



МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное образовательное учреждение
высшего образования

«МИРЭА – Российский технологический университет»

РТУ МИРЭА

Институт информационных технологий (ИТ)

Кафедра инструментального и прикладного программного обеспечения (ИиППО)

ОТЧЕТ ПО ПРАКТИЧЕСКИМ РАБОТАМ

по дисциплине

«Разработка клиентских частей интернет-ресурсов»

Выполнил студент группы ИКБО-20-19

Николаев-Аксенов И.С.

Принял
Ассистент

Меркулов Е.С.

Практические работы
выполнены

«__»_____2020 г.

(подпись студента)

«Зачтено»

«__»_____2020 г.

(подпись руководителя)

Москва 2020

Оглавление

Практическая работа №8 «JavaScript-анимация»	3
Задание 1: «Использование setInterval».....	3
Задание 2: «Функция «draw»	5
Задание 3: «Анимированный прыгающий мяч»	8
Вывод	10

Практическая работа №8 «JavaScript-анимация»

Цель работы: ознакомиться со структурой языка HTML, научиться создавать HTML-документы, оформлять комментарии в коде, работать с основными тегами языка, добавлять изображения и ссылки на страницы, подключать внешние CSS-файлы и придавать простейшие стили HTML-документам.

Задание 1: «Использование setInterval»

Выбрать изображение любого вида транспорта и реализовать анимацию (при нажатии на данное изображение) через последовательность кадров, каждый из которых немного меняет HTML/CSS-свойства. Например, изменение `style.left` от 0px до 100px – двигает элемент. И если это будет выполняться с помощью `setInterval`, изменяя на 2px с небольшими интервалами времени, например, 50 раз в секунду, тогда изменения будут выглядеть плавными. Принцип такой же, как в кино: 24 кадров в секунду достаточно, чтобы создать эффект плавности.

Для реализации необходимо:

1. Запомнить время начала анимации;
2. Добавить функцию для расчета времени, которое прошло с начала анимации;
3. Закончить анимацию через 2 секунды;
4. В то время как `timePassed` идёт от 0 до 2000 `left` изменяет значение от 0px до 400px.

Описание выполнения задания

1. Создание HTML-документа в редакторе Visual Studio Code (Листинг 1.1).

Листинг 1.1 – Создание HTML-документа

```
1. <!DOCTYPE html>
2. <html lang="en">
3.   <head>
4.     <meta charset="UTF-8" />
5.     <meta name="viewport" content="width=device-width, initial-scale=1.0" />
6.     <title>Задача 1</title>
7.   </head>
8.   <body>
9.     
16.    <script>
17.      function move(e) {
18.        var left = 0;
19.        var size = window.screen.width - e.width;
20.        var a = setInterval(function () {
21.          if (left++ > size) {
22.            clearInterval(timerId);
23.          }
24.          e.style.left = left + "px";
25.        }, 1);
26.      }
27.    </script>
28.  </body>
29. </html>
```

Результат выполнения кода на странице (Рисунок 1.1, Рисунок 1.2):



Рисунок 1.1 – Результат выполнения кода



Рисунок 1.2 – Результат выполнения кода

Задание 2: «Функция «draw»»

Создать анимацию набирающегося текста в элементе формы textarea при нажатии на кнопку. Добавить в <head> следующую функцию для анимации (дописав функции для textarea и расчёта времени bounce):

```
function animate(options) {

    var start = performance.now();

    requestAnimationFrame(function animate(time) {
        // timeFraction от 0 до 1
        var timeFraction = (time - start) / options.duration;
        if (timeFraction > 1) timeFraction = 1;

        // текущее состояние анимации
        var progress = options.timing(timeFraction)

        options.draw(progress);

        if (timeFraction < 1) {
            requestAnimationFrame(animate);
        }
    });
}
```

Описание выполнения задания

2. Создание HTML-документа в редакторе Visual Studio Code (Листинг 2.1).

