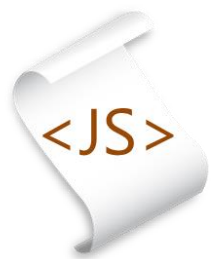


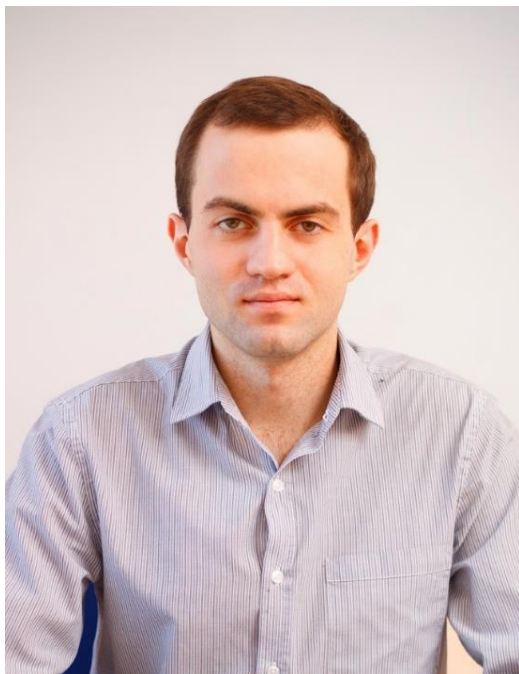
JavaScript Essential



Введение в JavaScript

JavaScript Essential

Автор курса



Дмитрий Охрименко
МСТ



MCID: 9210561

JavaScript Essential

После урока обязательно



Повторите этот урок в видео формате на [ITVDN.com](http://itvdn.com)



Проверьте как Вы усвоили данный материал на [TestProvider.com](http://testprovider.com)

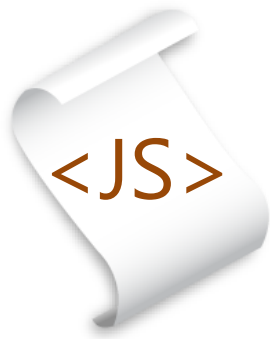
JavaScript Essential

Тема

Введение в JavaScript

Язык программирования JavaScript

The JavaScript Programming Language



Наиболее широко применяется в браузерах как язык сценариев для придания интерактивности web-страницам.

JavaScript – прототипно-ориентированный скриптовый язык программирования.

В соответствии со стандартом ECMA-262 язык официально называется ECMAScript.

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

Язык JavaScript по синтаксису напоминает C, C++ и Java программными конструкциями такими как: циклы, условные конструкции и операторы. Однако это подобие ограничивается синтаксической схожестью.

<http://www.ecma-international.org/publications/files/ECMA-ST/Ecma-262.pdf>

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.



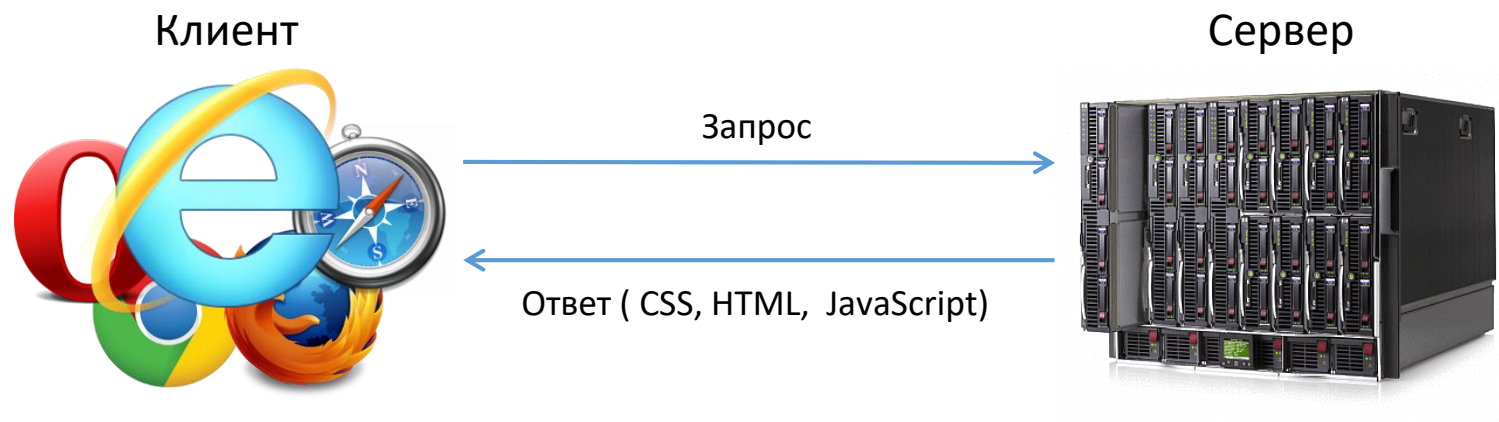
В апреле 1995 года начал работу в Netscape , где разработал язык программирования JavaScript (изначально названный «Mocha», затем переименован в «LiveScript») для браузера Netscape Navigator.

