|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  |  | | --- | --- | --- | |  |  |  | | 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-01-19 | | Иноземцев Н.С. | |
|  | |  | |
| Принял  *Ассистент* | | Коваленко М.А. | |
| Практические работы выполнены | «\_\_\_»\_\_\_\_\_\_\_2020 г. | | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  (подпись студента) | |
| «Зачтено» | «\_\_\_»\_\_\_\_\_\_\_2020 г. | | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  (подпись руководителя) | |
|  |  | |  | |

Москва 2020

**Оглавление**

[Практическая работа №8: «JavaScript-анимация» 3](#_Toc58665893)

[**Задача 1.** 3](#_Toc58665894)

[**Задача 2.** 4](#_Toc58665895)

[**Задача 3.** 6](#_Toc58665896)

# Практическая работа №8: «JavaScript-анимация»

**Цель работы:** Изучение работы с простыми анимациями.

**Задача 1.**

Использование setInterval: Выбрать изображение любого вида транспорта и реализоватьанимацию (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:**

Листинг 1 – создать простую анимацию движения изображения вправо.

<img id="car" src="https://encrypted-tbn0.gstatic.com/images?q=tbn:ANd9GcSPfhPPEuFNJWf5zXdsjkLEIOtlPLISflSVcg&usqp=CAU" style="position: relative; transform: rotateY(180deg);"; width="300px" alt="Car">

<script>

car.onclick = () => {

let start = Date.now()

let timer = setInterval(() => {

let timePassed = Date.now() - start;

if (timePassed >= 2000) {

clearInterval(timer)

return

}

car.style.left = timePassed / 5 + 'px'

}, 20)

}

</script>

Результат выполнения кода на странице (см. Рисунок 1.1-1.2):



Рисунок 1.1

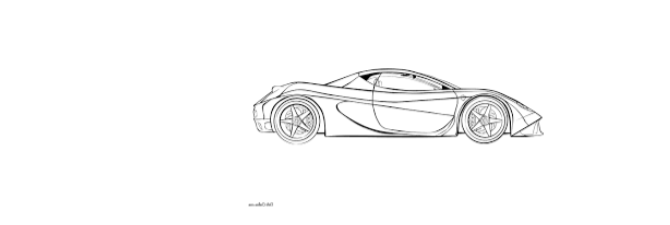


Рисунок 1.2

**Задача 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:**

Листинг 2 – создать анимацию набирающегося текста.

<textarea id="text" rows="10">Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. 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.

</textarea>

<button onclick="startAnim(text)">Start animation</button>

<script>

function animate(options) {

let start = performance.now()

requestAnimationFrame(function animate(time) {

let timeFraction = (time - start) / options.duration

if (timeFraction > 1) timeFraction = 1

let progress = options.timing(timeFraction)

options.draw(progress)

if (timeFraction < 1) {

requestAnimationFrame(animate)

}

})

}

function startAnim(textArea) {

let text = textArea.value

let to = text.length,

from = 0

animate({

duration: 20000,

timing: bounce,

draw: function (progress) {

let result = (to - from) \* progress + from

textArea.value = text.substr(0, Math.ceil(result))

}

})

}

function bounce(timeFraction) {

for (let a = 0, b = 1; true; a += b, b /= 2)

if (timeFraction >= (7 - 4 \* a) / 11)

return -Math.pow((11 - 6 \* a - 11 \* timeFraction) / 4, 2) + Math.pow(b, 2)

}

</script>

Результат выполнения кода на странице (см. Рисунок 2):



Рисунок 2

**Задача 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:**

Листинг 3 – создать анимацию падающего меча, который затем отскакивает от платформы.

<style>

#field {

height: 200px;

border-bottom: 3px black groove;

position: relative;

}

#ball {

position: absolute;

cursor: pointer;

}

</style>

<div id="field">

<img src="https://encrypted-tbn0.gstatic.com/images?q=tbn:ANd9GcShhVooV5EaijQzPBYzjDPTRQKyD5711DGtbkKMT05Zryl9ErJYnEGyYbZInG9FZ5Ig4W5VcHXA&usqp=CAc" width="40" height="40" id="ball" alt="Ball">

</div>

<script>

function animate(options) {

let start = performance.now()

requestAnimationFrame(function animate(time) {

let timeFraction = (time - start) / options.duration

if (timeFraction > 1) timeFraction = 1

let progress = options.timing(timeFraction)

options.draw(progress)

if (timeFraction < 1) {

requestAnimationFrame(animate)

}

})

}

function makeEaseOut(timing) {

return timeFraction => {

return 1 - timing(1 - timeFraction)

}

}

function bounce(timeFraction) {

for (let a = 0, b = 1; 1; a += b, b /= 2)

if (timeFraction >= (7 - 4 \* a) / 11)

return -Math.pow((11 - 6 \* a - 11 \* timeFraction) / 4, 2) + Math.pow(b, 2)

}

ball.onclick = function () {

let to = field.clientHeight - ball.clientHeight

animate({

duration: 4000,

timing: makeEaseOut(bounce),

draw(progress) {

ball.style.top = to \* progress + 'px'

}

})

}

</script>

Результат выполнения кода на странице (см. Рисунок 3):

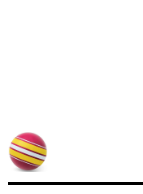


Рисунок 3