

**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 №7

**«Изучение основ JavaScript, создание простейших функций и использование
базовых операторов. Работа с элементами DOM с помощью JavaScript»**

по дисциплине:

«Web-программирование»

Выполнил:

Студент группы БФИ2102

Шаинян С.А.

Проверил:

Фатхулин Т.Д.

Москва, 2023

Цель работы: изучение основ JavaScript, создание простейших функций и использование базовых операторов, а также работа с элементами DOM с помощью JavaScript.

Задание на лабораторную работу

- 1) Написать функцию, которая фильтрует студентов по группе. Функция должна возвращать только тех студентов, что учатся в указанной группе.
- 2) Сделать другую реализацию функции-обработчика, которая уместилась бы в гораздо меньшее количество строк кода. Предложенный выше вариант довольно громоздок, однако он хорошо иллюстрирует некоторые возможности манипулирования DOM-ом с помощью JavaScript. Но весь описанный выше функционал можно реализовать с помощью CSS-классов и их изменения через функцию-обработчик. Сделайте так, чтобы класс `folded` добавлялся не кнопке, по которой кликнули, а родителю всего поста (это тот самый элемент, у которого в HTML-разметке установлен класс `“one-post”`). В таком случае свойство `display` можно изменять более элегантным путем, через CSS-стили: `.one-post.folded .article-author { /* данный стиль применится только для элементов класса .article-author, у которых родитель с классом one-post имеет также класс folded */ display: none; }`
- 3) По аналогии через CSS-стили установите исчезновение остальных элементов поста. В функции-обработчике останется только изменять текст кнопки и менять класс для элементов `one-post`.

Ход лабораторной работы.

Скопируем предыдущий проект. Добавим в проект файл `helloworld.js` в директории `articles/static/js`.

Чтобы подключить JavaScript-файл к веб-странице, достаточно вставить следующий тег в HTML-шаблон:

```
<script src="{% static 'articles/js/helloworld.js' %}"></script>
```

Подключим файл к archive.html. На рисунке 1 показан результат.

