

**МИНИСТЕРСТВО ЦИФРОВОГО РАЗВИТИЯ, СВЯЗИ И
МАССОВЫХ КОММУНИКАЦИЙ РОССИЙСКОЙ ФЕДЕРАЦИИ
Ордена Трудового Красного Знамени федеральное государственное
бюджетное образовательное учреждение высшего образования
«Московский технический университет связи и информатики»**

Лабораторная работа №9

«Дальнейшее изучение библиотеки jQuery, добавление эффекта
параллакса для главной страницы блога»

по дисциплине:

«Web-программирование»

Выполнил:

Студент группы БФИ2102

Шаинян С.А.

Проверил:

Фатхулин Т.Д.

Москва, 2023

1. Цель работы

Цель данной лабораторной работы — научиться разрабатывать эффект параллакса с помощью библиотеки jQuery.

2. Задание

В данной лабораторной работе необходимо выполнить следующие задания:

- 1) Создать эффект параллакса для изображений.
 - 2) Настроить также эффект параллакса для картинки-логотипа блога.
- Параметры для скорости перемещения задать самостоятельно.

3. Ход работы

Скопируем предыдущий проект (лабораторная работа №8).

Для того чтобы разместить картинки одна под другой с эффектом смещения и пересечения добавим следующие стили.

Листинг 1 – Стили для изображений

```
.icons-for-parallax {  
  margin-top: 50px;  
  height: 335px;  
  position: relative;  
}  
  
.icons-for-parallax img {  
  position: absolute;  
  top: 0;  
}  
  
.icon-for-parallax-first {  
  width: 200px;  
  margin-top: 120px;  
  z-index: 3;  
}  
  
.icon-for-parallax-second {  
  width: 175px;  
  margin-top: 60px;  
  left: 150px;  
  z-index: 2;  
}
```

```
.icon-for-parallax-third {  
  width: 150px;  
  left: 260px;  
  z-index: 1;  
}
```

После внесения всех изменений в верстку, добавим в папку `articles/static/js/` новый файл `parallax.js`. Именно в этом файле будет содержаться исходный код скрипта, который изменяет координаты изображений при прокрутке страницы.

Для выполнения всех шагов начнем со стандартной для каждого файла, где подключен jQuery, операции: ожидания события `ready` для документа:

Листинг 2 – Событие `ready`

```
$(document).ready(function () {  
  // Код будет здесь  
});
```

Теперь можно быть уверенным, что код запустится только после окончательно загрузки DOM. Для шагов инициализации, описанных выше, добавим в свой файл следующий исходный код:

Листинг 3 – Инициализация переменных

```
var yPosition;  
var scrolled = 0;  
var $parallaxElements = $(".icons-for-parallax img");
```

Чтобы можно было при каждой прокрутке выполнять какие-либо операции, на событие `scroll` объекта `window` добавим функцию-обработчик:

Листинг 4 – Событие прокрутки страницы

```
$(window).scroll(function () {  
  // код для изменения при прокрутке страницы  
});
```

Внутри этой функции сначала следует узнать количество прокрученных пикселей, для чего используем следующую функцию так, как показано ниже.

Листинг 5 – Получение количества прокрученных пикселей

```
scrolled = $(window).scrollTop();
```

Естественно, эту операцию нужно выполнять внутри функции-обработчика события scroll. Затем там же, в этой функции запустим цикл по всем элементам внутри переменной \$parallaxElements.

Листинг 5 – Цикл для прохода по изображениям

```
for (var i = 0; i < $parallaxElements.length; i++) {  
    // Изменение каждого изображения  
}
```

Внутри цикла сначала посчитаем на сколько пикселей вам нужно визуально опустить текущий элемент.

Листинг 6 – Подсчет отклонения позиции по вертикали для каждого i элемента

```
yPosition = scrolled * 0.15 * (i + 1);
```

Осталось только установить уже подсчитанные координаты для текущего элемента:

Листинг 7 – Применения отклонения

```
$parallaxElements.eq(i).css({ top: yPosition });
```

Метод eq(), позволяет из всего списка элементов, который хранится в jQuery-переменной, выбрать именно тот элемент, позиция которого совпадает с переданным методу числом. В данном случае в цикле последовательно из списка берём 0-ой элемент, затем 1-ый, и, наконец, 2-ой. Для каждого из них

устанавливаем координату, которая напрямую зависит от номера позиции, ведь в операции подсчета пикселей также фигурирует переменная *i*.

В конце изменим размер *header* для визуального вмещения в него картинок при прокрутке.

В конечном итоге код нашего скрипта имеет следующий вид:

Листинг 8 – *parallax.js*

```
$(document).ready(function () {  
    var yPosition;  
    var scrolled = 0;  
    var $parallaxElements = $(".icons-for-parallax img");  
    var $header = $(".header");  
  
    $(window).scroll(function () {  
        scrolled = $(window).scrollTop();  
        for (var i = 0; i < $parallaxElements.length; i++) {  
            yPosition = scrolled * 0.15 * (i + 1);  
            $parallaxElements.eq(i).css({ top: yPosition });  
        }  
    });  
  
    $header.height(initHeight + scrolled * 0.32);  
});
```

Результат работы параллакса представлен на рисунках 1 и 2.

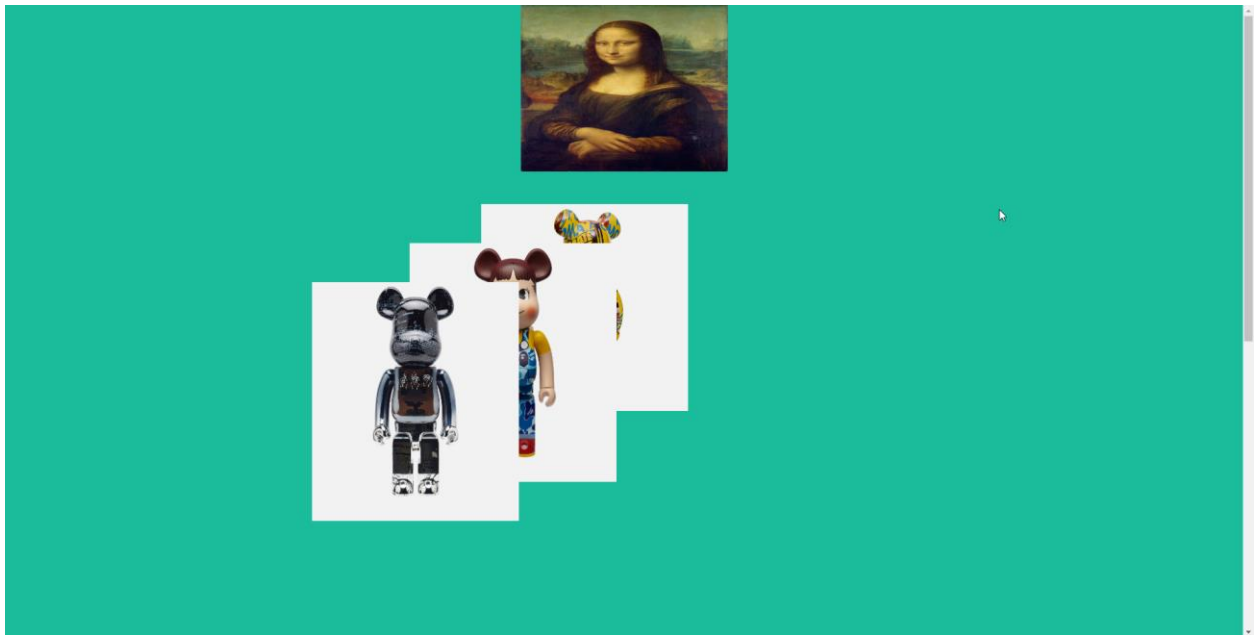


Рисунок 1 – Параллакс изображений. Начальное состояние

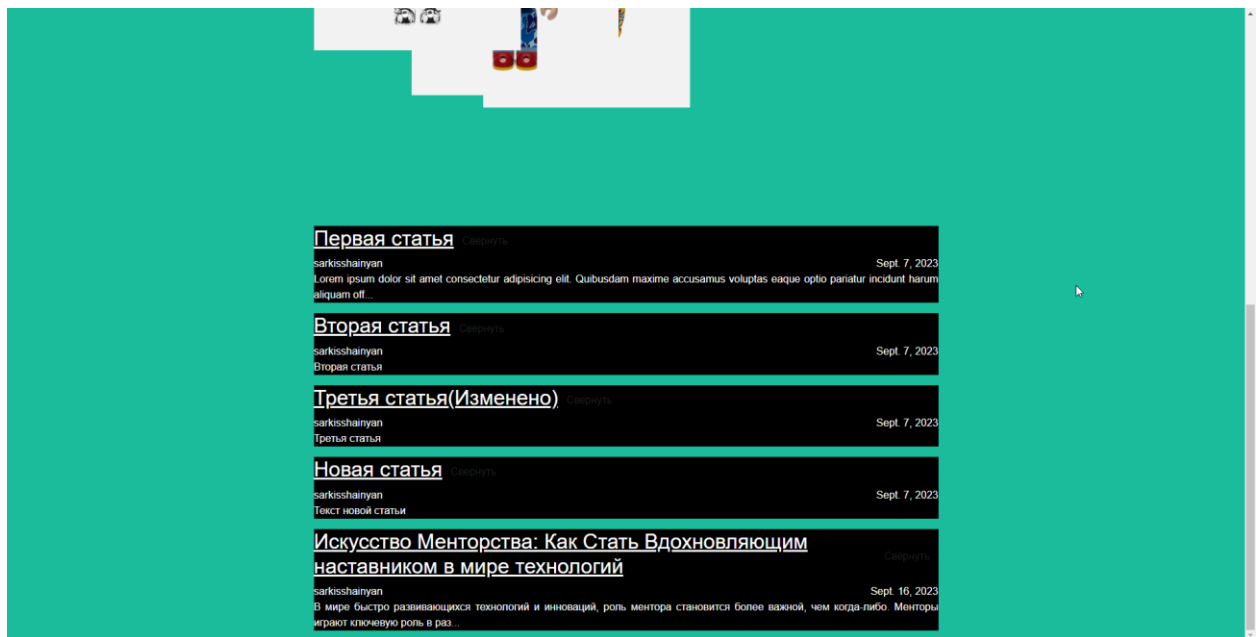


Рисунок 2 – Параллакс изображений. Произведена небольшая прокрутка страницы

Аналогично разработаем эффект параллакса для логотипа. Для этого добавим блок «logo» и еще одно изображение логотипа.

Листинг 9 – html-блок logo

```
<div class="logo">
  
  
</div>
```

Добавим стили для двух изображений логотипа.

Листинг 10 – Стили для изображений логотипа

```
.logo {
