

Блочные и строчные элементы

Выделяют две основные категории HTML-элементов, которые соответствуют типам их содержимого и поведению в структуре веб-страницы — **блочные** и **строчные элементы**. С помощью блочных элементов можно создавать структуру веб-страницы, строчные элементы используются для форматирования текстовых фрагментов (за исключением элементов `<area>` и ``).

Разделение элементов на блочные и строчные используется в спецификации HTML до версии 4.01. В HTML5 эти понятия заменены более сложным набором **категорий контента**, согласно которым каждый HTML-элемент должен следовать правилам, определяющим, какой контент для него допустим.

1. Модель визуального форматирования

HTML-документ организован в виде дерева элементов и текстовых узлов. Модель визуального форматирования CSS представляет собой алгоритм, который обрабатывает HTML-документ и выводит его на экран устройства.

Каждый блок в дереве представляет соответствующий элемент или псевдоэлемент, а текст (буквы, цифры, пробелы), находящийся между открывающим и закрывающим тегами, представляет содержимое текстовых узлов.

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. Некоторые значения свойства `display`, например, `display: list-item;`, генерируют блок основного блока и блок дочернего маркера. Другие, например, `display: none;`, приводят к тому, что элемент и/или его потомки вообще не генерируют блоки.

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).

2. 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> ,  
<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.

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 обернуты элементом `<p>`. Блочные элементы могут содержаться только в пределах блочных элементов.

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

Анонимные блоки 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 блок. Анонимный блок не связан ни с одним элементом. Поскольку такой контейнер не имеет названия, то к нему нельзя применить CSS-стили для оформления. Анонимные блоки наследуют свойства окружающего блока, а не наследуемые свойства принимают первоначальное значение.

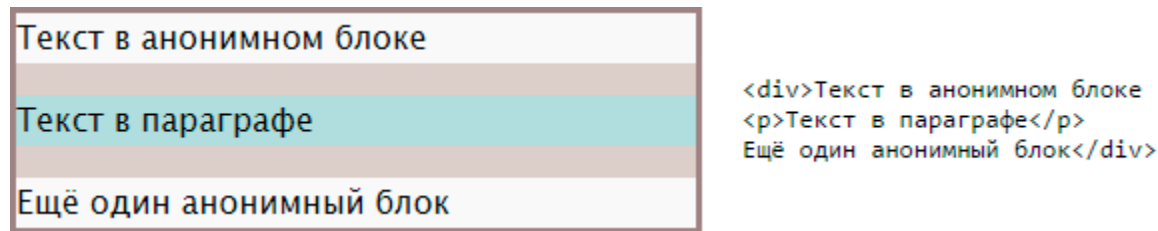


РИС. 1. АНОНИМНЫЕ БЛОКИ УРОВНЯ БЛОКА

3. 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> , <iframe> , <img> , <ins> ,
<kbd> ,
<label> ,
```

<map> , <mark> ,
 <s> , <samp> , <small> , , , <sub> , <sup> ,
 <time> ,
 <q> ,
 <ruby> ,
 <u> ,
 <var>

**span {padding: 10px;
 background: #c4c4c4;
 border: 2px dashed grey;}**

Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do
 eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut
 enim ad minim veniam, quis nostrud exercitation ullamco
 laboris nisi ut aliquip ex ea commodo consequat. Duis aute
 irure dolor in reprehenderit in voluptate velit esse cillum
 dolore eu fugiat nulla pariatur. Excepteur sint occaecat
 cupidatat non proident, sunt in culpa qui officia deserunt mollit
 anim id est laborum.

**span {padding: 10px;
 margin: 30px;
 background: #c4c4c4;
 border: 2px dashed grey;}**

Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do
 eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut
 enim ad minim veniam, quis nostrud exercitation ullamco
 laboris nisi ut aliquip ex ea commodo consequat. Duis
 aute irure dolor in reprehenderit in voluptate velit esse
 cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat
 cupidatat non proident, sunt in culpa qui officia deserunt mollit
 anim id est laborum.

**span {display: inline-block;
 padding: 10px;
 background: #c4c4c4;
 border: 2px dashed grey;}**

Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do
 eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut
 enim ad minim veniam, quis nostrud exercitation ullamco
 laboris nisi ut aliquip ex ea commodo consequat.
 Duis aute irure dolor in reprehenderit in voluptate velit esse
 cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat
 cupidatat non proident, sunt in culpa qui officia deserunt mollit
 anim id est laborum.

📷 РИС. 2. 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 блоки уровня строки

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

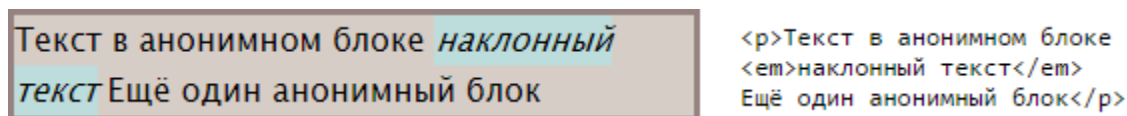


РИС. 3. АНОНИМНЫЙ БЛОК УРОВНЯ СТРОКИ

4. 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> .
```