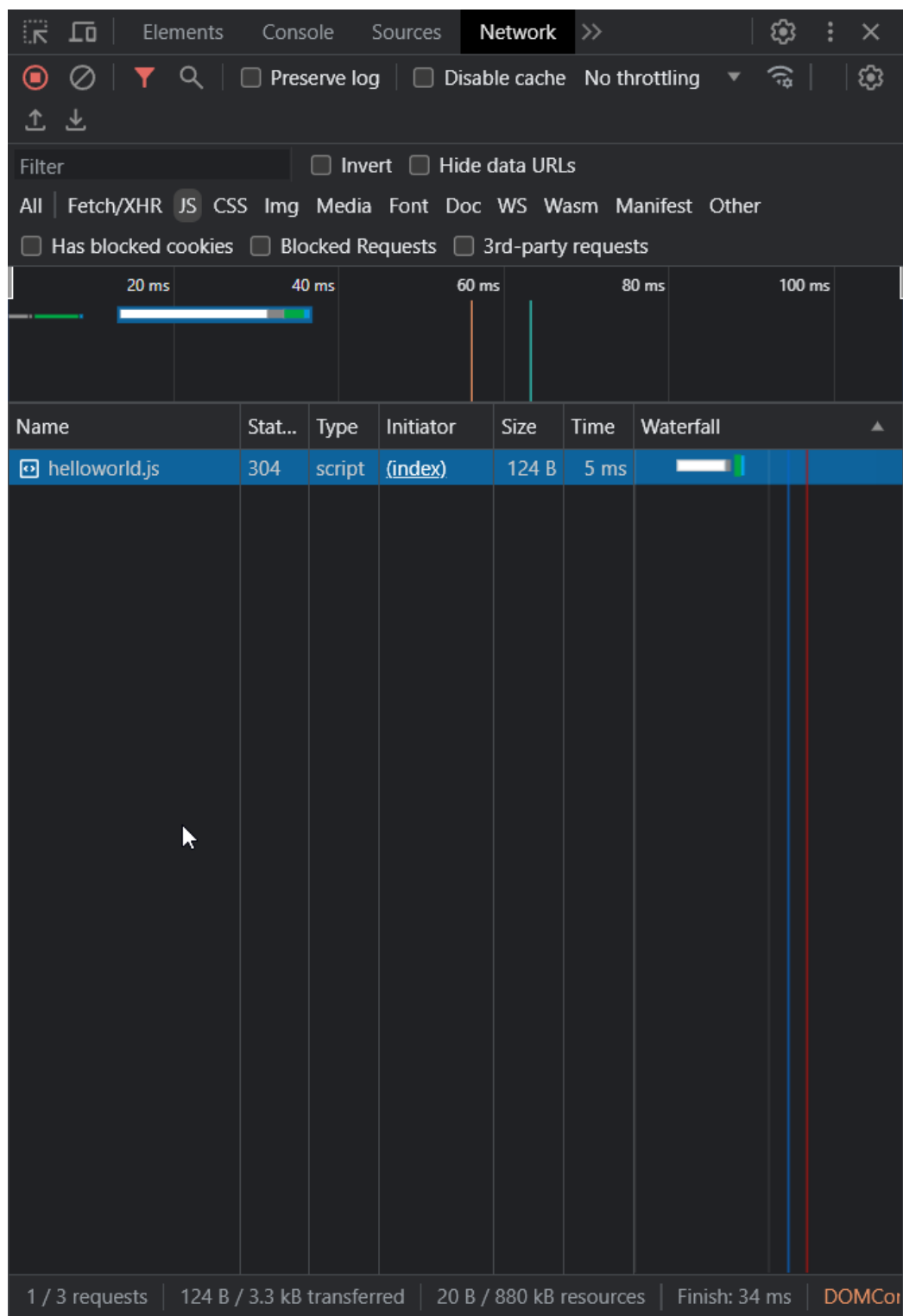


Рисунок 1 – helloworld.js

Добавим в файл код.

Листинг 1 – Файл «helloworld.js»

```
var groupmates = [  
  {  
    name: "Nikita",  
    group: "BFI2102",  
    age: 20,  
    marks: [4, 6, 5, 5, 4],  
  },  
  {  
    name: "Egor",  
    group: "BFI2102",  
    age: 20,  
    marks: [3, 2, 4, 4, 3],  
  },  
  {  
    name: "Danil",  
    group: "BFI2102",  
    age: 19,  
    marks: [2, 5, 4, 3, 5],  
  },  
  {  
    name: "Timofey",  
    group: "BFI2102",  
    age: 20,  
    marks: [5, 5, 5, 4, 5],  
  },  
];  
  
console.log(groupmates);
```

Чтобы можно было увидеть значение переменной (иными словами, для вывода переменной на экран), воспользуйтесь методом `console.log()`, который принимает через запятую аргументы, которые вы желаете увидеть:

```
console.log(groupmates);
```

На рисунке 2 показан результат.

```
helloworld.js:28
▼ (4) [{...}, {...}, {...}, {...}] ⓘ
  ▶ 0: {name: 'Nikita', group: 'BFI2102', age: 20, marks: Array(5)}
  ▶ 1: {name: 'Egor', group: 'BFI2102', age: 20, marks: Array(5)}
  ▶ 2: {name: 'Danil', group: 'BFI2102', age: 19, marks: Array(5)}
  ▶ 3: {name: 'Timofey', group: 'BFI2102', age: 20, marks: Array(5)}
    length: 4
  ▶ [[Prototype]]: Array(0)
```

Рисунок 2 – Массив groupmates

Создадим функцию, которая будет выводить в виде таблицы содержимое массива groupmates (вид таблицы должен напоминать тот, что использовался в первой лабораторной работе).

