



# Параллакс

**LoftSchool**  
от мыслителя к создателю

## Оглавление:

О Технике параллакс	3
Создание параллакс-эффекта самостоятельно	4
Создание параллакс-эффекта с помощью плагина parallax.js	8

## О Технике параллакс

**Параллакс** - это техника анимации.

Параллакс основан на взаимодействии фоновых изображений. Из - за того что происходит разная скорость движения background и контента - создается некая иллюзия 3d.

Чтобы объяснить более наглядно, посмотрим пример [предпросмотра темы Parallax One](#).

Как вы видите, фон на странице начинается с изображения города, занимающего по ширине всю страницу и большую область по высоте. Но когда мы двигаем мышкой над этим изображением, фон двигается, что смотрится очень красиво.

Кстати, параллакс применяется не только в вебе, его можно встретить и в играх, таких как Sonic (1991), Mortal Combat и т.д.

Давайте приступим к внедрению параллакса на нашу страничку, поговорим про методы и базовую концепцию.

Для создания параллакса существует огромное количество способов, рассмотрим некоторые из них.

# Создание параллакс-эффекта самостоятельно

Давайте напишем простейший параллакс-эффект.  
Нам понадобится:

- 3 любых картинки в качестве примера,
- библиотека jQuery

Создадим 3 секции, фоном в которые поместим наши картинки:

## html

```
<section class="first"></section>
<section class="second"></section>
<section class="third"></section>
```

## scss

*// Уберем отступы у body*

```
body {
  margin: 0;
  padding: 0;
}
```

*// Сделаем секциям высоту на весь экран*

```
section {
  display: flex;
  height: 100vh;
  background-position: 50% 0;
  background-attachment: fixed; // Для того, чтобы фон не
  двигался
  background-size: cover;
}
```

*// В каждую секцию положим фоновое изображение, которое будет*  
*скроллится медленнее, чем сам контент*

```
.first {
  background-image: url(img/1.jpg);
}
```

```

.second {
  background-image: url(img/2.jpg);
}

.third {
  background-image: url(img/3.jpg);
}

```

Теперь добавим атрибут `data-speed="3"`. Он будет браться далее - в JS и, в зависимости от его значения, будет меняться скорость параллакса.

```

<section class="first" data-speed="3">
  <div class="title">Parallax is cool</div>
</section>

```

Отцентрируем секцию, добавим цвет, размер:

```

.title {
  margin: auto;
  font: bold 50px Arial;
  color: white;
  text-transform: uppercase;
}

```

Теперь создадим файл **main.js**, подключим его в разметке после подключения jquery.

Логика работы будет такова:

- Получаем значение - насколько пользователь прокрутил страничку.
- Делим на значение скорости, которое мы указали в `data-speed`.
- Эти значения подставляем в `background-position` по оси Y

Начнем с того, что найдем нашу первую секцию с текстом и присвоим его переменной:

```
var section = $('.first');
```

Далее нам необходимо повесить на объект window событие `scroll`:

```
$(window).scroll(function() {  
  
    // Когда мы будем скролить колесиком мышки, весь код, который  
    // размещен внутри этой функции, будет выполняться.  
  
});
```

Внутри функции будем сохранять в переменную высоту, на которую мы проскроллили страничку (позицию скрола).

```
var scrollTop = -($(window).scrollTop());
```

Нам нужно отрицательно число, чтобы картинка сдвигалась вверх, а не вниз.

Возьмем значения `data-speed` из разметки

```
var speed = section.data('speed');
```

Так же нам понадобится переменная, которая будет содержать число, необходимое для сдвига картинки при скролле:

```
var coords = "50%" + scrollTop / speed + "px";
```

После этого, давайте применим параллакс к нашему тексту. Для этого воспользуемся свойством `background-position` и его значению будем присваивать нашу переменную `coords`.

```
section.css('background-position', coords);
```

Вот и все, теперь мы можем менять значение в `data-speed` атрибуте, наблюдая, как начинает меняться скорость параллакса.

Итоговый файл *main.js* будет таким:

```
var section = $('.first');
$(window).scroll(function () {
    var scrollTop = -($(window).scrollTop()),
        speed = section.data('speed'),
        coords = "50%" + scrollTop / speed + "px";
    section.css('background-position', coords);
});
```

## Создание параллакс-эффекта с помощью плагина `parallax.js`



### [parallax.js](#)

Скачиваем архив, копируем файл ***parallax.min.js*** из папки `deploy` подключаем его в ***index.html*** (после библиотеки `jquery`).

Создадим разметку для нашего параллакса:

```
<ul id="scene">
  <li class="layer" data-depth="0.00"></li>
  <li class="layer" data-depth="0.20"></li>
  <li class="layer" data-depth="0.40"></li>
  <li class="layer" data-depth="0.60"></li>
  <li class="layer" data-depth="0.80"></li>
  <li class="layer" data-depth="1.00"></li>
</ul>
```

Атрибут `data-depth="0.20"` будет определять, какой слой будет двигаться медленнее, какой быстрее, а какой - вообще не будет двигаться.



Нам нужно инициализировать наш параллакс. Для этого создаем файл *main.js*, подключаем его после наших библиотек.

В файле *main.js* и пишем следующий код:

```
var scene = document.getElementById('scene');  
var parallax = new Parallax(scene);
```

Готово!

У плагина есть [множество настроек](#)

Для того, чтобы воспользоваться какими-то настройками, достаточно передать их в конструктор при вызове. Например:

```
var parallax = new Parallax(scene, {  
  calibrateX: false,  
  calibrateY: true,  
  invertX: false,  
  invertY: true,  
  limitX: false,  
  limitY: 10,  
  scalarX: 2,  
  scalarY: 8,  
  frictionX: 0.2,  
  frictionY: 0.8,  
  originX: 0.0,  
  originY: 1.0  
});
```

Если в разработке мы используем библиотеку jQuery, то можем вызвать наш плагин более простым способом:

```
$('#scene').parallax();
```

А если мы хотим вызвать его с настройками, достаточно передать объект настроек вот так:

```
$('#scene').parallax({
  calibrateX: false,
  calibrateY: true,
  invertX: false,
  invertY: true,
  limitX: false,
  limitY: 10,
  scalarX: 2,
  scalarY: 8,
  frictionX: 0.2,
  frictionY: 0.8,
  originX: 0.0,
  originY: 1.0
});
```

Так же у данного плагина есть методы, с которыми вы легко сможете разобраться самостоятельно :

```
var $scene = $('#scene').parallax();
$scene.parallax('enable');
$scene.parallax('disable');
$scene.parallax('updateLayers');
$scene.parallax('calibrate', false, true);
$scene.parallax('invert', false, true);
$scene.parallax('limit', false, 10);
$scene.parallax('scalar', 2, 8);
$scene.parallax('friction', 0.2, 0.8);
$scene.parallax('origin', 0.0, 1.0);
```