5. Ширина содержимого: свойство width

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. Ширина линейных блоков определяется шириной содержащего блока, но может быть уменьшена из-за наличия свойства `float`.

Отрицательные значения не допускаются.

Свойство не наследуется.

width

Значения:

длина	Ширина элемента задается в единицах длины, например, <code>px</code> , <code>em</code> и т.д.
%	Вычисляется относительно ширины содержащего блока. Для абсолютно позиционированных элементов процент вычисляется с учетом ширины области отступов <code>padding</code> содержащего блока.
auto	Ширина вычисляется в зависимости от значений других свойств. Значение по умолчанию.
inherit	Наследует значение свойства от родительского элемента.

Синтаксис

CSS

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

6. Минимальная и максимальная ширина: свойства `min-width` и `max-width`

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`.

Свойства не наследуются.

min-width/max-width

Значения:

длина	Задает фиксированную минимальную или максимальную используемую ширину.
%	Указывает процент для определения используемого значения. Процент рассчитывается относительно ширины содержащего блока.
none	Означает отсутствие ограничений ширины блока.
inherit	Наследует значение свойства от родительского элемента.

Синтаксис

```
min-width: 100px;  
min-width: 10em;  
min-width: 50%;  
min-width: inherit;
```

CSS

```
max-width: 500px;  
max-width: 20em;  
max-width: 80%;  
max-width: none;  
max-width: inherit;
```

CSS

7. Высота содержимого: свойство height

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

Свойство не наследуется.

height

Значения:

длина	Высота области содержимого задается в единицах длины.
%	Задаёт высоту в процентах. Процент рассчитывается относительно высоты содержащего блока. Если высота содержащего блока не указана явно (то есть зависит от высоты содержимого) и этот элемент не является абсолютно позиционированным, значение вычисляется как <code>auto</code> . Для абсолютно позиционированных элементов процент вычисляется с учетом высоты области отступов <code>padding</code> содержащего блока.
auto	Высота зависит от значений других свойств. Значение по умолчанию.
inherit	Наследует значение свойства от родительского элемента.

Синтаксис

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

CSS

8. Минимальная и максимальная высота: свойства `min-height` и `max-height`

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

Свойства не наследуются.

min-height/max-height

Значения:

длина	Задаёт фиксированную минимальную или максимальную вычисленную высоту в единицах длины. Значения не могут быть отрицательными.
%	Указывает процент для определения используемого значения. Процент рассчитывается относительно высоты содержащего блока. Если высота содержащего блока не указана явно (т.е. зависит от высоты содержимого) и этот элемент не является абсолютно позиционированным, процентное значение обрабатывается как 0 для <code>min-height</code> или <code>none</code> для <code>max-height</code> .
none	Отсутствие ограничений высоты блока, только для <code>max-height</code> .
inherit	Наследует значение свойства от родительского элемента.