Листинг 2 – код для helloworld.js

```
var groupmates = [
  {
    name: "Nikita",
    group: "BFI2102",
    age: 20,
    marks: [4, 6, 5, 5, 4],
  },
  {
    name: "Egor",
    group: "BFI2102",
    age: 20,
    marks: [3, 2, 4, 4, 3],
  },
  {
    name: "Danil",
    group: "BFI2102",
    age: 19,
    marks: [2, 5, 4, 3, 5],
  },
  {
    name: "Timofey",
    group: "BFI2102",
    age: 20,
    marks: [5, 5, 5, 4, 5],
  },
];

var rpad = function (str, length) {
  str = str.toString();
  while (str.length < length) str = str + " ";
  return str;
};
```

```

var printStudents = function (students) {
  console.log(
    rpad("Имя студента", 15),
    rpad("Группа", 8),
    rpad("Возраст", 8),
    rpad("Оценки", 20)
  );

  for (var i = 0; i <= students.length - 1; i++) {
    console.log(
      rpad(students[i]["name"], 15),
      rpad(students[i]["group"], 8),
      rpad(students[i]["age"], 8),
      rpad(students[i]["marks"], 20)
    );
  }
  console.log("\n");
};

printStudents(groupmates);

```

Результат работы представлен на рисунке 3.

Имя студента	Группа	Возраст	Оценки	helloworld.js:35
Nikita	BFI2102	20	4,6,5,5,4	helloworld.js:43
Egor	BFI2102	20	3,2,4,4,3	helloworld.js:43
Danil	BFI2102	19	2,5,4,3,5	helloworld.js:43
Timofey	BFI2102	20	5,5,5,4,5	helloworld.js:43
				helloworld.js:50

Рисунок 3 – Массив groupmates

Создадим функцию, которая фильтрует студентов.

Листинг 3 – Файл «helloworld.js»

```

var groupmates = [
  {
    name: "Nikita",
    group: "BFI2102",
    age: 20,
    marks: [4, 6, 5, 5, 4],
  },
  {
    name: "Egor",
    group: "BFI2102",

```

```

    age: 20,
    marks: [3, 2, 4, 4, 3],
  },
  {
    name: "Danil",
    group: "BFI2102",
    age: 19,
    marks: [2, 5, 4, 3, 5],
  },
  {
    name: "Timofey",
    group: "BFI2102",
    age: 20,
    marks: [5, 5, 5, 4, 5],
  },
];

var rpad = function (str, length) {
  str = str.toString();
  while (str.length < length) str = str + " ";
  return str;
};

var printStudents = function (students) {
  console.log(
    rpad("Имя студента", 15),
    rpad("Группа", 8),
    rpad("Возраст", 8),
    rpad("Оценки", 20)
  );

  for (var i = 0; i <= students.length - 1; i++) {
    console.log(
      rpad(students[i]["name"], 15),
      rpad(students[i]["group"], 8),
      rpad(students[i]["age"], 8),
      rpad(students[i]["marks"], 20)
    );
  }
  console.log("\n");
};

printStudents(groupmates);

function filterStudents(age) {
  return groupmates.filter((student) => student.age === age);
}

console.log("Найдем студентов которым 20");
printStudents(filterStudents(20));

```

Имя студента	Группа	Возраст	Оценки	helloworld.js:35
Nikita	BFI2102	20	4,6,5,5,4	helloworld.js:43
Egor	BFI2102	20	3,2,4,4,3	helloworld.js:43
Danil	BFI2102	19	2,5,4,3,5	helloworld.js:43
Timofey	BFI2102	20	5,5,5,4,5	helloworld.js:43
				helloworld.js:50
Найдем студентов которым 20				helloworld.js:59
Имя студента	Группа	Возраст	Оценки	helloworld.js:35
Nikita	BFI2102	20	4,6,5,5,4	helloworld.js:43
Egor	BFI2102	20	3,2,4,4,3	helloworld.js:43
Timofey	BFI2102	20	5,5,5,4,5	helloworld.js:43
				helloworld.js:50

Рисунок 4 – Фильтрация массива groupmates

Добавим кнопки «свернуть» и «развернуть» для нашего проекта. Для этого создадим и подключим файл fold-post.js.

Добавим в данный файл событие нажатия кнопки. Поставим точку останова в данное событие. Нажмем на кнопку. Результат представлен на рисунке 5.

