Модель визуального форматирования CSS

* **Содержание:**
* [1. Модель визуального форматирования](https://html5book.ru/block-inline-elements/#part1)
* [2. Блочная модель](https://html5book.ru/block-inline-elements/#part2)
* [3. Блочные элементы и блочные контейнеры](https://html5book.ru/block-inline-elements/#part3)
* [4. Строчные элементы и строчные контейнеры](https://html5book.ru/block-inline-elements/#part4)
* [5. Строчно-блочные элементы](https://html5book.ru/block-inline-elements/#part5)
* [6. Как сделать строчный элемент блочным](https://html5book.ru/block-inline-elements/#part6)
* [7. Схлопывание вертикальных отступов](https://html5book.ru/block-inline-elements/#part7)
* [8. Отрицательные отступы](https://html5book.ru/block-inline-elements/#part8)
* [9. Выпадание вертикальных отступов](https://html5book.ru/block-inline-elements/#part9)

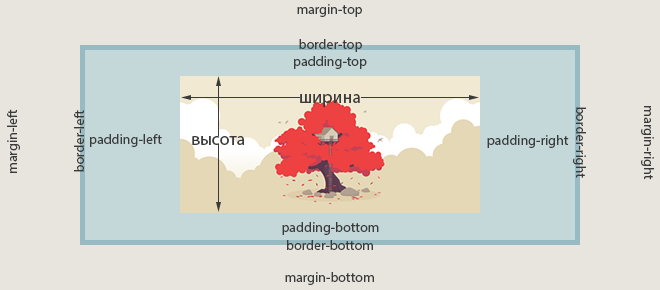
1. Модель визуального форматирования

Модель визуального форматирования CSS представляет собой алгоритм, который обрабатывает HTML-документ и выводит его на экран устройства. Данная модель преобразует каждый элемент документа таким образом, что он генерирует ноль или более прямоугольных боксов в соответствии с блочной моделью CSS. Положение этих боксов 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;  
— внутренними размерами содержащихся изображений;  
— внешней информацией (например, размеры окна браузера).

Положение и размеры бокса(ов) каждого элемента рассчитываются относительно краёв прямоугольной блока, так называемого **«содержащего блока»** (containing block). Размеры бокса не ограничиваются размерами содержащего блока, поэтому при определённых условиях он может выходить за его границы.

2. Блочная модель

В блочной модели элемент рассматривается как прямоугольный контейнер, имеющий область содержимого и необязательные рамки и отступы (внутренние и внешние). Свойство display определяет тип контейнера элемента. Для каждого элемента существует значение браузера по умолчанию.

РИС. 1. БЛОЧНАЯ МОДЕЛЬ ЭЛЕМЕНТА

**Область содержимого** — это содержимое элемента, например, текст или изображение.

**Внутренний отступ** задаётся свойством padding. Внутренний отступ — это расстояние между основным содержимым и его границей (рамкой). Если для элемента задать фон, то он распространится также и на поля элемента. Внутренний отступ не может принимать отрицательных значений, в отличие от внешнего отступа.

**Внешний отступ** задаётся свойством margin. Он добавляет расстояние снаружи элемента от внешней границы рамки до соседних элементов, тем самым разделяя элементы на странице. Внешние отступы всегда остаются прозрачными и через них виден фон родительского элемента.

Значения padding и margin задаются в следующем порядке: верхнее, правое, нижнее и левое.

**Граница**, или рамка элемента, задаётся с помощью свойства border. Если цвет рамки не задан, она принимает цвет основного содержимого элемента, например, текста. Если рамка имеет разрывы, то сквозь них будет проступать фон элемента.

Внешние, внутренние отступы и рамка 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;

}

3. 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.

<address>, <article>, <aside>,  
<blockquote>,  
<dd>, <div>, <dl>, <dt>, <details>,  
<fieldset>, <figcaption>, <figure>, <footer>, <form>,  
<h1>-<h6>, <header>, <hr>,  
<iframe>,  
<li>, <legend>,  
<nav>, <noscript>,  
<ol>, <output>, <optgroup>, <option>,  
<p>, <pre>,  
<section>, <summary>,  
<table>,  
<ul>

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 (если для элемента не задано значение width).

