

***ОСНОВЫ РАЗРАБОТКИ
WEB-ИНТЕРФЕЙСОВ
(2 СЕМЕСТР)***

Основы JavaScript

ПРОФЕССИОНАЛЬНАЯ ЗАДАЧА

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

ИСТОРИЯ JAVASCRIPT

JavaScript разработан в компании Netscape и по синтаксису напоминает C, C++ и Java некоторыми программными конструкциями: циклы, условные конструкции, операторы

В соответствии со стандартом ECMA-262 язык официально называется ECMAScript

Брендан Айк

Брендан Айк – родился в 1961 году в Питтсбурге (Пенсильвания).

Айк начал карьеру в Silicon Graphics, где проработал семь лет. Следующие три года проработал в MicroUnity Systems Engineering.

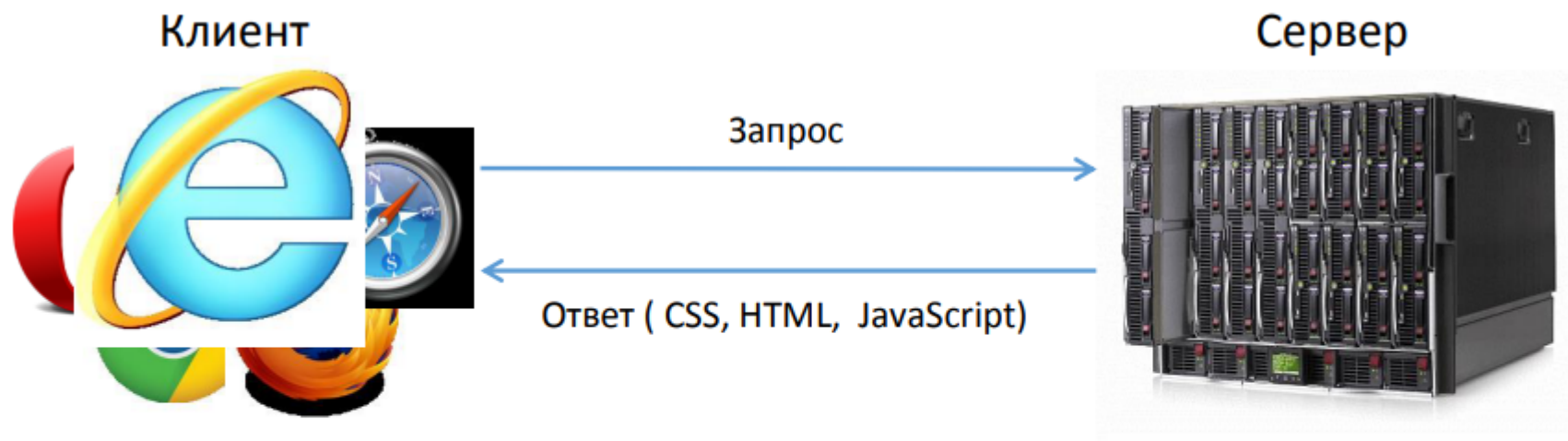


В апреле 1995 года начал работу в Netscape , где разработал язык программирования JavaScript (изначально названный «Mocha», затем переименован в «LiveScript») для браузера Netscape Navigator.

В начале 1998 года участвовал в открытии mozilla.org в качестве главного архитектора. Когда в июле 2003 года AOL закрыла подразделение Netscape, Айк перешёл в Mozilla Foundation.

В августе 2005 года, проработав главным технологом и членом совета директоров Mozilla Foundation, Брендан стал техническим директором Mozilla Corporation.

КЛИЕНТСКИЙ JAVASCRIPT



Интерпретатор JavaScript, которым оснащены браузеры, позволяет выполнять клиентские сценарии загруженные вместе с разметкой страницы.

РОЛЬ КЛИЕНТСКОГО JAVASCRIPT

- Пользовательский интерфейс
 - Создание визуальных эффектов (анимация)
 - Выполнение не сложных вычислений.
 - Проверка данных введенных пользователем.
 - Манипуляция данными введенными пользователем в формы.
 - Поиск по данным, встроенным в загружаемую страницу.
 - Сохранение данных в cookies.
- Динамическое изменение содержимого страницы (DHTML).
- Прямое взаимодействие с сервером без перезагрузки всей страницы (AJAX).

Пример клиентского JavaScript

JavaScript код выполняется на стороне клиента Web-браузером.

```
<html>
<head>
  <title>Клиентский JavaScript</title>
</head>
<body>
  <h1>Клиентский JavaScript</h1>
  <script>
    for (var i = 1; i < 10; i++) {
      document.write(i + "<br>");
    }
  </script>
</body>
</html>
```



ЛЕКСИЧЕСКИЕ СТРУКТУРЫ

Лексическая структура языка программирования – это набор элементарных правил, определяющих, как пишутся программы на языке.

Лексическая структура JavaScript:

- Чувствителен к регистру.
- Инструкции должны заканчиваться точкой с запятой или начинаться с новой строки.
- Игнорирует пробелы и табуляции.

