



Урок 2

Позиционирование элементов.

Практическая верстка

Позиционирование элементов, расположение элементов на странице (позиционирование элементов по вертикали и по горизонтали)

[Введение](#)

[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](#)

[Практическое задание](#)

[Используемая литература](#)

Введение

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.

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	Описание
static	Значение по умолчанию, означает отсутствие позиционирования. Элементы отображаются последовательно 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. 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. 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 становится блочным, то есть принимает значение display: block
fixed	Фиксирует элемент в нужном месте страницы. Блоком-контейнером фиксированного элемента является окно просмотра, при этом элемент полностью удаляется из потока документа
initial	Задаёт значение свойства по умолчанию
inherit	Элемент наследует значение свойства от родительского элемента

Свойства смещения

Эти 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.

position	
Значение	Описание
auto	Значение по умолчанию. Вычисляемое значение свойства равно нулю
%	Процентные значения вычисляются относительно высоты блока-контейнера для <code>top</code> и <code>bottom</code> и ширины блока-контейнера для <code>right</code> и <code>left</code>
initial	Задаёт значение свойства по умолчанию
inherit	Элемент наследует значение свойства от родительского элемента

Положительные значения свойства `top` смещают верхний край позиционируемого элемента ниже, а отрицательные – выше верхнего края его блока-контейнера. Положительные значения свойства `left` сдвигают левый край 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, то его размер по горизонтали или вертикали будут определяться шириной или высотой содержимого элемента. Если же ширина или высота объявлены явно, то именно такой размер и будет иметь элемент.

Если внутри блока с position: absolute расположены элементы, для которых задано обтекание float, то высота этого блока будет равна наибольшей высоте внутреннего элемента.

Для элемента с position: absolute нельзя задавать свойство float, а для элемента с position: relative – можно.

Если предок позиционированного элемента – блочный элемент, то блок-контейнер формируется областью содержимого, ограниченной рамкой (border). Если предок – строчный 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 друг за другом сверху вниз. 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 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 меняет 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	Задаёт значение свойства по умолчанию
inherit	Элемент наследует значение свойства от родительского элемента

При использовании свойства float для блочных элементов нужно обязательно задавать ширину. Тогда рядом с элементом появится 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 отступов 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. Для предотвращения отображения 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	Отменяет обтекание с обеих сторон элемента, смещая его вниз
none	Значение по умолчанию. Отменяет действие свойства clear, при этом обтекание элемента происходит, как задано с помощью свойства float или других настроек. Также отменяет очистку обтекания, установленную для элементов одной группы
initial	Задаёт значение свойства по умолчанию

inherit	Элемент наследует значение свойства от родительского элемента
---------	---

Очистка потока стилями при помощи класса clearfix и псевдокласса :after

Предположим, имеется блок-контейнер, для которого не заданы ни ширина, ни высота. Внутри него есть 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;
}
```

Такой блок-контейнер схлопывается, и плавающий блок выходит за его пределы.

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;  
}
```

Практическое задание

1. Завершить работу над выбранным макетом.
2. Проверить страницу на кроссбраузерность.
3. Использовать `position: absolute`; только в исключительных случаях.
4. Проверить работу страницы при увеличении и уменьшении масштаба.
5. * У всех учеников разные возможности и разное количество времени, которое они могут посвятить домашней работе, но чем больше работ у верстальщика в портфолио, тем лучше. Поэтому, если вы справились с одним макетом, можно браться за новый.

Используемая литература

1. <http://www.wisdomweb.ru/>
2. <http://html5book.ru/>
3. Гоше Х. HTML5. Для профессионалов. СПб.: Питер, 2013.
4. Хоган Б. HTML5 и CSS3. Веб-разработка по стандартам нового поколения. СПб.: Питер, 2012.
5. Макфарланд Д. Большая книга CSS3. СПб.: Питер, 2014.