`, то первое значение закруглит верхний левый и нижний правый угол, а второе — верхний правый и нижний левый.

Значения, заданные через `/`, определяют горизонтальные и вертикальные радиусы.

Свойство box-sizing

По умолчанию в блочной модели CSS ширина и высота, которая задаётся элементу, применяется только для контента элемента. Если у элемента есть рамка или поле, то они добавляются к ширине и высоте, чтобы получить отображаемый на экране размер. Например, если у вас есть четыре блока с `width: 25%;`, и у какого-нибудь из них есть рамка или поле слева или справа, то по умолчанию они не поместятся на одной строке.

Свойство `box-sizing` может изменять это поведение:

- `content-box` даёт стандартное поведение свойства `box-sizing`. Если вы выставите элементу ширину 100 пикселей, то ширина его контента будет 100 пикселей, а ширина рамки и внутренних полей при рендере будет добавлена к финальной ширине, делая элемент шире ста пикселей.
- `border-box` говорит браузеру учитывать любые границы и внутренние отступы в значениях, которые вы указываете в ширине и высоте элемента. Если вы выставите элементу ширину 100 пикселей, то эти 100 пикселей будут включать в себя рамку и внутренние поля, а контент сожмётся, чтобы выделить для них место. Обычно это упрощает работу с размерами элементов.