

Позиционирование

CSS предоставляет возможности по позиционированию элемента, то есть мы можем поместить элемент в определенное место на странице

Основным свойством, которые управляют позиционированием в CSS, является свойство `position`. Это свойство может принимать одно из следующих значений:

`static`: стандартное позиционирование элемента, значение по умолчанию

`absolute`: элемент позиционируется относительно границ элемента-контейнера, если у того свойство `position` не равно `static`

`relative`: элемент позиционируется относительно его позиции по умолчанию. Как правило, основная цель относительного позиционирования заключается не в том, чтобы переместить элемент, а в том, чтобы установить новую точку привязки для абсолютного позиционирования вложенных в него элементов

`fixed`: элемент позиционируется относительно окна браузера, это позволяет создать фиксированные элементы, которые не меняют положения при прокрутке

Не следует одновременно применять к элементу свойство `float` и любой тип позиционирования, кроме `static` (то есть тип по умолчанию).

Абсолютное позиционирование

Область просмотра браузера имеет верхний, нижний, правый и левый края. Для каждого из этих четырех краев есть соответствующее свойство CSS: `left` (отступ от края слева), `right` (отступ от края справа), `top` (отступ от края контейнера сверху) и `bottom` (отступ снизу). Значения этих свойств указываются в пикселях, `em` или процентах. Необязательно задавать значения для всех четырех сторон. Как правило, устанавливают только два значения - отступ от верхнего края `top` и отступ от левого края `left`.

```
<!DOCTYPE html>
```

```
<html>
```

```
  <head>
```

```
    <meta charset="utf-8">
```

```
    <title>Блочная верстка в HTML5</title>
```

```
    <style>
```

```
      .header {
```

```
        position: absolute;
```

```
        left: 100px;
```

```
        top: 50px;
```

```
        width: 430px;
```

```
        height: 100px;
```

```
        background-color: rgba(128, 0, 0, 0.5);
```

```
      }
```

```
</style>
</head>
<body>
  <div class="header"></div>
  <p>HELLO WORLD</p>
</body>
</html>
```

Здесь элемент div с абсолютным позиционированием будет находиться на 100 пикселей слева от границы области просмотра и на 50 снизу.

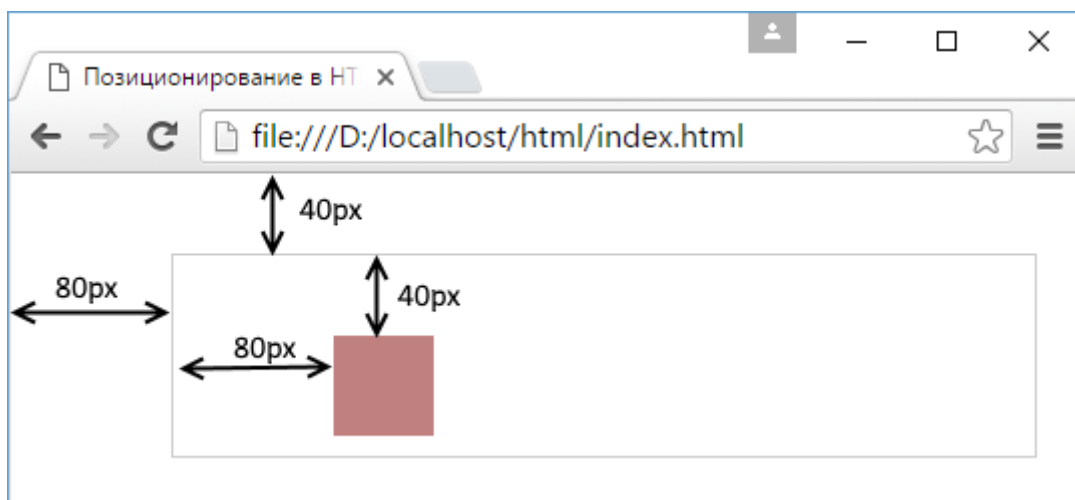


При этом не столь важно, что после этого элемента div идут какие-то другие элементы. Данный блок div в любом случае будет позиционироваться относительно границ области просмотра браузера.