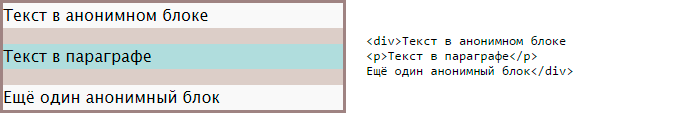
Свойства width и height устанавливают ширину и высоту области содержимого элемента. Фактическая ширина элемента складывается из ширины полей (внутренних отступов) padding, границ border и внешних отступов margin.

Блочные элементы могут содержать как строчные, так и блочные элементы, но не оба типа элементов сразу. При необходимости, строки текста, принадлежащие блочному контейнеру, могут быть обёрнуты анонимными контейнерами, которые будут вести себя внутри блока как элементы со значением display: block;, а строчные элементы обёрнуты элементом <p>. Блочные элементы могут содержаться только в пределах блочных элементов.

Элемент <p> относится к блочным элементам, но он не может содержать внутри себя другой элемент <p>, а также любой другой блочный элемент.

Анонимные боксы уровня блока

Как говорилось выше, блочные элементы могут содержать только блочные или только строчные элементы. В случае смешанного контента, когда блочный элемент одновременно содержит текстовое содержимое и другой блочный элемент, алгоритм визуального форматирования добавляет дополнительную обёртку для текстового содержимого — так называемый анонимный бокс. Поскольку такой контейнер не имеет названия, то к нему нельзя применить CSS-стили для оформления. Анонимные боксы наследуют свойства окружающего блока, а не наследуемые свойства принимают первоначальное значение.

РИС. 2. АНОНИМНЫЕ БОКСЫ УРОВНЯ БЛОКА

4. Строчные элементы и строчные контейнеры

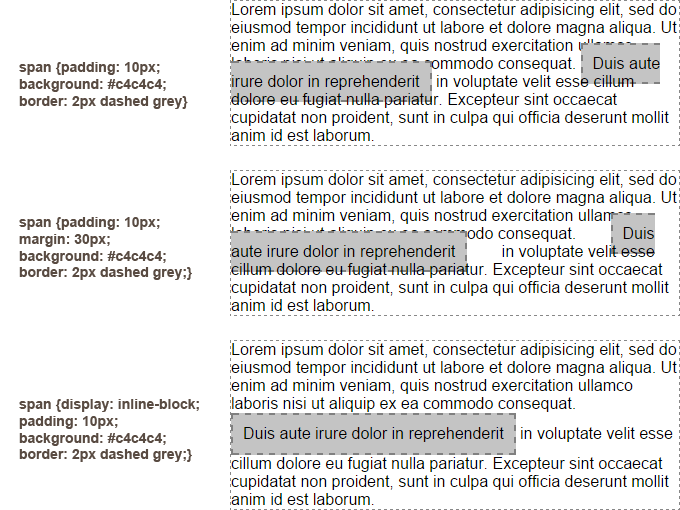
**Встроенные (строчные) элементы** генерируют внутристрочные контейнеры. Они не формируют новые блоки контента. Значения свойства display, такие как inline и inline-table делают элементы строчными.

<a>, <area>,  
<b>, <bdo>, <bdi>,  
<cite>, <code>,  
<dfn>, <del>,  
<em>,  
<i>, <img>, <ins>,  
<kbd>,  
<label>,  
<map>, <mark>,  
<s>, <samp>, <small>, <span>, <strong>, <sub>, <sup>,  
<time>,  
<q>,  
<ruby>,  
<u>,  
<var>

Строчные элементы являются потомками блочных элементов. Они игнорируют верхние и нижние margin и padding, но если для элемента задан фон, он будет распространяться на верхний и нижний padding, заходя на соседние строки текста.

Ширина и высота строчного элемента зависит только от его содержимого, задать размеры с помощью CSS нельзя. Можно увеличить расстояние между соседними элементами по горизонтали с помощью горизонтальных полей и отступов.

Для того чтобы верхние и нижние поля и отступы работали для строчного элемента, нужно использовать конструкцию {display: inline-block}. Элемент останется встроенным, но к нему можно будет полноценно применить поля, отступы, задать ширину и высоту.

РИС. 3. ОТЛИЧИЕ ВОЗДЕЙСТВИЯ CSS-СВОЙСТВ PADDING И MARGIN НА СТРОЧНЫЕ И СТРОЧНО-БЛОЧНЫЕ ЭЛЕМЕНТЫ

Строчные элементы могут содержать только данные и другие строчные элементы. Исключение составляет элемент <a>, который согласно спецификации HTML5 может оборачивать целые абзацы, списки, таблицы, заголовки и целые разделы при условии, что они не содержат другие интерактивные элементы — другие ссылки и кнопки.

