

## Тема 11: Позиционирование

В прошлой теме мы уже познакомились с одним из свойств позиционирования – `position`. Теперь изучим ещё несколько необходимых свойств.

*top* - определяет расстояние от верхнего края родительского элемента до верхнего края дочернего элемента. Отсчёт координат зависит от значения свойства `position`. Если оно равно `absolute`, в качестве родителя выступает окно браузера и положение элемента определяется от его верхнего края. В случае значения `relative`, *top* отсчитывается от верхнего края исходного положения элемента. Если для родительского элемента задано `position: relative`, то абсолютное позиционирование дочерних элементов определяет их положение от верхнего края родителя.

*right* - определяет расстояние от правого края родительского элемента, не включая отступ, поле и ширину рамки, до правого края дочернего элемента. Отсчёт координат зависит от значения свойства `position`. Если оно равно `absolute`, в качестве родителя выступает окно браузера и положение элемента определяется от его правого края. В случае значения `relative`, *right* отсчитывается от правого края исходного положения элемента. Если для родительского элемента задано `position: relative`, то абсолютное позиционирование дочерних элементов определяет их положение от правого края родителя.

*bottom* - устанавливает положение нижнего края содержимого элемента без учёта толщины рамок и отступов. Отсчёт координат зависит от свойства `position`, оно обычно принимает значение `relative` (относительное положение) или `absolute` (абсолютное положение).

*left* - определяет расстояние от левого края родительского элемента, не включая отступ, поле и ширину рамки, до левого края дочернего элемента. Отсчёт координат зависит от значения свойства `position`.

Любые позиционированные элементы на веб-странице могут накладываться друг на друга в определенном порядке, имитируя тем самым третье измерение, перпендикулярное экрану. Каждый элемент может находиться как ниже, так и выше других объектов веб-страницы, их размещением по z-оси и управляет *z-index*. Это свойство работает только для элементов, у которых значение `position` задано как `absolute`, `fixed` или `relative`.

### Практическая работа №11.1

Попробуем поработать со свойствами, которые мы изучили.

```

<!DOCTYPE html>
<html>
<head>
  <title>Позиционирование</title>
  <meta charset="utf-8">
  <link rel="stylesheet" type="text/css" href="css/style.css">
</head>
<body>
  <div class="layer1">
    <div class="layer2">
      Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod
      tempor incididunt ut labore et dolore magna aliqua.
    </div>
  </div>
  <div id="z-index">
    
    
    
  </div>
</body>
</html>

```

```

.layer1 {
  position: absolute;
  left: 20px;
  background: #fc3;
  margin: 7px;
}
.layer2 {
  position: relative;
  left: -12px;
  top: 13px;
  border: 1px solid black;
  padding: 5px;
  margin: 7px;
}
#z-index {
  position: absolute;
  top: 100px;
}
.pic {
  position: relative;
}
.shade {
  top: 45px;
  left: 60px;
  z-index: 9;
}
.pic1 {
  left: -400px;
  z-index: 5;
}
.pic2 {
  top: 25px;
  left: -950px;
  z-index: 7;
}
.pic:hover {
  z-index: 11;
}

```

*vertical-align* - выравнивает элемент по вертикали относительно своего родителя, окружающего текста или ячейки таблицы (значение по умолчанию *baseline*).

Может принимать значения:

*baseline* - выравнивает базовую линию блока по базовой линии родителя. Если у блока нет базовой линии, то за неё принимается нижняя граница. *bottom* - выравнивает низ блока по нижней части строки.

*middle* - выравнивает вертикальную среднюю точку блока по базовой линии родительского блока плюс половина высоты буквы «х».

*sub* - опускает базовую линию блока вниз для создания нижнего индекса. Не оказывает влияние на размер текста.

*super* - поднимает базовую линию блока вверх для создания верхнего индекса. Не оказывает влияние на размер текста. *text-bottom* - нижняя граница элемента выравнивается по нижнему краю содержимого родителя.

*text-top* - верхняя граница элемента выравнивается по верхнему краю содержимого родителя. *top* - выравнивает верх блока по верхней части строки.