Если элемент с абсолютным позиционированием располагается в другом контейнере, у которого в свою очередь значение свойства position не равно static, то элемент позиционируется относительно границ контейнера:

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <title>Позиционирование в HTML5</title>
    <style>
      .outer {
        position: absolute;
```

```
    left: 80px;
    top: 40px;
    width: 430px;
    height: 100px;
    border: 1px solid #ccc;
}
.inner{
    position: absolute;
    left: 80px;
    top: 40px;
    width: 50px;
    height: 50px;
    background-color: rgba(128, 0, 0, 0.5);
}
</style>
</head>
<body>
    <div class="outer">
        <div class="inner"></div>
    </div>
</body>
</html>
```



Относительное позиционирование

Относительное позиционирование также задается с помощью значения `relative`. Для указания конкретной позиции, на которую сдвигается элемент, применяются те же свойства `top`, `left`, `right`, `bottom`:

```
<!DOCTYPE html>

<html>

  <head>

    <meta charset="utf-8">

    <title>Позиционирование в HTML5</title>

    <style>

      .outer {

        position: relative;

        left: 80px;

        top: 40px;

        width: 300px;

        height: 100px;

        border: 1px solid #ccc;

      }

      .inner{

        position: absolute;

        left: 80px;

        top: 40px;

        width: 50px;

        height: 50px;

        background-color: rgba(128, 0, 0, 0.5);

      }

    </style>

  </head>

  <body>

    <div class="outer"><div class="inner"></div></div>

  </body>

</html>
```

Свойство z-index

По умолчанию при совпадении у двух элементов границ, поверх другого отображается тот элемент, который определен в разметке html последним. Однако свойство z-index позволяет изменить порядок следования элементов при их наложении. В качестве значения свойство принимает число. Элементы с большим значением этого свойства будут отображаться поверх элементов с меньшим значением z-index.

Например:

```
<!DOCTYPE html>

<html>

  <head>

    <meta charset="utf-8">

    <title>Позиционирование в HTML5</title>

    <style>

      body{

        margin:0;

        padding:0;

      }

      .content{

        position: relative;

        top: 15px;

        left: 20px;

        width: 250px;

        height: 180px;

        background-color: #eee;

        border: 1px solid #ccc;

      }

      .redBlock{

        position: absolute;

        top: 20px;

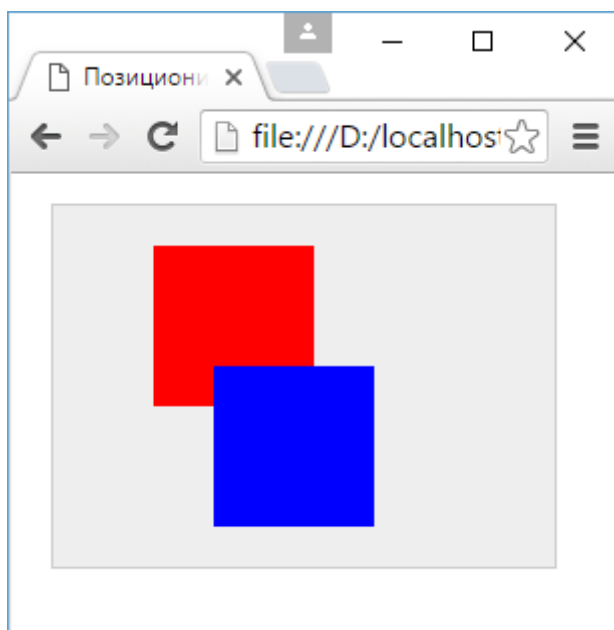
        left:50px;

        width: 80px;

        height: 80px;
```

```
        background-color: red;
    }

    .blueBlock{
        position: absolute;
        top: 80px;
        left: 80px;
        width: 80px;
        height: 80px;
        background-color: blue;
    }
</style>
</head>
<body>
    <div class="content">
        <div class="redBlock"></div>
        <div class="blueBlock"></div>
    </div>
</body>
</html>
```



Теперь добавим к стилю блока redBlock новое правило:

```
.redBlock{  
  z-index: 100;  
  
  position: absolute;  
  top: 20px;  
  left: 50px;  
  width: 80px;  
  height: 80px;  
  background-color: red;  
}
```

Здесь z-index равен 100. Но это необязательно должно быть число 100. Так как у второго блока z-index не определен и фактически равен нулю, то для redBlock мы можем установить у свойства z-index любое значение больше нуля.

И теперь первый блок будет накладываться на второй, а не наоборот, как было в начале:

