Для вёрстки страниц часто используются два основных инструмента — *позиционирование (positioning)* и *свободное перемещение (floating)*. CSS-позиционирование позволяет указать, где появится блок элемента, а свободное перемещение перемещает элементы к левому или правому краю блока-контейнера, позволяя остальному содержимому «обтекать» его.

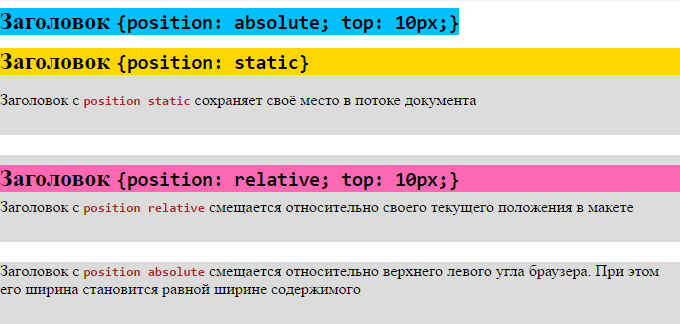
Позиционирование и свободное перемещение элементов

* **Содержание:**
* [1. Типы позиционирования](https://html5book.ru/css-position/#position)
* [2. Свойства смещения](https://html5book.ru/css-position/#position-prop)
* [3. Позиционирование внутри элемента](https://html5book.ru/css-position/#part3)
* [4. Проблемы позиционирования](https://html5book.ru/css-position/#part4)
* [5. Свободное перемещение элементов](https://html5book.ru/css-position/#float)
* [6. Отмена обтекания элементов](https://html5book.ru/css-position/#part6)

1. Типы позиционирования

Свойство position позволяет точно задать новое местоположение блока относительно того места, где он находился бы в нормальном потоке документа. По умолчанию все элементы располагаются последовательно один за другим в том порядке, в котором они определены в структуре HTML-документа. Свойство не наследуется.

|  |  |
| --- | --- |
| **position** | |
| Значение: |  |
| static | Значение по умолчанию, означает отсутствие позиционирования. Элементы отображаются последовательно один за другим в том порядке, в котором они определены в HTML-документе. Используется для очистки любого другого значения позиционирования. |
| relative | Относительно позиционированный элемент сдвигается со своего обычного места в разных направлениях относительно границ родительского контейнера, а пространство, которое он занимал, не исчезает. При этом такой элемент может перекрывать другое содержимое на странице.  Если для относительно позиционированного элемента одновременно задать свойства top и bottom или left и right, то в первом случае сработает только top, во втором — left.  Относительное позиционирование позволяет задавать z-index для элемента, а также абсолютно позиционировать дочерние элементы внутри блока. |
| absolute | Абсолютно позиционированный элемент полностью удаляется из потока документа и позиционируется относительно границ его блока-контейнера (другого элемента или окна браузера). Блок-контейнер для абсолютно позиционированного элемента — ближайший элемент-предок, значение свойства position которого не равно static.  Местоположение краёв элемента определяется с помощью свойств смещения. Пространство, которое занимал такой элемент, схлопывается, как будто элемента не существовало на странице. Абсолютно позиционированный элемент может перекрывать другие элементы или быть перекрытым ими (за счёт свойства z-index). Любой абсолютно позиционированный элемент генерирует блок, то есть принимает значение display: block;. |
| fixed | Фиксирует элемент в указанном месте страницы. Блоком-контейнером фиксированного элемента является окно просмотра, то есть элемент всегда фиксируется относительно окна браузера и не меняет своего положения во время прокрутки страницы. Сам элемент при этом полностью удаляется из основного потока документа и создаётся в новом потоке документа. |
| initial | Устанавливает значение свойства в значение по умолчанию. |
| inherit | Наследует значение свойства от родительского элемента. |

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

2. Свойства смещения

Свойства описывают смещение относительно ближайшей стороны блока-контейнера. Задаются для элементов, для которых значение свойства position не равно static. Могут принимать как положительные, так и отрицательные значения. Не наследуются.

|  |  |
| --- | --- |
| **top, right, bottom, left** | |
| Значение: |  |
| auto | Значение по умолчанию. Вычисляемое значение свойства равно нулю. |
| длина | Смещение задаётся в единицах длины. |
| % | Процентные значения вычисляются относительно высоты блока-контейнера для top и bottom и ширины блока-контейнера для right и left. |
| initial | Устанавливает значение свойства в значение по умолчанию. |
| inherit | Наследует значение свойства от родительского элемента. |

Для свойства top положительные значения перемещают верхний край позиционируемого элемента ниже, а отрицательные — выше верхнего края его блока-контейнера. Для свойства left положительные значения сдвигают край позиционируемого элемента вправо, а отрицательные значения — влево. То есть, положительные значения смещают элемент внутрь блока-контейнера, а отрицательные — за его пределы.