Анонимные строчные боксы

Любой текст, содержащийся непосредственно внутри блочного элемента и не внутри строчного элемента, рассматривается как анонимный строчный элемент. Также как и анонимные блочные боксы они наследуют свойства родительского блока, а не наследуемые свойства принимают первоначальное значение.

https://html5book.ru/wp-content/uploads/2014/12/anonymous-inline.pngРИС. 4. АНОНИМНЫЙ СТРОЧНЫЙ БОКС

5. Строчно-блочные элементы

Существует еще одна группа элементов, которые браузер обрабатывает как строчно-блочные {display: inline-block;}. Такие элементы являются встроенным, но для них можно задавать поля, отступы, ширину и высоту.  
<audio>,  
<button>,  
<canvas>,  
<embed>,  
<input>,  
<keygen>,  
<meter>,  
<object>,  
<progress>,  
<select>,  
<textarea>,  
<video>.

6. Как сделать строчный элемент блочным, а блочный — строчным

В некоторых случаях бывает необходимо, чтобы строчный элемент вел себя как блочный и наоборот. Для этого необходимо установить соответствующее значение свойства display, например:

a {

display: block;

}/\*превращаем строчный элемент в блочный, благодаря чему весь блок, а не только текст ссылки, становится ссылкой\*/

div {

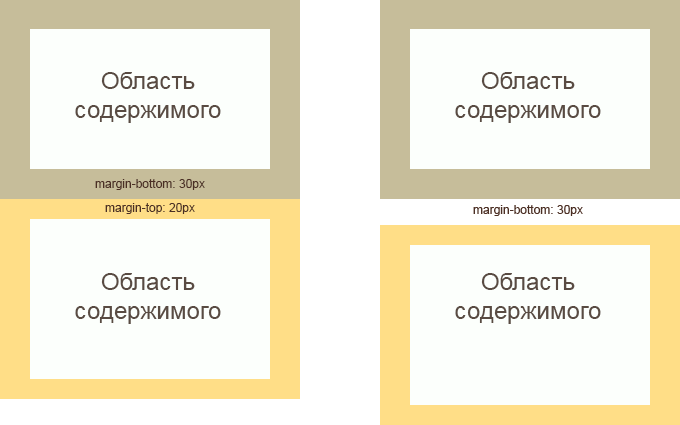
display: inline;

} /\*превращаем блочный элемент в строчный\*/

В результате меняется только способ представления элемента браузером, при этом сам элемент свой тип не меняет.

7. Схлопывание вертикальных отступов

Соприкасающиеся вертикальные отступы margin объединяются. При этом ширина общего отступа равна ширине большего из исходных отступов.

РИС. 5. ДВА БЛОКА ДО И ПОСЛЕ СЛИЯНИЯ ОТСТУПОВ

Слияние выполняется только для блочных элементов в нормальном потоке документа. Внешние вертикальные отступы строчных, плавающих и абсолютно позиционированных элементов не сливаются.

Чтобы получить желаемый промежуток, можно задать, например, для верхнего элемента padding-bottom, а для нижнего элемента — margin-top.

Если среди схлопывающихся отступов есть отрицательные значения, то браузер добавит отрицательное значение к положительному, а полученный результат и будет расстоянием между элементами.

8. Отрицательные отступы

Отрицательные отступы можно использовать, чтобы убрать пустые области между элементами. Например, необходимо расположить в ряд несколько элементов меню с заданными размерами. Для этого нужно задать для li {display: inline-block;}.

menu_inline_block

Если нужно, чтобы элементы меню прилегали друг к другу, используем отрицательный margin:

li {

margin-right: -4px;

}

li {

border-top: 1px solid #888888; /\*Убираем двойные рамки между соседними элементами\*/

border-bottom: 1px solid #888888;

border-left: 1px solid #888888;

}

li:last-child {

border-right: 1px solid #888888;

}

menu_inline_block_1

9. Выпадание вертикальных отступов

Если внутри одного блока расположить другой блок и задать ему margin-top, то внутренний блок прижмется к верхнему краю родительского, а у родительского элемента появится отступ сверху, т.е. внутренний блок «выпадет» из родительского блока. Если у родительского элемента также был задан верхний отступ, то выберется наибольшее из значений.

Чтобы избавиться от эффекта выпадания, можно задать родительскому элементу padding-top или добавить border-top: 1px solid transparent.