Листинг 2.1 – Создание HTML-документа

```
1. <!DOCTYPE html>
2. <html lang="en">
3.   <head>
4.     <meta charset="UTF-8" />
5.     <meta name="viewport" content="width=device-width, initial-scale=1.0" />
6.
7.     <title>Задача 2</title>
8.   </head>
9.   <body>
10.    <script>
11.      function animate({ timing, draw, duration }) {
12.        let start = performance.now();
13.
14.        requestAnimationFrame(function animate(time) {
15.          let timeFraction = (time - start) / duration;
16.          if (timeFraction > 1) timeFraction = 1;
17.
```

```

18.     let progress = timing(timeFraction);
19.
20.     draw(progress);
21.
22.     if (timeFraction < 1) {
23.         requestAnimationFrame(animate);
24.     }
25. });
26. }
27.
28. function animateText(textArea) {
29.     let text = textArea.value;
30.     let to = text.length,
31.         from = 0;
32.
33.     animate({
34.         duration: 5000,
35.         timing: bounce,
36.         draw: function (progress) {
37.             let result = (to - from) * progress + from;
38.             texttextArea.value = text.substr(0, Math.ceil(result));
39.         },
40.     });
41. }
42.
43. function bounce(timeFraction) {
44.     for (let a = 0, b = 1, result; 1; a += b, b /= 2) {
45.         if (timeFraction >= (7 - 4 * a) / 11) {
46.             return (
47.                 -Math.pow((11 - 6 * a - 11 * timeFraction) / 4, 2) +
48.                 Math.pow(b, 2)
49.             );
50.         }
51.     }
52. }
53. </script>
54.
55. <textarea id="ta" rows="5" cols="60">Lorem ipsum dolor sit amet consectetur, adipisici
ng elit. Molestiae, reiciendis. Pariatur facere aliquam mollitia laboriosam, ex excepturi
ad expedita modi, libero officia nisi temporibus tenetur quaerat, dicta perspiciatis solut
a qui. </textarea>
56. <button style="display: block; margin-top: 1%" onclick="animateText(ta)">
57.     Animate textarea
58. </button>
59. </body>
60. </html>

```

Результат выполнения кода на странице (Рисунок 2.1, Рисунок 2.2):

```

Lorem ipsum dolor sit amet consectetur, adipisicing elit.
Molestiae, reiciendis. Pariatur facere aliquam mollitia
laboriosam, ex excepturi ad expedita modi, libero officia nisi
temporibus tenetur quaerat, dicta perspiciatis soluta qui.

```

Animate textarea

Рисунок 2.1 – Результат выполнения кода

```

Lorem ipsum dolor sit amet consectetur

```

Animate textarea

Рисунок 2.2 – Результат выполнения кода

Задание 3: «Анимированный прыгающий мяч»

Создать анимацию с прыгающим мячом. При нажатии на мяч он должен падать вниз до созданной горизонтальной линии и несколько раз подскакивать от этой поверхности. Чтобы получить эффект «скачущего» мяча, можно использовать функцию расчёта времени bounce в режиме easeOut.

Описание выполнения задания

3. Создание HTML-документа в редакторе Visual Studio Code (Листинг 3.1).

Листинг 3.1 – Создание HTML-документа

```
1. <!DOCTYPE html>
2. <html lang="en">
3.   <head>
4.     <meta charset="UTF-8" />
5.     <meta name="viewport" content="width=device-width, initial-scale=1.0" />
6.     <title>Задача 3</title>
7.   </head>
8.   <body>
9.     <div
10.      id="field"
11.      style="height: 500px; border-bottom: 3px black groove; position: relative"
12.    >
13.      
19.    </div>
20.
21.    <script>
22.      function animate(options) {
23.        var start = performance.now();
24.
25.        requestAnimationFrame(function animate(time) {
26.          var timeFraction = (time - start) / options.duration;
27.          if (timeFraction > 1) timeFraction = 1;
28.
29.          var progress = options.timing(timeFraction);
30.
31.          options.draw(progress);
32.
33.          if (timeFraction < 1) {
34.            requestAnimationFrame(animate);
35.          }
36.        });
37.      }
38.
39.      function makeEaseOut(timing) {
40.        return function (timeFraction) {
41.          return 1 - timing(1 - timeFraction);
42.        };
43.      }
44.
45.      function bounce(timeFraction) {
```



```

46.     for (let a = 0, b = 1, result; 1; a += b, b /= 2) {
47.         if (timeFraction >= (7 - 4 * a) / 11) {
48.             return (
49.                 -Math.pow((11 - 6 * a - 11 * timeFraction) / 4, 2) +
50.                 Math.pow(b, 2)
51.             );
52.         }
53.     }
54. }
55.
56. ball.onclick = function () {
57.     let to = field.clientHeight - ball.clientHeight;
58.
59.     animate({
60.         duration: 2000,
61.         timing: makeEaseOut(bounce),
62.         draw(progress) {
63.             ball.style.top = to * progress + "px";
64.         },
65.     });
66. };
67. </script>
68. </body>
69. </html>

```

Результат выполнения кода на странице (Рисунок 3.1, Рисунок 3.2):



Рисунок 3.1 – Результат выполнения кода



Рисунок 3.2 – Результат выполнения кода

Вывод

В ходе выполнения практической работы №8, включающую в себя задания с №1 по №3 были освоены навыки программирования на языке JavaScript, а также HTML и CSS. В совокупности данные навыки позволяют написать сайт практически любой сложности.