

CSS-позиционирование

CSS рассматривает макет html-документа как дерево элементов. Уникальный элемент, у которого нет родительского элемента, называется **корневым** элементом. Модуль CSS-позиционирование описывает, как любой из элементов может быть размещен независимо от порядка документа (т.е. извлечен из «потока»).

В CSS2 каждый элемент в дереве документа генерирует ноль или более блоков в соответствии с блочной моделью. Модуль CSS3 дополняет и расширяет схему позиционирования. Расположение этих блоков регулируется:

- размерами и типом элемента,
- схемой позиционирования (нормальный поток, обтекание и абсолютное позиционирование),
- отношениями между элементами в дереве документа,
- внешней информацией (например, размер области просмотра, внутренними размерами изображений и т.д.).

В CSS блок элемента может быть расположен в соответствии с тремя схемами позиционирования:

Нормальный поток

Нормальный поток включает блочный контекст форматирования (элементы с `display block`, `list-item` или `table`), строчный (встроенный) контекст форматирования (элементы с `display inline`, `inline-block` или `inline-table`), и относительное и «липкое» позиционирование элементов уровня блока и строки.

Обтекание

В обтекающей модели блок удаляется из нормального потока и позиционируется влево или вправо. Содержимое обтекает правую сторону элемента с `float: left` и левую сторону элемента с `float: right`.

Абсолютное позиционирование

В модели абсолютного позиционирования блок полностью удаляется из нормального потока и ему присваивается позиция относительно содержащего блока. Абсолютное позиционирование реализуется с помощью значений `position: absolute;` и `position: fixed;`.

Элементом «вне потока» может быть плавающий, абсолютно позиционированный или корневой элемент.

1. Содержащий блок

Положение и размер блока(ов) элемента иногда вычисляются относительно некоторого прямоугольника, называемого **содержащим блоком** элемента (containing block). В общих словах, содержащий блок — это блок, который содержит другой элемент. В случае нормального потока корневой элемент `html` является содержащим блоком для элемента `body`, который, в свою очередь, является содержащим блоком для всех его дочерних элементов и так далее. В случае позиционирования содержащий блок полностью зависит от типа позиционирования. Содержащий блок элемента определяется следующим образом:

- Содержащий блок, в котором находится корневой элемент, представляет собой прямоугольник — так называемый **начальный содержащий блок**.
- Для некорневого элемента с `position: static;` или `position: relative;` содержащий блок формируется краем области содержимого ближайшего родительского блока уровня блока, ячейки таблицы или уровня строки.
- Содержащим блоком элемента с `position: fixed;` является окно просмотра.
- Для некорневого элемента с `position: absolute;` содержащим блоком устанавливается ближайший родительский элемент со значением `position: absolute/relative/fixed` следующим образом:
 - если предок — элемент уровня блока, содержащим блоком будет область содержимого плюс поля элемента `padding`;
 - если предок — элемент уровня строки, содержащим блоком будет область содержимого;
 - если предков нет, то содержащий блок элемента определяется как начальный содержащий блок.
- Для «липкого» блока содержащим блоком является ближайший предок с прокруткой или окно просмотра, в противном случае.

2. Выбор схемы позиционирования: свойство `position`

Свойство `position` определяет, какой из алгоритмов позиционирования используется для вычисления положения блока.

Свойство не наследуется.

position

Значение:

static

Блок располагается в соответствии с нормальным потоком. Свойства `top` , `right` , `bottom` и `left` не применяются. Значение по умолчанию.

relative

Положение блока рассчитывается в соответствии с нормальным потоком. Затем блок смещается относительно его нормального положения и во всех случаях, включая элементы таблицы, не влияет на положение любых следующих блоков. Тем не менее, такое смещение может привести к перекрытию блоков, а также к появлению полосы прокрутки в случае переполнения.

Относительно позиционированный блок сохраняет свои размеры, включая разрывы строк и пространство, первоначально зарезервированное для него.

Относительно позиционированный блок создает новый содержащий блок для абсолютно позиционированных потомков.