ЗАРЕЗЕРВИРОВАННЫЕ СЛОВА

В JavaScript имеется набор зарезервированных слов

break	do	if	switch	typeof
case	else	in	this	var
catch	false	instanceof	throw	void
continue	finally	new	true	while
default	for	null	try	with
delete	function	return		

ЗАРЕЗЕРВИРОВАННЫЕ СЛОВА

ECMAScript v3

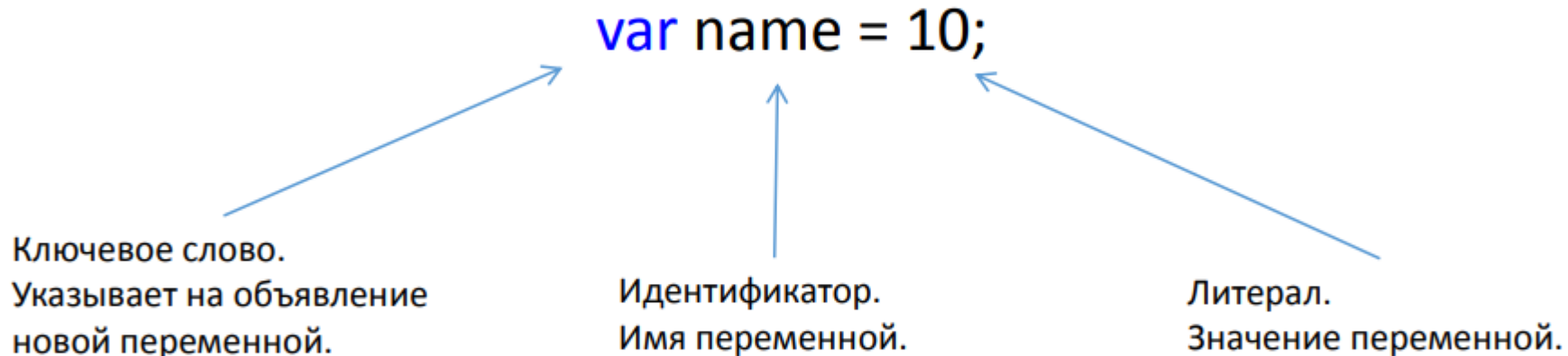
Зарезервированные ключевые слова в ECMAScript v3 в данный момент в JavaScript не используются, но зарезервированы в качестве возможных будущих расширений языка.

abstract	double	goto	navite	static
boolean	enum	implements	package	super
byte	export	import	private	synchroniz
char	extends	int	protected	ed
class	final	interface	public	throws
const	float	long	short	transient
debugger				volatile

ПЕРЕМЕННЫЕ

Создание переменных

Переменная – это символ или слово, обозначающее некоторое значение.



JavaScript – динамически типизированный, а не строго типизированный язык программирования.

ПЕРЕМЕННЫЕ

Правила именования переменных

Идентификатор должен быть одним словом:

myName



My Name

Имя не может начинаться с цифры:

element21



21element

Первым символом имени может быть символ '\$' или '_' :

\$str

_name

При именовании часто используются правила camelCase и PascalCase.

ПЕРЕМЕННЫЕ

Переменная — это способ сохранить данные, дав им понятное название.

```
let variableName = 'Я значение переменной!';
```

Есть устаревший способ объявления переменной - с помощью ключевого слова `var` вместо `let`. Но он устарел, и мы не будем его использовать.

КОНСТАНТЫ

Значение константы — всегда неизменно.

Они объявляются через **const**:

```
const gravityAcceleration = 9.8;
```

```
// ускорение свободного падения Земли
```

Константы уместно использовать, когда мы заранее знаем, что данные не изменятся. Если сомневаетесь — пишите **let**.

Переменные и константы

Обратите внимание, что `let/const` мы пишем только при объявлении переменной в первый раз. После этого мы используем переменную/константу, просто указывая её имя.

```
const myName = 'Вася';  
let myAge = 15;  
myName = 'Кекс';  
myAge = 3;
```

А так же каждую команду нужно как-то отделять от другой, для этого каждая строка должна заканчиваться точкой с запятой «;».

ТИПЫ ДАННЫХ

В языке JavaScript существуют типы:

- Три элементарных типа данных: числа, строки и значения логической истинности.
- Два тривиальных типа данных: `null` и `undefined`, каждый из которых определяет только одно значение.
- **Составной тип данных** – объект (`Object`), представляет собой коллекцию значений. Объекты в JavaScript имеют двойственную природу: может быть представлен как не упорядоченная коллекция именованных значений и упорядоченная коллекция пронумерованных значений (array-массивы).
- **Специальный тип объекта** – функции, с которым связан специальный код. Функция может вызываться для выполнения определенной операции.
- В языке JavaScript все числа хранятся в формате `float64` (8 байт) с плавающей точкой. В этом формате не всегда возможны точные вычисления.
- **Строки** – тип данных для представления текста (в строку может входить любой **UNICODE** символ) строковые литералы нужно заключать в согласованные пары одинарных или двойных кавычек.
- **Логический тип данных** – тип данных, которые могут принимать два возможных значения `true` и `false`.
- **Тип `null`** – это тип данных, указывающий на отсутствие объекта.
- **Тип `undefined`** – это тип данных, который используется для переменных или свойств объекта, значения для которых не определены.