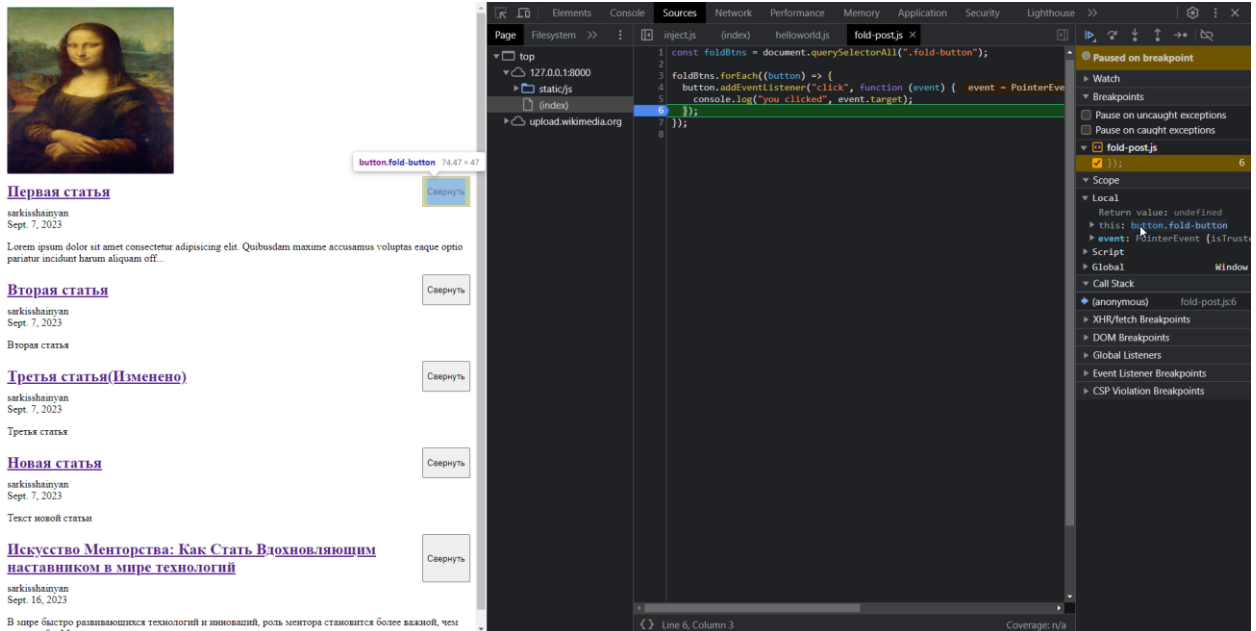


Рисунок 4 – Событие нажатия кнопки

Добавим код, выполняющий сворачивание/разворачивание информации о статье.

Листинг 4 – Файл «fold-post.js»

```
const foldBtns = document.querySelectorAll(".fold-button");

foldBtns.forEach((button) => {
  button.addEventListener("click", function (event) {
    if (event.target.className === "fold-button folded btn btn-outline") {
      event.target.innerHTML = "свернуть";
      event.target.className = "fold-button btn btn-outline";
      var displayState = "block";
    } else {
      event.target.innerHTML = "развернуть";
      event.target.className = "fold-button folded btn btn-outline";
      var displayState = "none";
    }

    event.target.parentElement.parentElement.getElementsByClassName(
      "article-info"
    )[0].style.display = displayState;
    event.target.parentElement.parentElement.getElementsByClassName(
      "article-text"
    )[0].style.display = displayState;
  });
});
```

На рисунках 4-5 показан результат.