3. Позиционирование внутри элемента

Для блока-контейнера абсолютно позиционированного элемента задаётся свойство position: relative без смещений. Это позволяет позиционировать элементы внутри элемента-контейнера.

<div class="wrap">

<div class="white">

</div>

</div>

.wrap {

padding: 10px;

height: 150px;

position: relative;

background: #e7e6d4;

text-align: right;

border: 3px dashed #645a4e;

}

.white {

position: absolute;

width: 200px;

top: 10px;

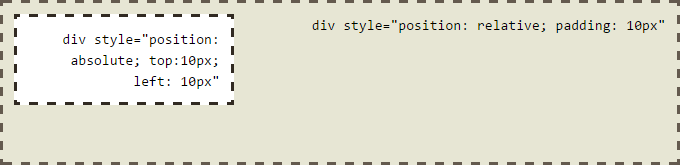
left: 10px;

padding: 10px;

background: #ffffff;

border: 3px dashed #312a22;

}

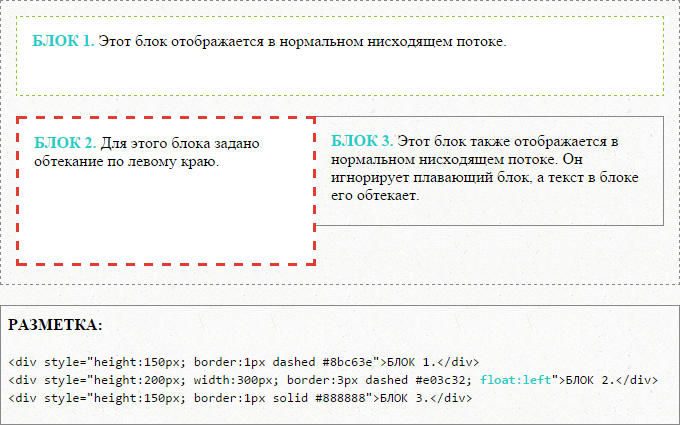
РИС. 2. АБСОЛЮТНОЕ ОТНОСИТЕЛЬНОЕ ПОЗИЦИОНИРОВАНИЕ

4. Проблемы позиционирования

1. Если ширине или высоте абсолютно позиционированного элемента присвоено значение auto, то её значение будет определяться шириной или высотой содержимого элемента. Если ширина или высота объявлена явно, то именно это значение и будет присвоено.  
2. Если внутри блока с position: absolute расположены элементы, для которых задано обтекание float, то высота этого элемента будет равна высоте самого высокого из этих элементов.  
3. Для элемента с position: absolute или position: fixed нельзя одновременно устанавливать свойство float, а для элемента с position: relative — можно.  
4. Если предок позиционированного элемента является блочным элементом, то блок-контейнер формируется областью содержимого, ограниченной рамкой (border). Если предок — строковый элемент, блок-контейнер формируется внешней границей его содержимого. Если предка нет, блоком-контейнером является элемент body.

5. Свободное перемещение элементов

В обычном порядке блочные элементы отображаются, начиная с верхнего края окна браузера по направлению к нижнему краю. Свойство float позволяет перемещать любой элемент, выравнивая его по левому или правому краю веб-страницы или содержащего его элемента-контейнера. При этом остальные блочные элементы будут его игнорировать, а строчные элементы будут смещаться вправо или влево, освобождая для него пространство и обтекая его.

РИС. 3. СВОБОДНОЕ ПЕРЕМЕЩЕНИЕ ЭЛЕМЕНТОВ

Плавающий блок принимает размеры своего содержимого с учетом внутренних отступов и рамок. Верхние и нижние отступы margin плавающих элементов не схлопываются. Свойство float применяется как к блочным элементам, так и к строчным элементам.

Левый или правый внешний край перемещаемого элемента, в отличие от позиционированных элементов, не может располагаться левее (или правее) внутреннего края его блока-контейнера, т.е. выходить за его границы. При этом, если для блока-контейнера заданы внутренние отступы, то плавающий блок будет отстоят от края блока-контейнера на заданное расстояние.

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 никакого влияния на элементы с display: flex и display: inline-flex.

|  |  |
| --- | --- |
| **float** | |
| Значение: |  |
| none | Значение по умолчанию. Также отменяет любое перемещение для элемента из группы элементов, для которых уже установлено обтекание. |
| left | Элемент изымается из нормального потока элементов и позиционируется по левому краю блока-контейнера. |
| right | Элемент позиционируется по правому краю блока-контейнера. |
| initial | Устанавливает значение свойства в значение по умолчанию. |
| inherit | Наследует значение свойства от родительского элемента. |

При использовании свойства float для блочных элементов обязательно задавать ширину. Тем самым браузер создаст место для другого содержимого. Но если совокупная ширина всех столбцов окажется больше доступного места, то последний элемент спустится вниз.