ФУНКЦИИ ДЛЯ РАБОТЫ С ДИАЛОГОВЫМИ ОКНАМИ

alert() – Выводит модальное окно с сообщением.

- Посетитель не сможет продолжить работу, пока не нажмет на кнопку "ОК" в модальном окне.

Пример: `alert('Добрый день');`

prompt() – Выводит сообщение в окне с текстовым полем и двумя кнопками: "ОК" и "ОТМЕНА".

- Возвращает введенное значение или `null`, если посетитель нажал на кнопку "ОТМЕНА".
- Как и в `alert()`, окно – модальное, то есть посетитель не может делать ничего другого, пока не выберет одну из кнопок.

Пример: `result = prompt("text", "value");`

confirm() – выводит окно с вопросом **question** с двумя кнопками: "ОК" и "ОТМЕНА"

Пример: `var a = confirm("Вы – администратор?");`

РЕЗЮМЕ

JavaScript – прототипно-ориентированный скриптовый язык программирования. В соответствии со стандартом ECMA-262 язык официально называется ECMAScript.

JavaScript – это не Java, язык Java был разработан в компании Sun Microsystems, а JavaScript в компании Netscape и кроме некоторой схожести синтаксиса и способности предоставлять исполняемое содержимое для Web-браузеров, эти два языка между собой ничего не связывает.

JavaScript – язык, чувствительный к регистру (case sensitivity), также игнорирует пробелы, табуляции и перевод строки.

Простые JavaScript инструкции обычно завершаются символом – (;) точка с запятой. Если точка с запятой не используется, то каждая инструкция должна начинаться с новой строки.

Что умеет JavaScript

Современный JavaScript – это язык программирования общего назначения. Он не предоставляет низкоуровневых средств работы с памятью, процессором, так как изначально был ориентирован на браузеры.

В браузере, JavaScript умеет делать все, что относится к манипуляции со страницей, взаимодействию с посетителем и, в какой-то мере, с сервером:

- Создавать новые HTML-теги, удалять существующие, менять стили элементов, прятать, показывать элементы и т.п.

РЕЗЮМЕ

- Реагировать на действия пользователя, обрабатывать клики мыши, перемещение курсора, нажатие на клавиатуру и т.п.
- Посылать запросы на сервер и загружать данные без перезагрузки страницы (технология "AJAX").
- Получать и устанавливать cookies-файлы, запрашивать данные, выводить сообщения.

Что не умеет JavaScript

- JavaScript не может читать/записывать произвольные файлы на жесткий диск, копировать их или вызывать программы. Он не имеет прямого доступа к операционной системе. Это сделано для безопасности пользователей, чтобы злоумышленник не мог с помощью JavaScript получить личные данные или как-то навредить компьютеру пользователя.
- Современные браузеры могут работать с файлами, но эта возможность ограничена специально выделенной директорией – песочницей (Isolated Storage).
- JavaScript, работающий в одной вкладке, не может общаться с другими вкладками и окнами. За исключением случая, когда он сам открыл это окно.
- Современные браузеры поддерживают интерфейс postMessage, который позволяет наладить взаимодействие между страницами. Страница, загруженная с одного домена не может отправлять сообщения или вызывать код загруженный с другого домена, а postMessage позволяет обойти это ограничение.
- Из JavaScript можно легко посылать запросы на сервер, с которого пришла страница. Запрос к другому домену возможен, но менее удобен, т.к. имеются ограничения.

РЕЗЮМЕ

В чем уникальность JavaScript?

- Полная интеграция с HTML/CSS.
- Поддерживается всеми распространенными браузерами и включен по умолчанию.

Идентификатор – это имя. Идентификаторы в JavaScript выступают в качестве названия переменных и функций, а также меток некоторых циклов.

РЕЗЮМЕ

Код на языке JavaScript называют скриптом.

Его сохраняют в отдельный файл с расширением .js



img



index.html



script.js



setting.css



style.css

РЕЗЮМЕ

В HTML для добавления JavaScript есть специальный тег:

```
<script src=«адрес_файла»></script>
```

```
<body>
```

```
<!-- Содержимое страницы -->
```

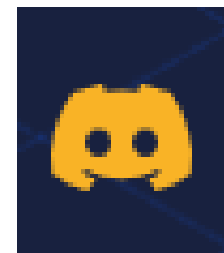
```
<script src=«script.js»></script>
```

```
</body>
```

*ЗАДАВАЙТЕ ВОПРОСЫ ПО
ЛЕКЦИИ!*



канал в дискорд



**[https://discord.gg/
wKft5yhH47](https://discord.gg/wKft5yhH47)**