

ДИСЦИПЛИНА	Фронтенд и бэкенд разработка
ИНСТИТУТ	Институт перспективных технологий и индустриального программирования
КАФЕДРА	Кафедра индустриального программирования
ВИД УЧЕБНОГО МАТЕРИАЛА	Практические занятия
ПРЕПОДАВАТЕЛЬ	Загородних Николай Анатольевич
СЕМЕСТР	1 семестр, 2024-2025 гг.

Практическое занятие 13. Отладка JavaScript-сценариев и создание обработчиков событий для элементов веб-документа

Изучаемые вопросы

1. Типы событий в JavaScript.
2. Понятие обработчика событий.
3. Создание скриптов, реагирующих на события.
4. Отладка JavaScript-сценариев.
5. Изменение элементов страницы через обработчики событий.

Краткая теория

Типы событий

- Кликовые события - click, dblclick
- События клавиатуры - keydown, keyup, keypress
- События мыши - mouseover, mouseout, mousemove
- События формы - submit, change, focus, blur
- События загрузки - load, DOMContentLoaded

Обработчик событий

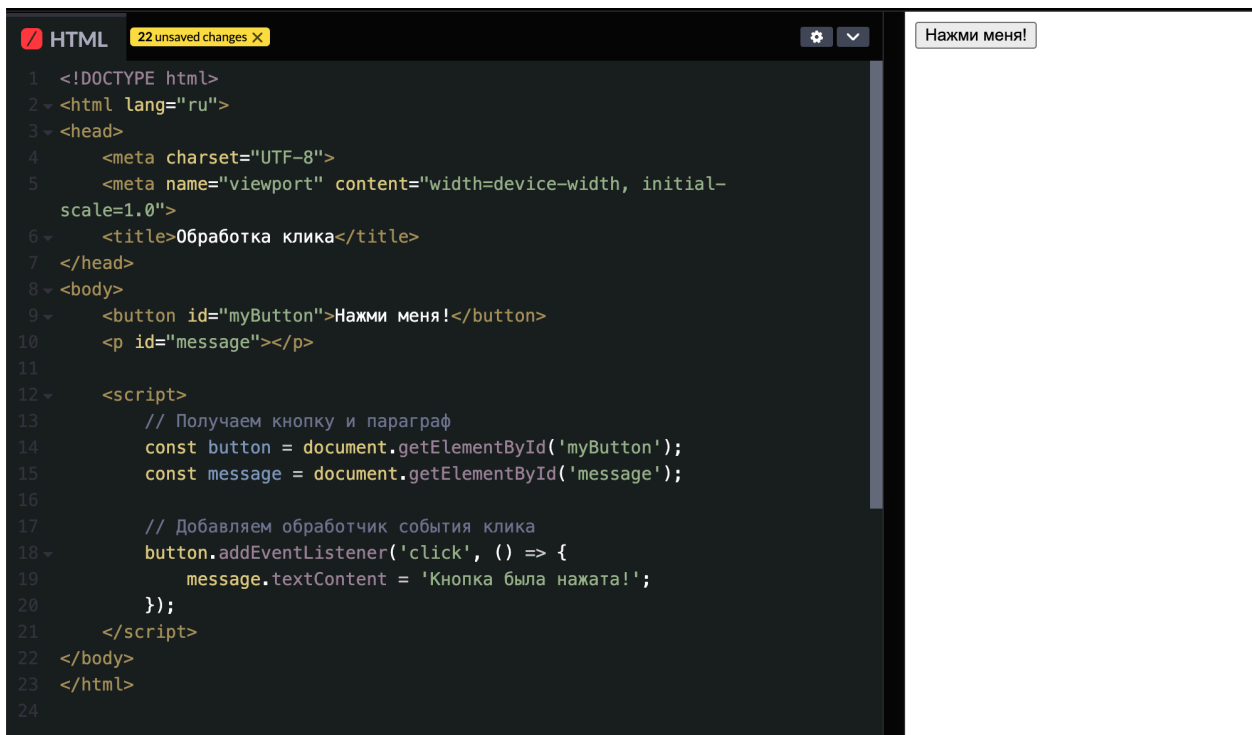
Обработчик события — это функция, которая выполняется при возникновении определенного события на элементе. Обработчики можно добавлять с помощью методов `addEventListener` или атрибутов HTML, таких как `onclick`.

