Лекция: Введение в JavaScript и основы программирования

## Цель занятия

Познакомить учеников с базовыми концепциями программирования на примере языка JavaScript. Научить писать простейшие скрипты: вывод сообщений, использование переменных, простые вычисления и условные конструкции. Эта база важна как будущим разработчикам, так и тестировщикам.

## Ключевые понятия

- JavaScript – язык, выполняющийся в браузере. Делает страницы интерактивными.  
- Подключение через <script> – сначала можно прямо в HTML, затем лучше выносить во внешний файл.  
- Переменные: let / const. Пример: let age = 25;  
- Типы данных: строки, числа, логические значения. Пример: "Hello", 42, true  
- Операции: +, -, \*, /, а также склеивание строк через +  
- console.log() – для вывода информации  
- Условия: if/else, логические операторы (===, >, <, &&, ||)  
- Циклы: for – пример вывода чисел от 1 до 5  
- Функции: function имя() { }, параметры и return

## Слайды и структура урока

Слайд 1: «JavaScript – оживляем страницу». Объяснение, зачем нужен JS и где он выполняется.  
Слайд 2: Первый скрипт – <script>console.log("Привет, мир!");</script>. Работа с консолью.  
Слайд 3: Переменные и типы. let/const. Примеры строк и чисел.  
Слайд 4: Операции – сложение, конкатенация. Вывод с пояснением.  
Слайд 5: Условия. Пример с возрастом и операторами сравнения.  
Слайд 6: Циклы – for с примером. Пояснение структуры цикла.  
Слайд 7: Функции – пример greet(username). Объяснение return и параметров.  
Слайд 8: Практика в консоли. Несколько простых заданий (вычисления, переменные, условия).  
Слайд 9: Итоги. Домашнее задание – скрипт с выводом чётных чисел от 1 до 10.

## Занятие 5: JavaScript и DOM

Цель: Научить работать с DOM – HTML-элементами через JavaScript. Рассмотреть DevTools браузера.  
  
Слайд 1: DOM – HTML представлен в виде дерева узлов. Объектная модель документа.  
Слайд 2: Поиск и изменение – document.getElementById, .innerText, querySelector  
Слайд 3: Демонстрация через консоль – изменение текста на странице вживую  
Слайд 4: Обработка событий – addEventListener, пример с кнопкой  
Слайд 5: Динамическое обновление – input + span + JS для приветствия  
Слайд 6: DevTools – вкладки Elements и Console. Работа с HTML и отладка JS  
Слайд 7: Debugger – как поставить breakpoint, шаг за шагом выполнять код  
Слайд 8: Отладка с ошибкой – как найти и исправить проблему по сообщению в консоли  
Слайд 9: Практика – форма с полем и кнопкой, обработка ввода через JS  
Слайд 10: Итоги. DevTools полезны и разработчику, и тестировщику

## Обмен данными между frontend и backend (введение)

Браузер (frontend) может отправлять и получать данные от сервера (backend).  
- Для этого используют HTTP-запросы: GET, POST и другие  
- В JavaScript это делается через fetch или библиотеки (например, axios)  
  
Пример:  
fetch('https://example.com/api/user')  
 .then(response => response.json())  
 .then(data => console.log(data));  
  
Сервер отправляет данные в формате JSON. Браузер может их использовать для отображения на странице.  
  
Это основа взаимодействия между частями приложения. Подробнее будем разбирать на следующем занятии.

## Домашнее задание

- Написать JS-скрипт, который:  
 • выводит сообщение в консоль  
 • определяет переменную с именем  
 • выводит приветствие  
 • проверяет возраст и выводит условие  
 • выводит чётные числа от 1 до 10  
  
- По желанию: создать HTML-файл с кнопкой и написать JS-код, который выводит alert при клике