  position: relative;
  height: 160px;
}

.logo img {
  position: absolute;
  left: calc(50% - 160px);
}

.logo img:nth-child(2) {
  position: absolute;
  left: calc(50% - 350px);
  margin-top: 50px;
}
```

Добавим в parallax.js изменение положения изображений логотипа при прокрутке страницы.

Листинг 11 – parallax.js

```
$(document).ready(function () {  
    var yPosition;  
    var scrolled = 0;  
    var $parallaxElements = $(".icons-for-parallax img");  
    var $logoElements = $(".logo img");  
    var $header = $(".header");  
    var initHeight = $header.height();  
  
    $(window).scroll(function () {  
        scrolled = $(window).scrollTop();  
        for (var i = 0; i < $parallaxElements.length; i++) {  
            yPosition = scrolled * 0.15 * (i + 1);  
            $parallaxElements.eq(i).css({ top: yPosition });  
        }  
  
        for (var i = 0; i < $logoElements.length; i++) {  
            yPosition = scrolled * 0.15 * ($logoElements.length - (i * 5 + 1));  
            $logoElements.eq(i).css({ top: yPosition });  
        }  
  
        $(".logo").css({ marginTop: Math.min(scrolled, 30) });  
  
        $header.height(initHeight + scrolled * 0.32);  
    });  
});
```

Конечный результат работы представлен на рисунках 3 и 4.

