



УРОК №11

Позиционирова ние



Позиционирование

- .Позиционирование позволяет вам изымать элементы из нормального потока макета документа и заставить их вести себя по-другому; например, располагаться друг на друге или всегда оставаться на одном и том же месте внутри окна просмотра браузера.
- .Существует несколько разных типов позиционирования, которые вы можете применить к элементам HTML. Для активации специфического типа позиционирования у элемента, мы используем свойство [position](#).

Статическое позиционирование

Статическое позиционирование — это умолчание, которое получает каждый элемент, что всего лишь значит "поставить элемент в его нормальное положение в потоке макета документа — ничего особенного для рассмотрения".

Html:

```
<p class="positioned"> ... </p>
```

Css:

```
.positioned { position: static; background: yellow; }
```

Относительное позиционирование

- Относительное позиционирование первый тип позиции, который мы рассмотрим. Оно очень похоже на статическое позиционирование, за исключением того что вы можете модифицировать окончательное положение позиционируемого объекта занявшего своё место в макете нормального потока, в том числе заставляя его перекрывать другие элементы на странице.

- **Css:**
- `position: relative;`
- Если вы сохраните и обновите на данном этапе, в результате вы совсем не увидите изменений.
- Вам необходимо использовать свойства `top`, `bottom`, `left` и `right`, используются вместе с `position` чтобы указать куда именно перемещать позиционируемый элемент
- **Css**
- `top: 30px;`
- `left: 30px;`

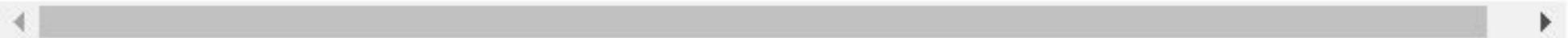
Relative positioning

I am a basic block level element. My adjacent block level elements sit on new lines below me.

By default we span 100% of the width of our parent element, and we are as tall as our child content. Our total width and height is our content + padding + border width/height.

We are separated by our margins. Because of margin collapsing, we are separated by the width of one of our margins, not both.

inline elements **like this one** and **this one** sit on the same line as one another, and adjacent text nodes, if there is space on the same line. Overflowing inline elements **wrap onto a new line if possible — like this one containing text**, or just go on to a new line if not, much like this image will do:



Абсолютное позиционирование

- Абсолютно позиционированный элемент больше не существует в нормальном потоке макета документа. Вместо этого он располагается на своём собственном слое отдельно от всего остального,
- Определяет расстояние, на котором элемент должен находиться от каждой из сторон содержащего элемента

Css:

```
position: absolute;
```

Absolute positioning

By default we span 100% of the width of our parent element, and we are as tall as our child content. Our total width and height is our content + padding + border width/height.

I am a basic block level element. My adjacent block level elements sit on new lines below me.

We are separated by our margins. Because of margin collapsing, we are separated by the width of one of our margins, not both.

inline elements like this one and this one sit on the same line as one another, and adjacent text nodes, if there is space on the same line. Overflowing inline elements wrap onto a new line if possible — like this one containing text, or just go on to a new line if not, much like this image will do:



Контекст позиционирования

.Если никакие из элементов предков не имеют конкретно заданного свойства позиции, то по умолчанию все элементы предков будут иметь статическую позицию. В результате этого абсолютно позиционируемый элемент будет содержаться в **исходным содержащем блоке**. Исходный содержащий блок имеет размеры области просмотра, а также является блоком, содержащим элемент [<html>](#). Проще говоря, абсолютно позиционируемый элемент будет отображаться за пределами элемента [<html>](#) и будет расположен относительно исходного окна просмотра.

.Если никакие из элементов предков не имеют конкретно заданного свойства позиции, то по умолчанию все элементы предков будут иметь статическую позицию. В результате этого абсолютно позиционируемый элемент будет содержаться **в исходным содержащем блоке**. Исходный содержащий блок имеет размеры области просмотра, а также является блоком, содержащим элемент <html>.

.Проще говоря, абсолютно позиционируемый элемент будет отображаться за пределами элемента <html> и будет расположен относительно исходного окна просмотра.

.position: relative;

z-index

• У веб-страниц имеется z-ось: воображаемая линия, которая направляется от поверхности вашего экрана к вашему лицу (или что ещё вам нравится иметь перед экраном).

• Значения [z-index](#) влияют на то где позиционируемый элемент располагается на этой оси;

По умолчанию все позиционируемые элементы имеют z-index auto, что фактически равно 0.

[z-index: 1;](#)

z-index

I am a basic block level element. My adjacent block level elements sit on new lines below me.

Now I'm absolutely positioned relative to the <body> element, not the <div> element.

We are separated by our margins. Because of margin collapsing, we are separated by the width of one of our margins, not both.

inline elements **like this one** and **this one** sit on the same line as one another, and adjacent text nodes, if there is space on the same line. Overflowing inline elements **wrap onto a new line if possible — like this one containing text**, or just go on to a new line if not, much like this image will do:



Используя z-index 2 и 3, вы получите тот же эффект что и 300 и 40000.

Фиксированное позиционирование

- Оно работает точно также как и абсолютное позиционирование, одним ключевым отличием:
- в то время как абсолютное позиционирование фиксирует элемент в месте относительно его ближайшего позиционированного предка (исходный содержащий блок если нет иного),
- **фиксированное позиционирование** *обычно* фиксирует элемент в месте относительно видимой части области просмотра
- Это значит, что вы можете создать элементы пользовательского интерфейса, которые зафиксированы на месте, как постоянные меню навигации, которые всегда видимы вне зависимости от того сколько прокручивается страница.

position: sticky

- это гибрид относительной и фиксированной позиции, который позволяет позиционируемому элементу вести себя как будто он относительно позиционирован, до тех пор пока он не будет прокручен до определённой пороговой точки (например, 10px от вершины окна просмотра), после чего он становится фиксированным. Это можно использовать, например, чтобы заставить панель навигации прокручиваться вместе со страницей до определённой точки, а затем задерживать в верхней части страницы.
- Интересное и общее использование position: sticky заключается в создании индексных страниц с прокруткой, где разные заголовки липнут к верху страницы, когда они достигают его.
- `.positioned { position: sticky; top: 30px; left: 30px; }`

Конец

ПОСЛЕСЛОВИЕ

Давайте подведем итоги урока!
Чему мы научились? Что мы использовали?
К чему мы пришли?