**Конспект «Основы JavaScript»**

**JavaScript: что это такое и как подключить его на страницу**

Язык программирования JavaScript придумали специально для того, чтобы создавать интерактивные сайты.

Код на языке JavaScript называют скриптом. Его сохраняют в отдельный файл с расширением js, а чтобы запустить, подключают этот файл на страницу. В HTML для добавления JavaScript есть специальный тег:

<script src="адрес\_файла"></script>

Подключают скрипт обычно в самом конце страницы, перед закрывающим тегом </body>.

Программа на JavaScript — это последовательность инструкций, то есть указаний браузеру выполнить какие-то действия. Инструкции выполняются последовательно, сверху вниз.

Чтобы сказать JavaScript, что инструкция закончена, нужно поставить точку с запятой или перейти на новую строку. Новая строка правильно работает *в большинстве случаев*, а точка с запятой — **всегда**. Поэтому лучше ставить точку с запятой в конце каждой инструкции.

JavaScript не меняет исходный файл с разметкой, но, выполняя инструкции, меняет страницу прямо в браузере пользователя.

**Комментарии**

Комментарий — это текст, поясняющий код. Он не выводится в браузер и никак не влияет на работу программы. Инструкции внутри комментария не выполняются, поэтому комментарии часто используют, если нужно временно отключить часть кода.

В JavaScript есть два вида комментариев:

// Однострочные комментарии.

/\*

И многострочные.

Они могут отключить сразу несколько строк кода.

\*/

**Метод querySelector**

Чтобы найти на странице элемент, нужно использовать метод querySelector, он ищет по селектору:

document.querySelector('селектор');

Эта инструкция состоит из двух частей. Первая часть — элемент, внутри которого будет искать JavaScript. Словом document обозначается веб-страница, к которой скрипт подключён. Неважно, как называется файл на самом деле, в JavaScript это всегда «документ». Он является элементом-родителем для любого другого элемента на странице.

Вторая часть инструкции — это то, что нужно сделать. Её называют методом.

**Консоль**

Консоль — инструмент разработчика, который помогает тестировать код. Если во время выполнения скрипта возникнет ошибка, в консоли появится сообщение о ней. А ещё в консоль можно выводить текстовые подсказки. Чтобы вывести сообщение в консоль, нужно использовать console.log:

console.log('Привет от JavaScript!');

// Выведет: Привет от JavaScript!

console.log(document.querySelector('.page'));

// Выведет в консоль найденный элемент

**Переменная**

Переменная — способ сохранить данные, дав им понятное название.

Переменную можно создать, или **объявить**, с помощью ключевого слова let. За ним следует имя переменной. После объявления в переменную нужно записать, или **присвоить**, какое-то значение:

let header = document.querySelector('header');

Имя переменной может быть почти любым, но не должно начинаться с цифры, а из спецсимволов разрешены только '\_' и '$'. Ещё для именования переменных нельзя использовать [зарезервированные слова](https://developer.mozilla.org/ru/docs/Web/JavaScript/Reference/Lexical_grammar#%D0%9A%D0%BB%D1%8E%D1%87%D0%B5%D0%B2%D1%8B%D0%B5_%D1%81%D0%BB%D0%BE%D0%B2%D0%B0). Имена переменных чувствительны к регистру: header, Header и HEADER — это разные переменные. Имя переменной должно описывать то, что в ней хранится.

Когда в коде встречается переменная, браузер вместо её имени подставляет присвоенное ей значение. Когда мы используем переменную, снова писать let не нужно:

console.log(header);

Ключевое слово let появилось в JavaScript в 2015 году, до этого для объявления переменных использовалось слово var.

**Методы для изменения классов**

Чтобы убрать у элемента класс, нужно использовать метод classList.remove. Он убирает с элемента тот класс, который указан в скобках:

элемент.classList.remove('класс');

Чтобы добавить элементу класс, нужно использовать метод classList.add:

элемент.classList.add('класс');

Метод-переключатель classList.toggle убирает у элемента указанный класс, если он есть, и добавляет, если этого класса нет:

элемент.classList.toggle('класс');

**Свойство textContent**

У каждого элемента имеется множество свойств: его размеры, цвет и так далее. Свойство textContent хранит в себе текстовое содержимое элемента. Свойствам можно присваивать новые значения:

let paragraph = document.querySelector('p');

paragraph.textContent = 'Здесь был Кекс. Мяу!';

**Свойство value**

У полей ввода есть особое свойство — value. Оно хранит данные, введённые в поле. Мы можем вывести их прямо на страницу:

let input = document.querySelector('input');

paragraph.textContent = input.value;

**Конкатенация**

Операция, когда мы «склеиваем» несколько значений, называется **конкатенацией** и в JavaScript выполняется с помощью знака плюс.

let name = 'Кекс';

paragraph.textContent = 'Вас зовут ' + name + '. Хорошего дня!';

console.log(paragraph.textContent);

// Выведет: Вас зовут Кекс. Хорошего дня!

**Обработчики событий onclick и onsubmit**

JavaScript следит за всем, что происходит на странице. Клик по кнопке или отправка формы — это *событие*. Мы можем сказать JavaScript, что сделать, когда некое событие произойдёт. Для этого используют обработчики событий. Инструкции, которые должны будут выполниться, когда событие произойдёт, располагают между фигурных скобок.

Свойство onclick означает «по клику»:

let button = document.querySelector('button');

button.onclick = function() {

console.log('Кнопка нажата!');

};

При каждом клике по кнопке в консоли будет появляться новое сообщение Кнопка нажата!.

За обработку отправки формы отвечает свойство onsubmit:

let form = document.querySelector('form');

form.onsubmit = function() {

console.log('Форма отправлена!');

};

После отправки формы в консоли появится сообщение Форма отправлена!.