Влияние `position: relative;` на элементы таблицы определяется следующим образом: Элементы с `table-row-group` , `table-header-group` , `table-footer-group` и `table-row` смещаются относительно их обычной позиции в таблице. Если ячейки таблицы занимают несколько строк, смещаются только ячейки начальной строки.

`table-column-group` , `table-column` не смещает соответствующий столбец и не оказывает визуального влияния.

`table-caption` and `table-cell` смещаются относительно своего нормального положения в таблице. Если ячейка таблицы охватывает несколько столбцов или строк, то она смещается целиком.

absolute

Положение блока (и, возможно, размер) задается с помощью свойств `top` , `right` , `bottom` и `left` . Эти свойства определяют явное смещение относительно его содержащего блока.

Абсолютно позиционированные блоки полностью удаляются из нормального потока, не влияя на расположение сестринских элементов.

Отступы `margin` абсолютно позиционированных блоков не схлопываются.

Абсолютно позиционированный блок создает новый содержащий блок для дочерних элементов нормального потока и потомков с `position: absolute;` .

Содержимое абсолютно позиционированного элемента не может обтекать другие блоки.

	<p>Абсолютно позиционированный блок могут скрывать содержимое другого блока (или сами могут быть скрыты), в зависимости от значения <code>z-index</code> перекрывающихся блоков.</p>
sticky	<p>Положение блока рассчитывается в соответствии с нормальным потоком. Затем блок смещается относительно своего ближайшего предка с прокруткой или окна просмотра, если ни у одного из предков нет прокрутки.</p> <p>«Липкий» блок может перекрывать другие блоки, а также создавать полосы прокрутки в случае переполнения.</p> <p>«Липкий» блок сохраняет свои размеры, включая разрывы строк и пространство, первоначально зарезервированное для него.</p> <p>«Липкий» блок создает новый содержащий блок для абсолютно и относительно позиционированных потомков.</p>
fixed	<p>Фиксированное позиционирование аналогично абсолютному позиционированию, с отличием в том, что для содержащим блоком устанавливается окно просмотра. Такой блок полностью удаляется из потока документа и не имеет позиции относительно какой-либо части документа. Фиксированные блоки не перемещаются при прокрутке документа. В этом отношении они похожи на фиксированные фоновые изображения.</p> <p>При печати фиксированные блоки повторяются на каждой странице, содержащим блоком для них устанавливается область страницы. Блоки с фиксированным положением, которые больше области страницы, обрезаются.</p>
initial	Устанавливает значение свойства в значение по умолчанию.
inherit	Наследует значение свойства от родительского элемента.

Синтаксис

```
position: static;
position: relative;
position: absolute;
position: sticky;
position: fixed;
position: initial;
```

CSS

```
position: inherit;
```

Заголовок {position: absolute; top: 10px;}

Заголовок {position: static}

Заголовок с **position static** сохраняет своё место в потоке документа

Заголовок {position: relative; top: 10px;}

Заголовок с **position relative** смещается относительно своего текущего положения в макете

Заголовок с **position absolute** смещается относительно верхнего левого угла браузера. При этом его ширина становится равной ширине содержимого

📷 РИС. 1. РАЗНИЦА МЕЖДУ СТАТИЧНЫМ, ОТНОСИТЕЛЬНЫМ И АБСОЛЮТНЫМ ПОЗИЦИОНИРОВАНИЕМ

3. Смещение блока: свойства top, right, bottom, left

Элемент считается позиционированным, если свойство `position` имеет значение, отличное от `static`. Позиционированные элементы генерируют позиционированные блоки и могут быть расположены в соответствии со следующими четырьмя физическими свойствами:

top	
Значение:	
auto	Влияние значения зависит от типа элемента. Значение по умолчанию.
длина	Смещение на фиксированном расстоянии от указанного края. Отрицательные значения допускаются.
%	Процентные значения вычисляются относительно высоты содержащего блока. Для «липкого»

	блока — относительно высоты корневого элемента. Отрицательные значения допускаются.
initial	Устанавливает значение свойства в значение по умолчанию.
inherit	Наследует значение свойства от родительского элемента.