Синтаксис

```
min-height: 100px;  
min-height: 2em;  
min-height: 50%;  
min-height: inherit;
```

CSS

```
max-height: 500px;  
max-height: 20em;  
max-height: 80%;  
max-height: none;  
max-height: inherit;
```

CSS

9. Расчет высоты строки: свойства `line-height` и `vertical-align`

Как описано выше, пользовательские агенты (браузеры) передают блоки встроенного уровня в вертикальный стек линейных блоков. Высота линейного блока определяется следующим образом:

- Высота каждого встроенного прямоугольника в линейном блоке вычисляется. Для замещаемых, `inline-block` и `inline-table` элементов это высота их области поля (`margin box`).
- Блоки уровня строки выравниваются вертикально в соответствии со значением свойства `vertical-align`. Если они выровнены по верху или по низу, они должны быть выровнены так, чтобы минимизировать высоту линейного блока.

Высота линейного блока — это расстояние между самой верхней и самой нижней частью блока. Пустые встроенные элементы генерируют пустые встроенные блоки, но эти блоки по-прежнему имеют поля, отступы, границы, высоту строки и, таким образом, влияют на эти вычисления также, как и элементы с содержимым.

В элементе уровня блока, содержимое которого состоит из элементов встроенного уровня, свойство `line-height` определяет минимальную высоту линейных блоков внутри элемента. Минимальная высота состоит из минимальной высоты над базовой линией и минимальной глубины под ней.

Для элементов уровня строки свойство `line-height` указывает высоту, которая используется при расчете высоты линейного блока.

Отрицательные значения не допустимы.

Свойство наследуется.

line-height

Значения:

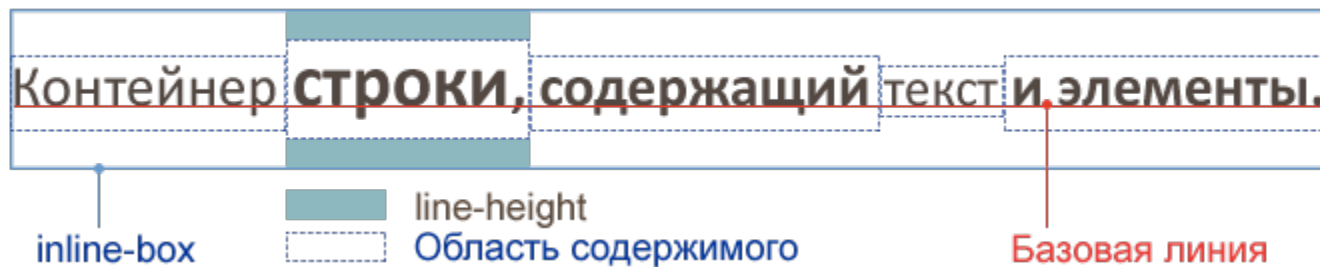
normal	Сообщает пользовательским агентам установить «разумное» значение на основе шрифта элемента. Значение по умолчанию. Когда элемент содержит текст, отображаемый более чем одним шрифтом, пользовательские агенты могут определить значение <code>normal</code> в соответствии с наибольшим размером шрифта.
длина	Значение задаётся в единицах длины, создавая фиксированное значение высоты строки. Если задать значение меньше единицы, смежные строки будут находить друг на друга.
число	Используемое значение свойства — это число, умноженное на размер шрифта элемента.

%	Вычисленное значение свойства — это процент, умноженный на вычисленный размер шрифта элемента.
inherit	Наследует значение свойства от родительского элемента.

Синтаксис

CSS

```
line-height: normal;
line-height: 2em;
line-height: 1.5;
line-height: 50%;
line-height: inherit;
```



📷 РИС. 4. КОНТЕЙНЕР СТРОКИ

Свойство `vertical-align` влияет на вертикальное позиционирование в линейном блоке элементов уровня строки: `display: inline` и `display: table-cell`. Значения этого свойства имеют другие значения в контексте таблиц.