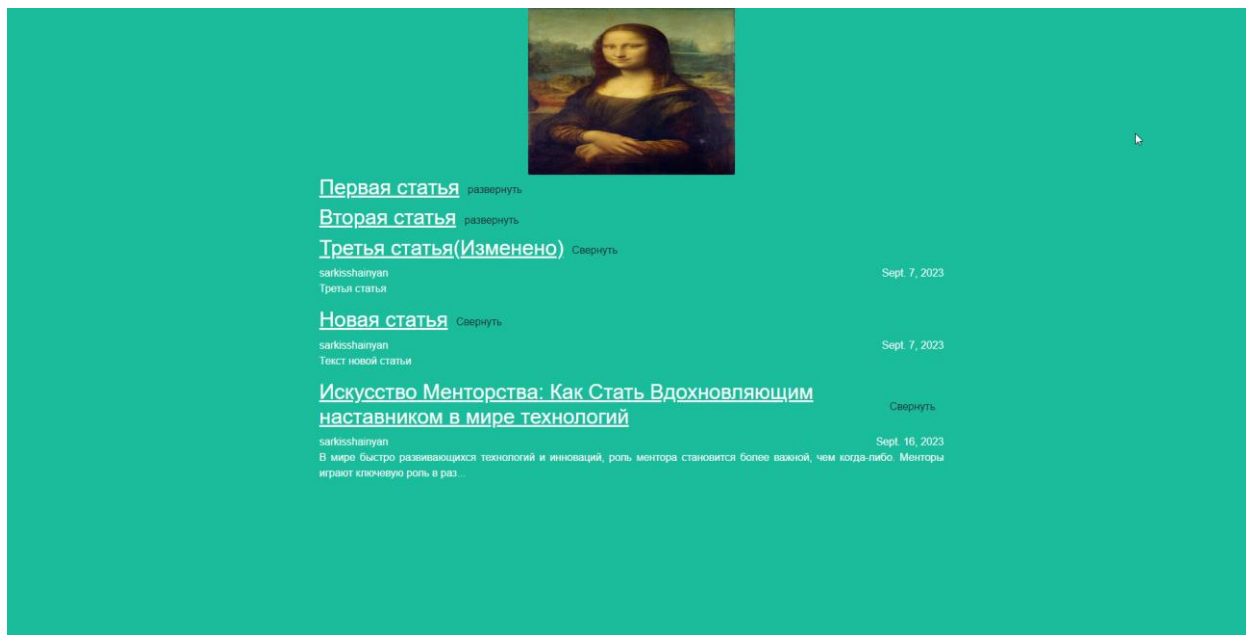


Рисунок 4 – Сворачивание текста

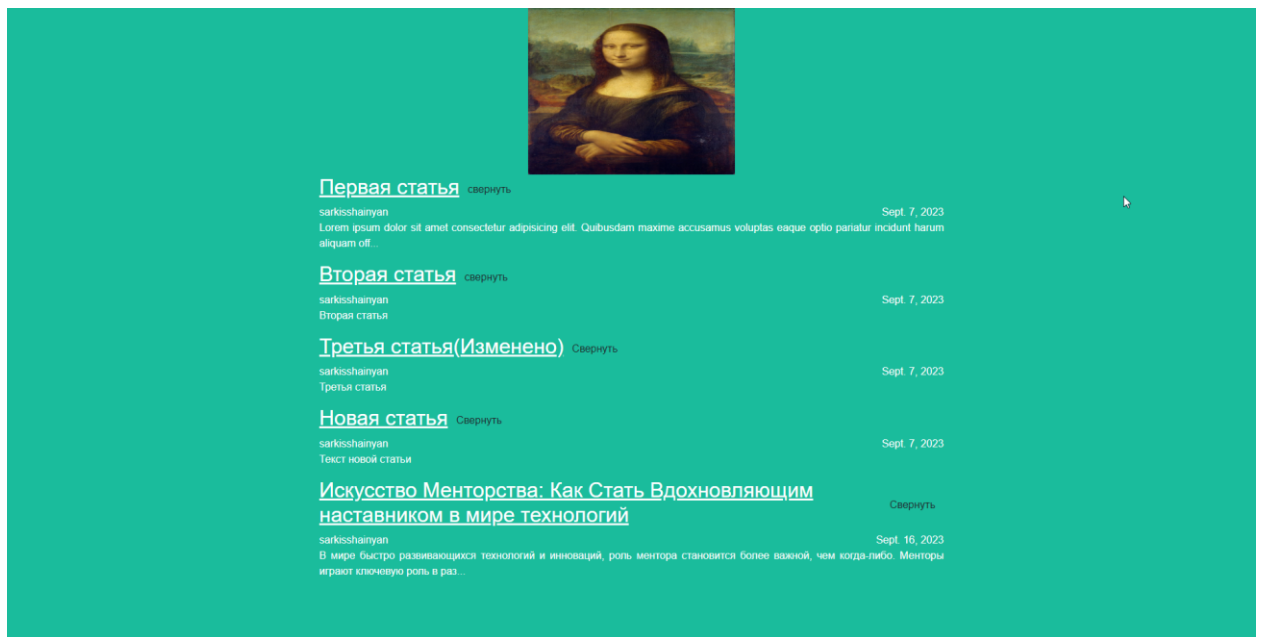


Рисунок 5 – Разворачивание текста

Добавим укороченный вариант выполнения данной возможности. Для этого добавим стили к css-классу и будем добавлять/удалять его при нажатии на кнопку.

Листинг 4 – Файл «article.css»

```
body {
  background: #1abc9c;
  font-family: Arial, Helvetica, sans-serif;
  color: #ffffff;
}
img {
  display: block;
  width: 318px;
  margin-left: auto;
  margin-right: auto;
}
.archive {
  width: 960px;
  margin-left: auto;
  margin-right: auto;
}
.post-title a {
  color: #ffffff;
}
.article-author {
  width: 50%;
  float: left;
}
.article-created-date {
  text-align: right;
```

```
}
.article-image {
  display: block;
  width: 318px;
  margin-left: 0;
}
.link {
  color: white;
  font-weight: bold;
  position: absolute;
  right: 470px;
  top: 180px;
}
.article-border p {
  text-align: right;
}
.article-text {
  width: 960px;
  text-align: justify;
}
.content {
  text-align: center;
  padding-top: 70px;
}
input[name="title"] {
  padding: 5px;
  margin-bottom: 10px;
  border: 1px solid #888;
  outline: none;
  -moz-appearance: none;
  width: 200px;
  text-align: center;
  border-radius: 40px;
}
textarea[name="text"] {
  padding: 25px;
  margin-bottom: 10px;
  border: 1px solid #888;
  outline: none;
  -moz-appearance: none;
  width: 650px;
  height: 350px;
  resize: none;
  border-radius: 40px;
  scrollbar-width: thin;
}
.create {
  display: flex;
  flex-direction: column;
  align-items: center;
}
```

```

.save_button {
  padding: 10px;
  width: 150px;
  background-color: white;