Синтаксис

	CSS
<pre> top: 10px; top: 2em; top: 50%; top: auto; top: inherit; top: initial; </pre>	

Свойство `top` задает расстояние, на которое верхний край абсолютно позиционированного блока (с учетом его `margin`) смещается ниже верхнего края содержащего блока. Для относительно позиционированных блоков определяет смещение относительно верхнего края самого блока (то есть блоку задается позиция в нормальном потоке, а затем смещение от этой позиции в соответствии с этим свойством).

right	
Значение:	
auto	Влияние значения зависит от типа элемента. Значение по умолчанию.
длина	Смещение на фиксированном расстоянии от указанного края. Отрицательные значения допускаются.
%	Процентные значения вычисляются относительно ширины содержащего блока. Для «липкого» блока — относительно ширины корневого элемента. Отрицательные значения допускаются.
initial	Устанавливает значение свойства в значение по умолчанию.
inherit	Наследует значение свойства от родительского элемента.

Синтаксис

CSS

```
right: -10px;  
right: .5em;  
right: -10%;  
right: auto;  
right: inherit;  
right: initial;
```

Свойство `right` указывает расстояние, на которое правый край абсолютно позиционированного блока (с учетом его `margin`) смещен влево от правого края содержащего блока. Для относительно позиционированных блоков определяет смещение относительно правого края самого блока.

bottom

Значение:

auto	Влияние значения зависит от типа элемента. Значение по умолчанию.
длина	Смещение на фиксированном расстоянии от указанного края. Отрицательные значения допускаются.
%	Процентные значения вычисляются относительно высоты содержащего блока. Для «липкого» блока — относительно высоты корневого элемента. Отрицательные значения допускаются.
initial	Устанавливает значение свойства в значение по умолчанию.
inherit	Наследует значение свойства от родительского элемента.

Синтаксис

CSS

```
bottom: 50px;  
bottom: -3em;  
bottom: -50%;
```

```
bottom: auto;
bottom: inherit;
bottom: initial;
```

Свойство `bottom` указывает расстояние, на которое нижний край блока смещен вверх относительно нижнего края содержащего блока. Для относительно позиционированных блоков определяет смещение относительно нижнего края самого блока.

left

Значение:

auto	Влияние значения зависит от типа элемента. Значение по умолчанию.
длина	Смещение на фиксированном расстоянии от указанного края. Отрицательные значения допускаются.
%	Процентные значения вычисляются относительно ширины содержащего блока. Для «липкого» блока — относительно ширины корневого элемента. Отрицательные значения допускаются.
initial	Устанавливает значение свойства в значение по умолчанию.
inherit	Наследует значение свойства от родительского элемента.

Синтаксис

```
left: 50px;
left: 10em;
left: 20%;
left: auto;
left: inherit;
left: initial;
```

CSS

Свойство `left` указывает расстояние, на которое левый край смещен вправо от левого края содержащего блока. Для относительно позиционированных блоков определяет смещение относительно левого края самого блока.

Положительные значения смещают элемент внутрь содержащего блока, а отрицательные — за его пределы.

4. Обтекание: свойство `float`

Обтекание позволяет блокам смещаться влево или вправо на текущей строке. «Плавающий блок» смещается влево или вправо до тех пор, пока его внешний край не коснется края содержащего блока или внешнего края другого плавающего блока. Если имеется линейный блок, внешняя верхняя часть плавающего блока выравнивается с верхней частью текущего линейного блока.

При использовании свойства `float` для элементов рекомендуется задавать ширину. Тем самым браузер создаст место для другого содержимого. Если для плавающего элемента недостаточно места по горизонтали, он будет смещаться вниз до тех пор, пока не уместится. При этом остальные элементы уровня блока будут его игнорировать, а элементы уровня строки будут смещаться вправо или влево, освобождая для него пространство и обтекая его.

Правила, регулирующие поведение плавающих боков, описываются свойством `float`.

Свойство не наследуется.

float	
Значение:	
none	Отсутствие обтекания. Значение по умолчанию.
left	Элемент перемещается влево, содержимое обтекает плавающий блок по правому краю.
right	Элемент перемещается вправо, содержимое обтекает плавающий блок по левому краю.
inherit	Наследует значение свойства от родительского элемента.