Свойство не наследуется.

vertical-align

Значения:

`baseline` Выравнивает базовую линию элемента по базовой линии его родителя, совмещая среднюю

линию элемента со средней линией родительского элемента.


sub	Делает элемент подстрочным (аналогично с элементом <code><sub></code>). Величина понижения элемента может меняться в зависимости от браузера пользователя.
super	Делает элемент надстрочным (аналогично с элементом <code><sup></code>). При этом значения <code>sup</code> и <code>super</code> не меняют размер шрифта, по умолчанию текст надстрочного и подстрочного элемента имеет такой же размер, как и текст родительского элемента.
top	Верхний край элемента совмещается с верхним краем самого высокого элемента в линии.
text-top	Верхний край элемента совмещается с верхним краем шрифта родительского элемента.
middle	Средняя линия элемента (обычно изображения) совмещается с линией, проходящей через середину родительского элемента.
bottom	Нижний край элемента совмещается с нижним краем самого низкого элемента в линии.
text-bottom	Нижний край элемента совмещается с нижним краем шрифта родительского элемента.
%	Не позволяет устанавливать <code>middle</code> , вычисляется как часть <code>line-height</code> элемента, а не его родителя, т.е. если установить значение <code>vertical-align</code> , равное <code>50%</code> для элемента с <code>line-height</code> равным <code>20px</code> , то базовая линия элемента поднимется на <code>10px</code> .
длина	Устанавливает значение в единицах длины, перемещая элемент на заданное расстояние.
inherit	Наследует значение свойства от родительского элемента.

Синтаксис


CSS

```
vertical-align: baseline;  
vertical-align: sub;  
vertical-align: super;  
vertical-align: text-top;  
vertical-align: text-bottom;  
vertical-align: middle;  
vertical-align: top;  
vertical-align: bottom;
```


```
vertical-align: 6em;  
vertical-align: 10px;  
vertical-align: 25%;  
vertical-align: inherit;
```


Изображение  {vertical-align: baseline} не имеет базовой линии, поэтому оно выравнивается по базовой линии родителя (в данном случае, текста).

Этот абзац содержит **надстрочный** {vertical-align: super} и **подстрочный** {vertical-align: sub} текст, который имеет такой же размер, как и основной текст.

Изображение  {vertical-align: bottom} выравнивается по низу контейнера строки, смещаясь ниже базовой линии текста.

Изображение  {vertical-align: text-bottom} выравнивается относительно нижней линии текста строки.

Изображение  {vertical-align: top} выравнивается по верху контейнера строки, при этом учитывается размер шрифта родительского элемента.

В данном примере изображение  {vertical-align: text-top} выравнивается относительно верхней линии текста строки.


Значение  {vertical-align: middle} выравнивает середину изображения по точке, которая находится на расстоянии, равном половине размера шрифта родительского элемента, над базовой линией текста.

 РИС. 5. СВОЙСТВО VERTICAL-ALIGN

10. Изменение блочной модели: свойство box-sizing

Свойство box-sizing переключает блочную модель с фиксированных размеров длины и ширины на content-box и border-box. Это влияет на интерпретацию всех свойств, определяющих размеры, включая flex-basis.

Свойство не наследуется.

box-sizing

Значения:

content-box	Это поведение ширины и высоты, как указано в CSS2.1. Заданные ширина и высота (и соответствующие min/max-свойства) применяются к ширине и высоте области содержимого элемента. Поля padding и рамка border элемента располагаются за пределами указанной ширины и высоты. Значение по умолчанию.
border-box	Любые padding или border , заданные для элемента, размечаются и отрисовываются внутри указанных значений ширины и высоты. Ширина и высота содержимого вычисляются путем вычитания ширины границ и полей соответствующих сторон из указанных свойств ширины и высоты. Значение auto свойств width и height не зависит от свойства box-sizing и всегда устанавливает размер блока с содержимым. Сумма padding и border не должна превышать заданные значения width и height , в противном случае размер области содержимого будет равен нулю.
initial	Устанавливает значение свойства в значение по умолчанию.
inherit	Наследует значение свойства от родительского элемента.

Некоторые HTML-элементы, например, `<button>` , по умолчанию имеют `box-sizing: border-box` .

Синтаксис

```
box-sizing: content-box;  
box-sizing: border-box;  
box-sizing: inherit;  
box-sizing: initial;
```

CSS