  border: none;
  border-radius: 40px;
  color: #1abc9c;
  font-weight: bold;
  letter-spacing: 0.06em;
  margin-top: 10px;
}
.save_button:hover {
  color: white;
  background-color: #1abc9c;
  box-shadow: 1px 1px 10px 10px;
  transition-duration: 0.3s;
}

.article_hidden .article-info {
  display: none;
}

.article_hidden .article-text {
  display: none;
}

```

Листинг 5 – Файл «fold-post.js»

```

const foldBtns = document.querySelectorAll(".fold-button");

foldBtns.forEach((button) => {
  button.addEventListener("click", function (event) {
    // if (event.target.className === "fold-button folded btn btn-outline") {
    //   event.target.innerHTML = "свернуть";
    //   event.target.className = "fold-button btn btn-outline";
    //   var displayState = "block";
    // } else {
    //   event.target.innerHTML = "развернуть";
    //   event.target.className = "fold-button folded btn btn-outline";
    //   var displayState = "none";
    // }

    // event.target.parentElement.parentElement.getElementsByClassName(
    //   "article-info"
    // )[0].style.display = displayState;
    // event.target.parentElement.parentElement.getElementsByClassName(
    //   "article-text"
    // )[0].style.display = displayState;
  });
});

```

```
event.target.innerHTML =  
    event.target.innerHTML === "Свернуть" ? "Развернуть" : "Свернуть";  
event.target.parentElement.parentElement.classList.toggle("article_hidden");  
});  
});
```

Вывод

В данной лабораторной работе мы изучили основы JavaScript, научились создавать простейшие функции и использовать базовые операторы. Также были изучена работа с элементами DOM с помощью JavaScript.