Свойства внешних и внутренних отступов:

*margin* - устанавливает величину отступа от каждого края элемента. Отступом является пространство от границы текущего элемента до внутренней границы его родительского элемента.

*padding* - устанавливает значение полей вокруг содержимого элемента. Полем называется расстояние от внутреннего края рамки элемента до воображаемого прямоугольника, ограничивающего его содержимое

Свойство *overflow* управляет отображением содержания блочного элемента, если оно целиком не помещается и выходит за область заданных размеров.

## Практическая работа №11.2

```
<div id="wrapper">
  <p>
    Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod
    tempor incididunt ut labore et dolore magna aliqua.
  </p>
  <p>
    Ut enim ad minim veniam,
    quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo
    consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse
    cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non
    proident, sunt in culpa qui officia deserunt mollit anim id est laborum.
  </p>
</div>

#wrapper p {
  width: 300px;
  height: 100px;
  display: inline-block;
  vertical-align: middle;
  border: 2px solid red;
  overflow: auto;
}
#wrapper > p:first-child {
  margin: 15px 5px 10px 20px;
  padding: 15px;
  text-align: center;
}
#wrapper > p:last-child {
  margin-top: 20px;
  padding: 5px 10px 5px 20px;
  text-align: justify;
}
```

Теперь попробуем применить эти свойства на примере.

### Проверка знаний

Теперь давайте изменим наш «Мини сайт» и добавим, уберем или изменим некоторые свойства.

```

/* Обнуляем все отступы*/
* {
    margin: 0;
    padding: 0;
}
/* Назначаем шрифты, ширину, отображение на середине и фон страницы */
body {
    font-family: Helvetica, Tahoma, Arial;
    width: 80em;
    margin: 0 auto;
    background: url(img/main_bg.png) no-repeat grey fixed;
    background-size: cover;
}
header {
    background-color: rgba(0, 0, 255, .3);
}
/* Логотип */
header > img {
    display: inline-block;
    width: 10em;
    margin-left: 3em;
    vertical-align: middle;
}
/* Название организации */
header > h1 {
    display: inline-block;
    font-size: 7em;
    vertical-align: middle;
    margin-left: 1em;
}
/* Меню */
header > nav {
    width: 80em;
    border-top: 1px solid black;
}
/* Пункты меню */
header > nav > a {
    text-decoration: none;
    font-size: 3em;
    color: black;
    margin-left: 3em;
}
/* Посещенные ссылки */
header > nav > a:visited {
    color: black;
}
/* Контентная область */
main {
    background-color: rgba(0, 255, 0, .3);
}
main > section:first-child > article {
    position: relative;
}
/* Изображение слайдера */
main > section:first-child > article > img {
    width: 79.7em;
    align-self: center;
}

```

### **Итоговая практическая работа №3**

Вариант 1

Создайте мини сайт по теме: «Бижутерия».

Вариант 2

Создайте мини сайт по теме: «Точилки для карандашей».

Вариант 3

Создайте мини сайт по теме: «Литые диски».

Вариант 4

Создайте мини сайт по теме: «Масонский заговор».

Вариант 5

Создайте мини сайт по теме: «Лучшие четвероногие друзья людей».

Вариант 6

Создайте мини сайт по теме: «Круасаны всей Франции».

Вариант 7

Создайте мини сайт по теме: «Лучшие велосипедисты России».

Вариант 8

Создайте мини сайт по теме: «Ёжики и дикобразы».

Вариант 9

Создайте мини сайт по теме: «Производство молочной продукции».

Вариант 10

Создайте мини сайт по теме: «Малоизвестные звезды».

Вариант 11

Создайте мини сайт по теме: «Вулканы мира».

Вариант 12

Создайте мини сайт по теме: «Тропический лес».

Вариант 13

Создайте мини сайт по теме: «Отдых на море».

Вариант 14

Создайте мини сайт по теме: «Проекторы».

Вариант 15

Создайте мини сайт по теме: «Лунные кратеры».