

Приднестровский государственный университет им. Т.Г. Шевченко

Физико-технический институт

**ОСНОВЫ ВЕБ-РАЗРАБОТКИ С ПРИМЕНЕНИЕМ
ИИ-АССИСТЕНТОВ**

Лабораторный практикум

Разработал:
ст. преподаватель
кафедры ИТ
Бричаг Д.В.

г. Тирасполь
2025 г.

Лабораторная работа №5

Введение в JavaScript.

Цель работы:

- Понять, какую роль JavaScript играет в веб-разработке и как он «оживляет» статичные HTML-страницы.
- Освоить базовые конструкции языка: переменные, функции, события.
- Научиться работать с браузерной консолью для выполнения кода и отладки.
- Получить навык использования ИИ для генерации и адаптации простых скриптов для решения практических задач.

Теоретическая справка

Что такое JavaScript и зачем он нужен?

Если HTML – это скелет страницы, а CSS – её внешний вид, то JavaScript (JS) – это её мышцы и нервная система. Это язык программирования, который выполняется прямо в браузере пользователя и позволяет делать страницы интерактивными:

- Реагировать на действия пользователя (клики, наведение мыши, ввод текста).
- Изменять содержимое и стили элементов прямо на лету, без перезагрузки страницы.
- Отправлять данные на сервер и получать их обратно.
- Создавать анимации, слайдеры, модальные окна и многое другое.

Фактически, любая динамика на современном сайте создана с помощью JavaScript. Кстати, в одной из предыдущих работ, создавая переключатель светлой/тёмной темы, вы уже сталкивались с JS – именно он отслеживал клик по кнопке и менял классы у элементов.

Браузерная консоль – главный инструмент разработчика

У каждого современного браузера есть инструменты разработчика, а в них – консоль. Это интерактивная среда, где можно:

- Выполнять любой JavaScript-код в реальном времени.
- Видеть ошибки, которые возникли в вашем скрипте.
- Выводить отладочную информацию с помощью команды `console.log()`.

Как открыть консоль:

- Нажмите клавишу F12.
- Либо используйте сочетание Ctrl + Shift + I (или Cmd + Opt + I на macOS).
- Либо кликните правой кнопкой мыши на странице → «Просмотреть код» или «Исследовать элемент» → перейдите на вкладку «Console».

Попробуйте прямо в консоли написать `2 + 2` и нажать Enter. А затем – `console.log('Привет, JavaScript!');`. Вы сразу увидите результат.

Основы взаимодействия со страницей

Чтобы «оживить» страницу, JavaScript должен уметь три вещи:

1. Находить нужные HTML-элементы.
2. Слушать события, которые на них происходят (например, клик).
3. Изменять элементы в ответ на эти события.

Для этого используется DOM (Document Object Model) — объектное представление вашей HTML-страницы, с которым и работает JavaScript. В предыдущей лабораторной вы уже использовали JS при переключении темы сайта.

Теперь мы научимся делать более сложные интерактивные элементы.

1. Поиск элемента:

```
// Находим кнопку по её уникальному id
const myButton = document.querySelector('#myButton');

// Находим первый попавшийся заголовок h1
const mainTitle = document.querySelector('h1');
```

2. Добавление обработчика событий:

```
// Говорим: "Когда по кнопке myButton произойдёт клик,  
// выполни функцию, указанную вторым аргументом"  
myButton.addEventListener('click', function() {  
  // Этот код выполнится только после клика  
  console.log('Кнопка была нажата!');  
});
```

3. Изменение элемента:

```
myButton.addEventListener('click', function() {  
  // Находим заголовок и меняем его текст  
  const mainTitle = document.querySelector('h1');  
  mainTitle.textContent = 'Текст изменён с помощью JS!';  
});
```

Этот простой пример — основа почти всей интерактивности в вебе.

Базовые конструкции языка

Переменные

```
let name = "Иван";  
const age = 20;
```

Условия

```
if (age >= 18) {  
  console.log("Совершеннолетний");  
} else {  
  console.log("Несовершеннолетний");  
}
```

Функции

```
function greet(user) {  
  alert("Привет, " + user + "!");  
}
```

Обработчики событий

```
document.querySelector("button").addEventListener("click", function() {  
    alert("Кнопка нажата!");  
});
```

Циклы

```
for (let i = 1; i <= 5; i++) {  
    console.log(i);  
}  
// Вывод: 1 2 3 4 5
```

Сначала выполняет тело цикла, а затем проверяет условие.

```
let count = 0;  
while (count < 3) {  
    console.log(count);  
    count++;  
}  
// Вывод: 0 1 2
```