Скрипты по событию

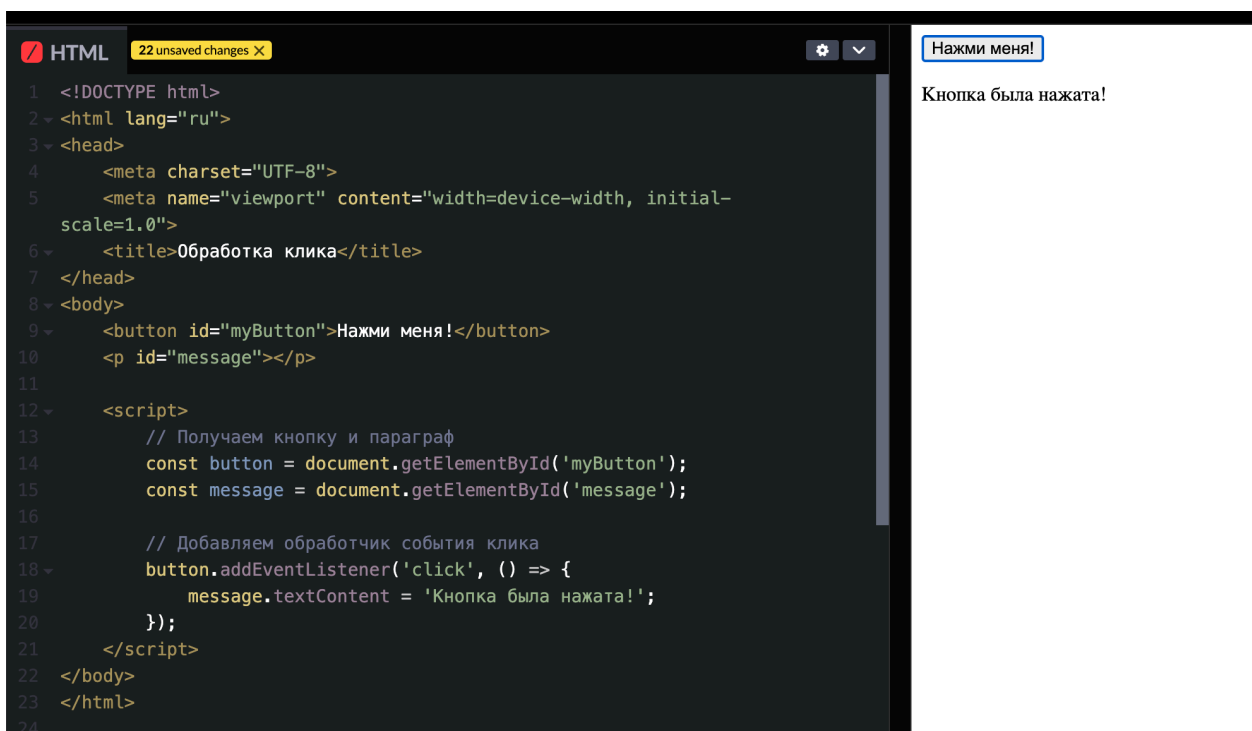
Скрипты могут изменять содержимое страницы, добавлять новые элементы или изменять стили в ответ на действия пользователя.

Обработчики событий — это функции, которые выполняются в ответ на определённые события. Они могут быть добавлены к элементам с помощью методов, таких как `'addEventListener()'` или через атрибуты HTML.