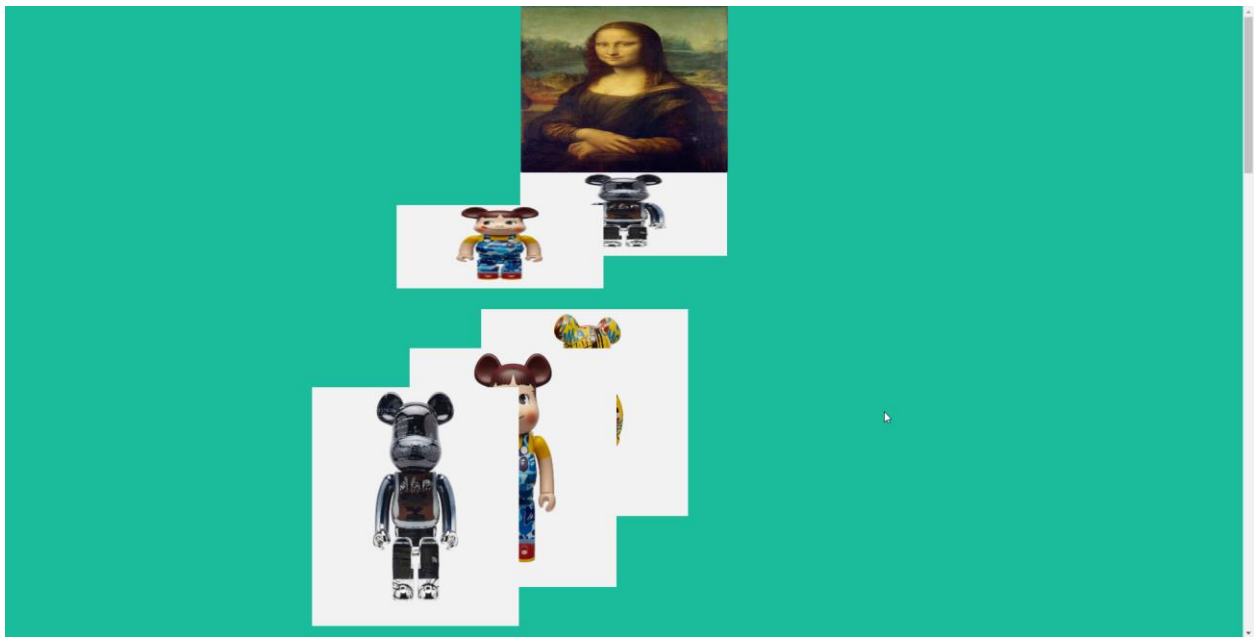


Рисунок 1 – Параллакс изображений логотипа. Начальное состояние

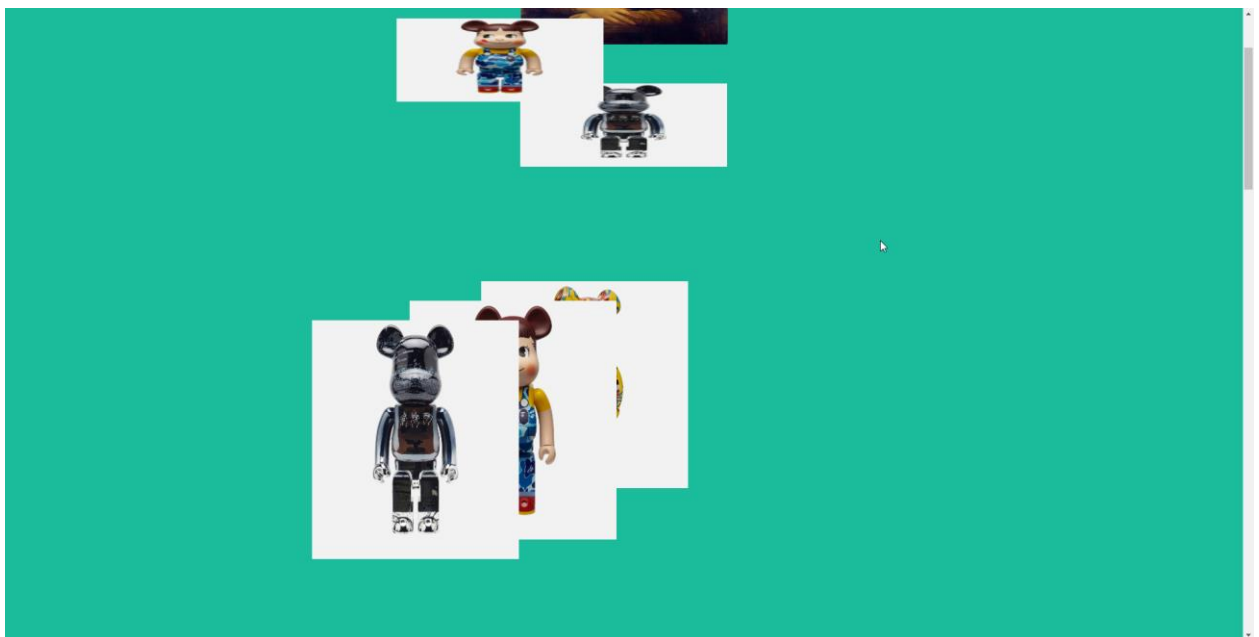


Рисунок 4 – Параллакс изображений логотипа. Произведена небольшая прокрутка страницы

Вывод

В данной лабораторной работе мы научились научиться разрабатывать эффект параллакса с помощью библиотеки jQuery.