Синтаксис

```
float: left;  
float: right;  
float: none;  
float: inherit;
```

Плавающий блок принимает размеры своего содержимого с учетом внутренних отступов и рамок. Верхние и нижние отступы `margin` плавающих элементов не схлопываются.

Плавающие элементы могут использовать отрицательные отступы `margin`, чтобы перемещаться за пределы области содержимого их родительского элемента.

Свойство автоматически изменяет вычисляемое (отображаемое в браузере) значение свойства `display` на `display: block` для следующих значений: `inline`, `inline-block`, `table-row`, `table-row-group`, `table-column`, `table-column-group`, `table-cell`, `table-caption`, `table-header-group`, `table-footer-group`. Значение `inline-table` меняет на `table`.

Свойство не оказывает влияние на элементы с `display: flex` и `display: inline-flex`. Не применяется к абсолютно позиционированным элементам.

БЛОК 1. Этот блок отображается в нормальном нисходящем потоке.

БЛОК 2. Для этого блока задано обтекание по левому краю.

БЛОК 3. Этот блок также отображается в нормальном нисходящем потоке. Он игнорирует плавающий блок, а текст в блоке его обтекает.

РАЗМЕТКА:

```
<div style="height:150px; border:1px dashed #8bc63e">БЛОК 1.</div>
<div style="height:200px; width:300px; border:3px dashed #e03c32; float:left">БЛОК 2.</div>
<div style="height:150px; border:1px solid #888888">БЛОК 3.</div>
```

 РИС. 2. ОБТЕКАНИЕ ЭЛЕМЕНТОВ

5. Управление потоком рядом с плавающими элементами: свойство clear

Свойство `clear` указывает, какие стороны блока/блоков элемента не должны прилегать к плавающим блокам, находящемуся выше в исходном документе. В CSS2 и CSS 2.1 свойство применяется только к неплавающим элементам уровня блока.

Свойство не наследуется.

clear	
Значение:	
none	Означает отсутствие ограничений на положение элемента относительно плавающих блоков. Значение по умолчанию.
left	Смещает элемент вниз относительно нижнего края любого плавающего слева элемента, находящегося выше в исходном документе.
right	Смещает элемент вниз относительно нижнего края любого плавающего справа элемента, находящегося выше в исходном документе.
both	Смещает элемент вниз относительно нижнего края любого плавающего слева и справа элемента, находящегося выше в исходном документе.
inherit	Наследует значение свойства от родительского элемента.

Синтаксис

```
clear: none;
clear: left;
```

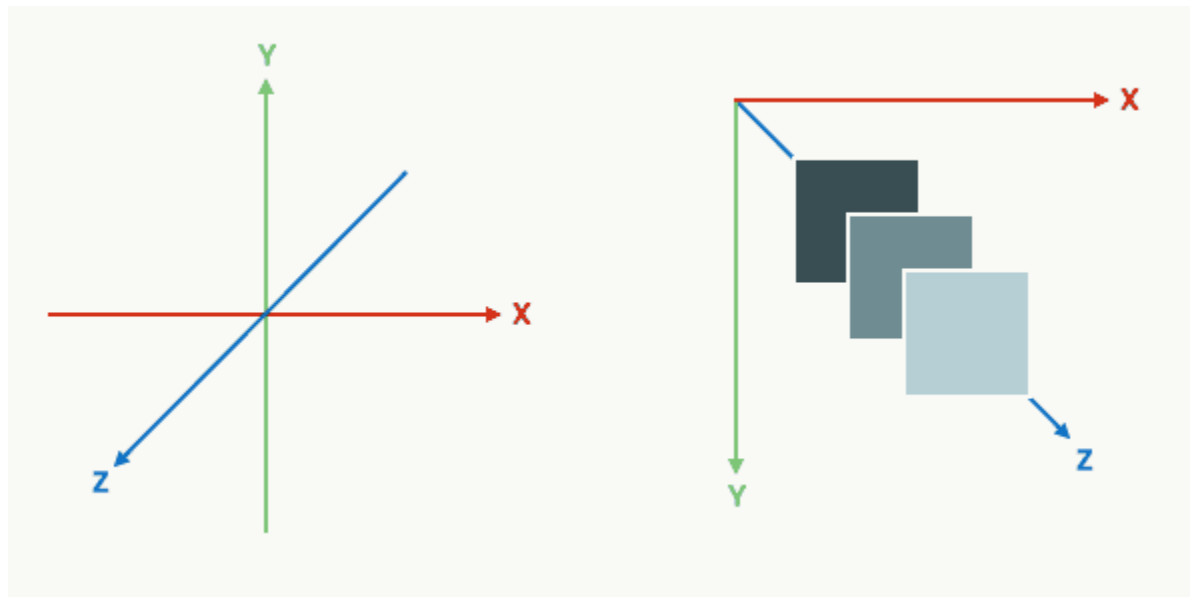
CSS

```
clear: right;  
clear: both;  
clear: inherit;
```