Пример. Обработка клика по кнопке



```
1 <!DOCTYPE html>
2 <html lang="ru">
3 <head>
4   <meta charset="UTF-8">
5   <meta name="viewport" content="width=device-width, initial-
    scale=1.0">
6   <title>Обработка клика</title>
7 </head>
8 <body>
9   <button id="myButton">Нажми меня!</button>
10  <p id="message"></p>
11
12  <script>
13    // Получаем кнопку и параграф
14    const button = document.getElementById('myButton');
15    const message = document.getElementById('message');
16
17    // Добавляем обработчик события клика
18    button.addEventListener('click', () => {
19      message.textContent = 'Кнопка была нажата!';
20    });
21  </script>
22 </body>
23 </html>
24
```



```
1 <!DOCTYPE html>
2 <html lang="ru">
3 <head>
4   <meta charset="UTF-8">
5   <meta name="viewport" content="width=device-width, initial-
    scale=1.0">
6   <title>Обработка клика</title>
7 </head>
8 <body>
9   <button id="myButton">Нажми меня!</button>
10  <p id="message"></p>
11
12  <script>
13    // Получаем кнопку и параграф
14    const button = document.getElementById('myButton');
15    const message = document.getElementById('message');
16
17    // Добавляем обработчик события клика
18    button.addEventListener('click', () => {
19      message.textContent = 'Кнопка была нажата!';
20    });
21  </script>
22 </body>
23 </html>
24
```

Отладка сценариев

Отладка JavaScript-сценариев может выполняться с помощью инструментов разработчика в браузере (F12). Важно использовать консоль для отслеживания ошибок и логирования значений переменных.

Полезные источники информации

1. MDN Web Docs - События

<https://developer.mozilla.org/ru/docs/Web/Events>

2. Обработка событий

<https://habr.com/ru/articles/244041/>

https://developer.mozilla.org/ru/docs/Learn/JavaScript/Building_blocks/Events

<https://doka.guide/js/events/?ysclid=m2ywkoslq836552118>

Тренажеры

<https://codepen.io/pen>

Задание

Часть 1: Создание обработчиков событий

1. Добавьте обработчик события click на заголовок H1, который будет вызывать alert с сообщением "Вы кликнули на заголовок - так держать!".

2. На странице есть таблица с темами практик на первый семестр. Добавьте кнопку «Посмотреть практики второго семестра», и по нажатию на нее поменяйте информацию в таблице на темы второго семестра.

Темы практик на 2-й семестр:

1. Базовое бэкенд-приложение
2. HTTP-запросы
3. JSON и работа с ним
4. HTTP-ответы
5. Проектирование API
6. Роутинг и его настройка
7. NoSQL базы данных
8. Обеспечение авторизации и доступа пользователей
9. Работа сторонних сервисов уведомления и авторизации
10. Основы ReactJS
11. Работа с компонентами динамической DOM
12. Использование хуков в React
14. Основы микросервисной архитектуры
15. Разработка классических модулей веб-приложений
16. Разработка классических модулей веб-приложений

3. Добавьте обработчик события `mouseover` на фотографию студента, который будет увеличивать ее размер (например, увеличить ширину и высоту на 10%). Добавьте обработчик события `click` который при клике на фотографию студента заменит ее на фотографию его любимого преподавателя (используйте поиск Яндекса и социальные сети для поиска контента).

4. Добавьте обработчик события `dblclick` на фотографию студента, который будет вызывать `alert` с текстом «Не налегай, у меня не так много любимых преподавателей».

Часть 2: Отладка сценариев

1. Проверьте все написанные скрипты через инструменты разработчика, убедитесь, что нет ошибок в консоли.

2. Используйте отладку для проверки корректности работы всех функций, созданных в части 1.

Критерии оценивания

- Корректность выполнения задания (40%)
 - Все обработчики событий реализованы правильно.
 - Функции работают без ошибок.
- Качество кода (30%)
 - Код структурирован и читаем.
 - Используются комментарии для объяснения логики.
- Использование отладки (20%)
 - Применены инструменты разработчика для проверки работы скриптов.

Форма сдачи

Студенты должны представить свой код в виде файлов `.html` `.css` `.js` (в случае подключения внешних скриптов), который можно открыть в браузере + скриншоты консоли в процессе отладки. Также важно прокомментировать код, чтобы объяснить, что делает каждый элемент.