При этом вертикальные отступы margin обтекаемых элементов не схлопываются с отступами соседних элементов, в отличие от обычных блочных элементов.

6. Отмена обтекания элементов

6.1. Свойство clear

Свойство clear определяет, как будет располагаться элемент, идущий следом за плавающим элементом. Свойство отменяет обтекание с одной или обоих сторон элемента, установленное свойством float. Для предотвращения отображение фона или границ под плавающими элементами используется правило {overflow: hidden;}.

|  |  |
| --- | --- |
| **clear** | |
| Значение: |  |
| left | Отменяет обтекание по левой стороне, смещая элемент вниз. Правое обтекание остается в силе. |
| right | Отменяет обтекание по правой стороне, смещая элемент вниз. Левое обтекание остается в силе. |
| both | Отменяет обтекание с обеих сторон элемента, смещая его вниз. |
| none | Значение по умолчанию. Также отменяет очистку обтекания, установленное для элементов одной группы. |
| initial | Устанавливает значение свойства в значение по умолчанию. |
| inherit | Наследует значение свойства от родительского элемента. |

6.2. Очистка потока стилями при помощи класса clearfix и псевдокласса :after

Предположим, имеется блок-контейнер, для которого не заданы ширина и высота. Внутри него помещен плавающий блок с заданными размерами.

<div class="container">

<div class="floatbox">

</div>

</div>

.container {

padding: 20px;

background: #e7e6d4;

border: 3px dashed #645a4e;

}

.floatbox {

float: left;

width: 300px;

height: 150px;

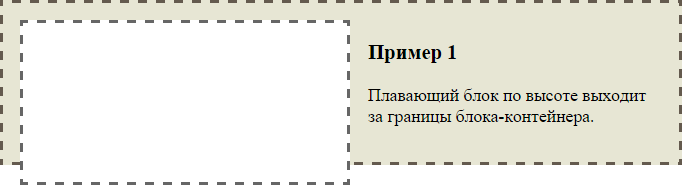
margin-right: 20px;

padding: 0 20px;

background: #ffffff;

border: 3px dashed #666666;

}

РИС. 4. «ЭФФЕКТ СХЛОПЫВАНИЯ» БЛОКА-КОНТЕЙНЕРА

**Решение проблемы:**  
Создаем класс .clearfix и в сочетании с псевдоклассом :after применяем его к блоку-контейнеру.

<div class="container clearfix">

<div class="floatbox">

</div>

</div>

.container {

padding: 20px;

background: #e7e6d4;

border: 3px dashed #645a4e;

}

.floatbox {

float: left;

width: 300px;

height: 150px;

margin-right: 20px;

padding: 0 20px;

background: #ffffff;

border: 3px dashed #666666;

}

.clearfix:after {

content: "";

display: block;

height: 0;

clear: both;

visibility: hidden;

}

**Второй вариант очистки потока:**

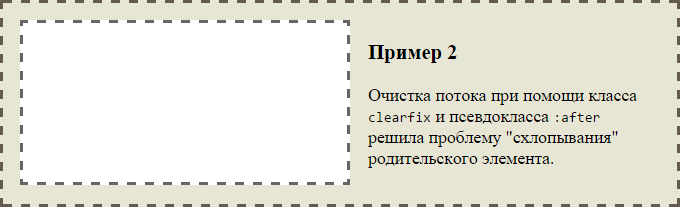
.clearfix:after {

content: "";

display: table;

clear: both;

}

РИС. 5. ПРИМЕНЕНИЕ «ОЧИЩАЮЩЕГО» МЕТОДА ДЛЯ БЛОКА-КОНТЕЙНЕРА, СОДЕРЖАЩЕГО ПЛАВАЮЩИЙ ЭЛЕМЕНТ

6.3. Легкий способ очистки потока

Существует ещё один прием очистки потока для элемента, содержащего плавающие элементы, например, для маркированного горизонтального списка:

<ul>

<li></li>

<li></li>

<li></li>

</ul>

ul {

margin: 0;

list-style: none;

padding: 20px;

background: #e7e6d4;

overflow: auto;

}

li {

float: left;

width: calc(100% / 3 - 20px);

height: 50px;

margin-right: 20px;

background: #ffffff;

border: 3px dashed #666666;

}

li:last-child {margin-right: 0;}

РИС. 6. ОЧИСТКА ПОТОКА ГОРИЗОНТАЛЬНОГО СПИСКА