Для предотвращения отображения фона или границ под плавающими элементами используется правило `{overflow: hidden;}`.

6. Определение контекста наложения: свойство `z-index`

В CSS каждый блок имеет позицию в трех измерениях. В дополнение к горизонтальному и вертикальному положению, блоки выкладываются вдоль оси *Z* друг над другом. Положение вдоль оси *Z* особенно важно, когда блоки визуальнo накладываются друг на друга.



📷 РИС. 3. ПОЛОЖЕНИЕ ЭЛЕМЕНТОВ ВДОЛЬ ОСИ *Z*

Порядок, в котором дерево документа отрисовывается на экране, описывается с помощью **контекста наложения**. Каждый блок принадлежит одному контексту наложения. Каждый блок в данном контексте наложения имеет целочисленный уровень, который является его положением на оси *Z* относительно других блоков в том же контексте наложения.

Блоки с более высокими уровнями всегда отображаются перед блоками с более низкими уровнями, а блоки с одинаковым уровнем располагаются снизу вверх в соответствии с порядком следования элементов в исходном документе. Блок элемента имеет ту же позицию, что и блок его родителя, если только ему не присвоен другой уровень свойством `z-index`.

Свойство `z-index` позволяет изменить порядок наложения позиционированных элементов в случае, когда они накладываются друг на друга.

Свойство не наследуется.

z-index

Значение:

<code>auto</code>	Вычисляется в <code>0</code> . Если для блока задано <code>position: fixed;</code> или это корневой элемент, значение <code>auto</code> также устанавливает новый контекст наложения. Значение по умолчанию.
целое число	Определяет положение блока в текущем контексте наложения. Также устанавливает новый локальный контекст наложения. Можно использовать любое целое число, включая отрицательные числа. Отрицательные значения помещают элемент вглубь экрана.
<code>inherit</code>	Наследует значение свойства от родительского элемента.
<code>initial</code>	Устанавливает значение свойства в значение по умолчанию.

Синтаксис

```
z-index: auto;  
z-index: 0;  
z-index: 5;  
z-index: 999;  
z-index: -1;  
z-index: inherit;  
z-index: initial;
```

CSS

6.1. Контекст наложения

Если для элементов свойства `z-index` и `position` не заданы явно, контекст наложения равен порядку их расположения в исходном коде и браузер отображает элементы на странице в следующем порядке:

- Корневой элемент `<html>` , который содержит все элементы веб-странице.
- Блочные элементы, неплавающие и непозиционированные.
- Плавающие `float` непозиционированные элементы в порядке их расположения в исходном коде.
- Строковые непозиционированные элементы (текст, изображения).
- Позиционированные `position` элементы в порядке их следования в исходном коде. Последний из них будет расположен на переднем плане.

Свойство `z-index` создает новый контекст наложения. Оно позволяет изменить порядок наложения **позиционированных элементов**. Элементы будут отображаться на странице в следующем порядке (если для них не заданы свойства, влияющие на контекст наложения — `opacity` , `filter` , `transform`):

- Корневой элемент `<html>` , который содержит все элементы веб-странице.
- Позиционированные элементы с отрицательным значением `z-index` .
- Блочные элементы, неплавающие и непозиционированные.
- Плавающие `float` непозиционированные элементы в порядке их расположения в исходном коде.
- Строковые непозиционированные элементы (текст, изображения).
- Позиционированные элементы со значениями `z-index: 0;` и `z-index: auto;` .