

Федеральное государственное образовательное бюджетное учреждение  
высшего образования  
**«Финансовый университет при Правительстве  
Российской Федерации»  
(Финансовый университет)**

**Колледж информатики и программирования**

ПМ.08      Разработка      кода  
информационных систем

УТВЕРЖДАЮ

Председатель предметно-цикловой  
комиссии информационных систем  
и программирования

Группа: ЗИСИП-622

\_\_\_\_\_ Т.Г. Аксёнова

« \_\_\_\_ » \_\_\_\_\_ 2025 г.

**ОТЧЁТ ПО ЛАБОРАТОРНОЙ РАБОТЕ №2**

Преподаватель

\_\_\_\_\_ Р. Р. Абзалимов

Исполнитель

\_\_\_\_\_ Л. Д. Слепцов

Оценка: \_\_\_\_\_

« \_\_\_\_ » \_\_\_\_\_ 2025 г.

Москва

2025

## Цель работы

Познакомиться с методом `document.getElementById`, который позволяет получить доступ к элементам HTML-документа по их идентификатору (ID) и изменять их содержимое. В этой лабораторной работе мы создадим простую веб-страницу с несколькими счетчиками, которые можно увеличивать и уменьшать с помощью кнопок.

## Ход работы

1. Что такое DOM и как он связан с HTML?

DOM (Document Object Model) — это программный интерфейс для веб-документов. Он представляет структуру документа в виде объекта, что позволяет языкам программирования, таким как JavaScript, взаимодействовать с содержимым, структурой и стилями документа.

HTML используется для создания веб-страниц. Когда браузер загружает HTML, он создает DOM — дерево объектов, где каждый элемент HTML становится узлом. Это позволяет JavaScript взаимодействовать с содержимым и структурой страницы, а также отслеживать действия пользователя.

2. Как с помощью JavaScript получить элемент по его ID? Приведите пример кода.

Чтобы получить элемент по его ID с помощью JavaScript, можно использовать метод `document.getElementById()`

```
<!DOCTYPE html>
<html>
<body>
  <span id="count">100</span>
  <button onclick="Counter()">+</button>

  <script>
    function Counter() {
      let counterElement = document.getElementById("count");
      let currentValue = parseInt(counterElement.innerHTML);
      counterElement.innerHTML = currentValue + 1;
    }
  </script>
</body>
</html>
```

```
    }  
  </script>  
</body>  
</html>
```

3. Что такое событие в JavaScript? Приведите примеры событий.

Событие в JavaScript — это любое действие, которое происходит в браузере и может быть обработано программой. События могут быть вызваны различными взаимодействиями пользователей с веб-страницей, такими как:

- Нажатие кнопки мыши
- Перемещение мыши
- Нажатие клавиш на клавиатуре
- Загрузка страницы
- Изменение размера окна

```
<!DOCTYPE html>  
<html>  
<body>  
  <button onclick="clicker()">Выполнить onclick</button>  
  <button onmousedown="mouser()">Выполнить onmousedown</button>  
  
  <script>  
    function clicker() {  
      alert("onclick работает!");  
    }  
  
    function mouser() {  
      alert("onmousedown работает!");  
    }  
  </script>  
</body>  
</html>
```

4. Как добавить обработчик события на кнопку в JavaScript? Напишите код.

Чтобы добавить обработчик события на кнопку в JavaScript, вы можете использовать метод `addEventListener`. Этот метод позволяет вам установить функцию, которая будет вызвана при возникновении определенного события на целевом элементе.

```

<!DOCTYPE html>
<html>
<body>
    <button id="event_button">Выполнить addEventListener</button>

    <script>
        let event_button = document.getElementById('event_button');
        event_button.addEventListener('click', function() {
            alert('addEventListener работает!');
        });
    </script>
</body>
</html>

```

5. Как изменить текст элемента с помощью JavaScript? Напишите соответствующий код.

Чтобы изменить текст элемента на веб-странице с помощью JavaScript, вы можете использовать свойство `textContent` или `innerHTML` элемента.

```

<!DOCTYPE html>
<html>
<body>
    <span id="change_span">Исходный текст</span>
    <button onclick="changeText()" id="change_button">Изменить
    текст</button>

    <script>
        function changeText() {
            document.getElementById('change_span').innerText = 'Новый
    текст';
        }
    </script>
</body>
</html>

```

6. Что такое массивы в JavaScript и как их использовать для работы с элементами DOM?

Массивы в JavaScript — это структуры данных, которые позволяют хранить упорядоченные коллекции элементов. Массивы могут содержать данные разных типов, включая числа, строки, объекты и даже другие массивы.

Для работы с элементами DOM массивы могут быть полезны, когда нужно хранить или обрабатывать набор элементов, например, список элементов на странице.

```
<!DOCTYPE html>
<html>
<body>
  <div class="item">Элемент 1</div>
  <div class="item">Элемент 2</div>
  <div class="item">Элемент 3</div>
  <button onclick="changeArrayText()" id="buttonArray">Изменить
текст</button>

  <script>
    function changeText() {
      document.getElementById('change_span').innerText = 'Новый
текст';
    }
  </script>
</body>
</html>
```

7. Как удалить элемент из DOM с помощью JavaScript? Приведите пример.

Чтобы удалить элемент из DOM с помощью JavaScript, можно использовать метод `remove()`, который доступен у объекта элемента.

```
<!DOCTYPE html>
<html>
<body>
  <span id="remove_span">Исходный текст</span>
  <button onclick="removeText()" id="remove_button">Удалить
текст</button>

  <script>
    function removeText() {
      let element = document.getElementById('remove_span');
      element.remove();
    }
  </script>
</body>
</html>
```