Перебирает свойства объекта.

```
const user = { name: 'Alex', password: '123' };  
for (const key in user) {  
    console.log(user[key]);  
}  
// Вывод: Alex 123
```

Перебирает значения итерируемых объектов, таких как массивы.

```
const arr = [10, 20, 30];  
for (const value of arr) {  
    console.log(value);  
}  
// Вывод: 10 20 30
```

Практическая часть

Сформируем простое техническое задание. Необходимо:

Используя верстку из ЛР №2 (семантическая страница):

- добавить несколько интерактивных элементов с помощью JavaScript;
- написать скрипты при помощи ИИ;
- протестировать их работу через консоль.

Подготовка проекта

- Создайте папку lab5.
- Скопируйте в неё верстку из ЛР №2 (ваш адаптивный макет).
- Добавьте файл script.js и подключите его внизу HTML:

```
<script src="script.js"></script>
```

Использование ИИ для генерации кода

Примеры промптов:

Напиши JavaScript, который добавляет кнопку "Наверх". При прокрутке страницы вниз она появляется, при нажатии – плавно прокручивает вверх.

Сделай аккордеон на чистом JS: несколько заголовков, при клике открывается и закрывается блок текста.

Создай фотогалерею с фильтрацией по категориям. При выборе категории показывай только соответствующие фото.

Добавь обработчик: при клике на фото открывается модальное окно с увеличенным изображением и затемнением фона.

Кнопка «Наверх»

1. В HTML добавьте кнопку:

```
<button id="scrollTopBtn" class="scroll-btn">↑</button>
```

2. В script.js вставьте код, предложенный ИИ.

3. Убедитесь, что:

- Кнопка появляется при прокрутке вниз;
- При нажатии страница плавно прокручивается вверх.

Аккордеон с текстом

1. Добавьте несколько блоков в секцию контента:

```
<div class="accordion-item">  
  <h3 class="accordion-title">Заголовок 1</h3>  
  <div class="accordion-content">Текст 1</div>  
</div>
```

2. В script.js подключите аккордеон по шаблону ИИ.

3. Проверьте: открытие одного блока скрывает другие.

Фотогалерея с фильтрацией

1. Создайте контейнер с изображениями, например:

```
<div class="filters">  
  <button data-category="all">Все</button>  
  <button data-category="nature">Природа</button>  
  <button data-category="city">Город</button>  
</div>  
<div class="gallery">  
    
    
  ...  
</div>
```

2. Напишите код (через ИИ) для фильтрации изображений по категориям.

Модальное окно с увеличением фото

1. Добавьте контейнер для модального окна:

```
<div id="modal" class="modal">  
  <span class="close">&times;</span>  
  <img class="modal-content" id="modalImg">  
</div>
```

2. С помощью ИИ создайте обработчик:

- При клике на фото — открыть модалку с увеличенным изображением.
- При клике по крестику или фону — закрыть.

Проверка работы скриптов

- Откройте index.html.
- Проверьте, что все элементы работают: кнопка, аккордеон с текстом, фильтры фотогалереи, модальное окно.
- Убедитесь, что в консоли нет ошибок. В случае если есть, уточните у нейросети причину и как исправить.

Работа с git и GitHub

1. Инициализируйте репозиторий:

```
git init  
git add .  
git commit -m "Lab5: интерактивная страница на JS"
```

2. Создайте новый репозиторий на GitHub.
3. Свяжите локальный проект с GitHub:


```
git remote add origin https://github.com/<ваш_логин>lab5_repo.git
git branch -M main
git push -u origin main
```

Отчёт (в README.md)

В README.md добавить:

- Заголовок: *Лабораторная работа №5*
- Скриншоты работы интерактивных элементов.
- Ответы на вопросы:
 - Какие интерактивности добавлены на страницу?
 - Где использовался ИИ, а где код писался вручную?
 - Что показалось самым интересным или сложным?
 - Какие запросы делали LLM?

Результаты работы

В итоге у вас должно быть:

1. Страница из ЛР2, дополненная интерактивными элементами.
2. Подключённый файл script.js.
3. Репозиторий на GitHub с проектом.
4. README.md с отчётом и скриншотами.

Критерии оценки

- Добавлена кнопка «Наверх» (2 балла).
- Работает аккордеон (2 балла).
- Реализована фотогалерея с фильтрацией (2 балла).
- Реализовано модальное окно (2 балла).
- Код выложен на GitHub и оформлен отчёт (2 балла).