



## Урок 1

# Позиционирование элементов.

## Практическая верстка

Позиционирование элементов. Расположение элементов на странице (позиционирование по вертикали, по горизонтали).

[Типы позиционирования](#)

[Свойства смещения](#)

[Позиционирование внутри элемента](#)

[Проблемы позиционирования](#)

[Свободное перемещение элементов](#)

[Отмена обтекания элементов](#)

[Свойство clear](#)

[Очистка потока стилями при помощи класса clearfix и псевдокласса :after](#)

[Домашнее задание](#)

[Дополнительные материалы](#)

[Используемая литература](#)

# Типы позиционирования

Для верстки 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.

Свойство **position** задает новое положение блоков относительно того места, где они находились бы в документе. Все элементы на странице располагаются в соответствии со структурой html, последовательно друг за другом.

| position |  |
|----------|--|
| Значение | Описание   |
| static   | Значение по умолчанию, означает отсутствие позиционирования. Элементы отображаются один за другим в том порядке, в котором они определены в HTML-документе   |
| relative | Относительно позиционированный элемент сдвигается со своего обычного места, а пространство, которое он занимал, не исчезает. При этом такой элемент может перекрывать другое содержимое на странице.<br>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  |
| absolute | Абсолютно позиционированный элемент полностью удаляется из потока документа и позиционируется относительно его блока-контейнера (другого элемента или окна браузера). Блок-контейнер для абсолютно позиционированного элемента – ближайший элемент-предок, значение свойства position которого не равно static.<br>Местоположение краев 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 счет свойства z-index). Любой абсолютно позиционированный элемент генерирует блок, то есть принимает значение display: block; |
| fixed    | Фиксирует элемент в нужном месте страницы. Блок-контейнером фиксированного элемента является окно просмотра, при этом элемент полностью удаляется из потока документа  |
| initial  | Устанавливает значение свойства в значение по умолчанию  |
| inherit  | Наследуется от родительского элемента  |

## Свойства смещения

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

| position |  |
|----------|--|
| Значение | Описание   |
| Auto     | Значение по умолчанию. Вычисляемое значение свойства равно нулю  |
| %        | 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 – влево. То есть положительные значения смещают элемент внутрь блока-контейнера, а отрицательные – за его пределы.

## Позиционирование внутри элемента

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

Разметка HTML:

```
<div class="wrap">
<div class="white">
</div>
</div>
```

CSS-стили:

```
.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: #fff;
border: 3px dashed #312a22;
}
```

## Проблемы позиционирования

Если ширине или высоте абсолютно позиционированного элемента присвоено значение `auto`, ее значение будет определяться шириной или высотой содержимого элемента. Если ширина или высота объявлена явно, то именно это значение и будет присвоено.

Если внутри блока с `position: absolute` расположены элементы, для которых задано обтекание `float`, высота этого элемента будет равна высоте самого высокого из этих элементов.

Для элемента с `position: absolute` нельзя одновременно устанавливать свойство `float`, а для элемента с `position: relative` можно.

Если предок позиционированного элемента является блочным элементом, 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`.

## 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.

```
<div style="height:150px; border: 3px dashed #E7373A;"> <p>Блок 1 </p>
</div>
<div style="height: 200px; width: 300px; border: 3px dashed #C7BDBD;
float:left; margin: 15px;"> <p> Блок 2</p> </div>
<div style="height:150px; border: 3px solid #2AC1C8; "> <p> Блок 3</p>
</div>
<div style="clear:both"> </div>
```

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, 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 отстоять 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` меняется на `table`.

Свойство не оказывает никакого влияния на элементы с `display: flex` и `display: inline-flex`.

| float    |   |
|----------|---|
| Значение | Описание  |
| 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. Если совокупная ширина всех столбцов окажется меньше 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.

# Отмена обтекания элементов

## Свойство clear

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

| clear    |   |
|----------|---|
| Значение | Описание  |
| 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     | Значение по умолчанию. Также отменяет очистку обтекания, установленного для элементов одной группы  |
| None     | Значение по умолчанию. Также отменяет очистку обтекания, установленного для элементов одной группы. |
| initial  | Устанавливает значение свойства в значение по умолчанию   |
| inherit  | Наследует значение свойства от родительского элемента   |

## 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: 1em;
background: #e7e6d4;
border: 3px dashed #645a4e;
}
.floatbox {
float: left;
width: 300px;
height: 150px;
margin-right: 1em;
padding: 0 1em;
background: #fff;
border: 3px dashed #666666;
}
```

В результате создается эффект схлопывания, когда плавающий блок выходит за пределы блока-контейнера.

Для решения 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: 1em;
background: #e7e6d4;
border: 3px dashed #645a4e;
}
.floatbox {
float: left;
width: 300px;
height: 150px;
margin-right: 1em;
padding: 0 1em;
background: #fff;
border: 3px dashed #666666;
}
.clearfix:after {
content: "";
display: block;
height: 0;
clear: both;
visibility: hidden;
}
```

## Домашнее задание

1. Ознакомиться с макетами интернет-магазина, прикрепленного к данному уроку.
2. Создать главную страницу нашего интернет-магазина (1\_index.psd).
3. \* У учеников разное количество времени и возможностей, а верстка занимает большую часть свободного времени. Поэтому, если справились с главной страницей, можно приступить к созданию страницы товара каталога.

## Дополнительные материалы

Для проверки HTML-кода на валидность (ошибки): <https://validator.w3.org/>.

Modernizr: <https://modernizr.com/>.

Первый взгляд на Avocode: <https://habrahabr.ru/post/231381/>.

Игра для изучения flexbox: <https://flexboxfroggy.com/>.

## Используемая литература

Для подготовки данного методического пособия были использованы следующие ресурсы:

1. <http://www.wisdomweb.ru/>.
2. <http://html5book.ru/>.
3. Гоше Х. HTML5. Для профессионалов. – СПб. : Питер, 2013. – 496 с.: ил.
4. Хоган Б. HTML5 и CSS3. Веб-разработка по стандартам нового поколения. – СПб. : Питер, 2012. – 272 с.
5. Макфарланд Д. Большая книга CSS3. – СПб. : Питер, 2016. – 608 с.