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

JavaScript



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

Клиентский JavaScript

Роль клиентского JavaScript

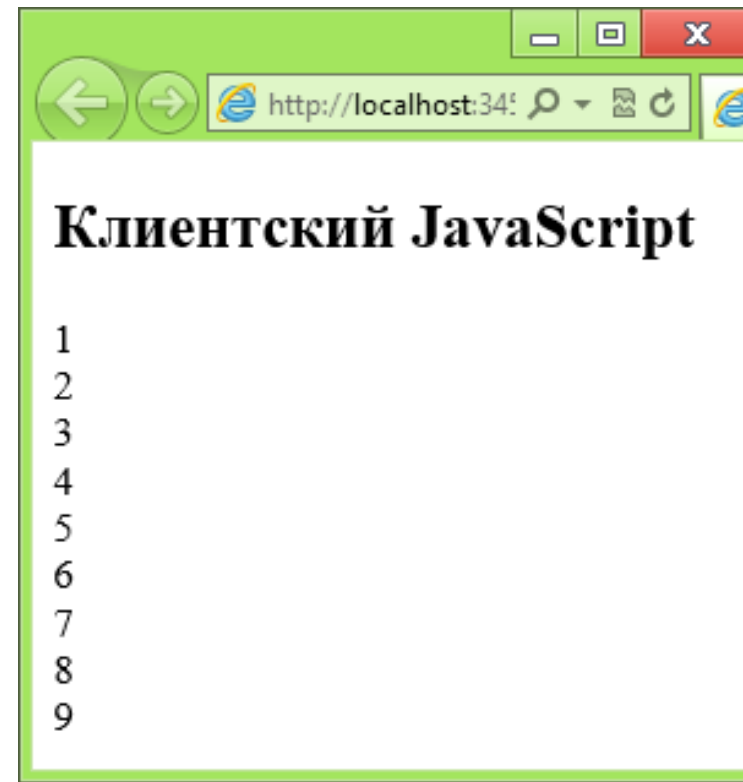
- Пользовательский интерфейс:
 - Создание визуальных эффектов (анимация).
 - Выполнение не сложных вычислений.
 - Проверка данных введенных пользователем.
 - Манипуляция данными введенными пользователем в формы.
 - Поиск по данным, встроенным в загружаемую страницу.
 - Сохранение данных в cookies.
-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

Клиентский JavaScript

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

Lexical structure

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

JavaScript

В JavaScript имеется набор зарезервированных слов

break	do	if	private	try
case	else	import	protected	typeof
catch	enum	in	public	var
class	export	instanceof	super	void
const	false	let	switch	while
continue	finally	new	this	with
default	for	null	throw	
delete	function	return	true	

Зарезервированные слова

ECMAScript

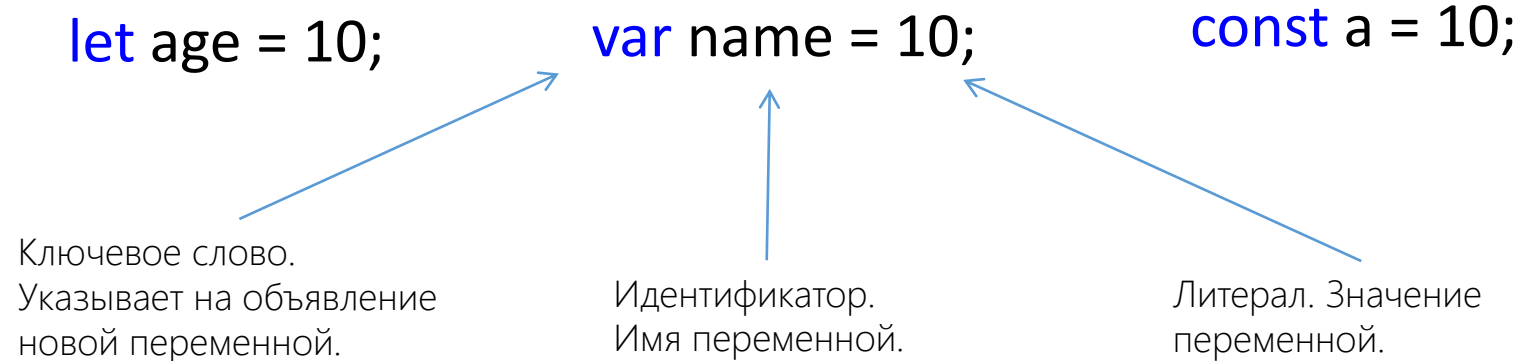
Зарезервированные ключевые слова в ECMAScript в данный момент в JavaScript не используются, но зарезервированы в качестве возможных будущих расширений языка.

abstract	double	implements	package	throws
boolean	extends	int	short	transient
byte	final	interface	static	volatile
char	float	long	synchronized	
debugger	goto	navite		

Переменные и типы данных

Создание переменных

Переменная – это символ или слово, обозначающее некоторое значение.



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

Идентификаторы

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.

Типы данных

Data Types

Элементарные типы

- Числа ([Number](#))
- Строки ([String](#))
- Логические значения ([Boolean](#))

Тривиальные типы

- [null](#) – отсутствие значения
- [undefined](#) – неопределенное значение

Составные типы

- Объект ([Object](#))
- Массив ([Array](#))

Специальный тип

- Функция ([function](#))

Объявление переменных

Data Types

```
let a = 10;    // тип Number  
let b = 0.8;   // тип Number
```

```
let c = "string"; // тип String  
let d = 'string'; // тип String
```

```
let e = true;   // тип Boolean  
let f = false;  // тип Boolean
```

```
let g = null;    // тип null  
let h = undefined; // тип undefined
```

```
let i = new Date()    // тип Object  
let j = new Object(); // тип Object
```

```
let k = [1,2,3]; // Массив  
let l = Array(); // Массив
```

```
let m = function () { }; // Функция
```


Операторы

Присваивание с операцией

Оператор	Пример	Эквивалент
+=	a += b	a = a + b
-=	a -= b	a = a - b
*=	a *= b	a = a * b
/=	a /= b	a = a / b
%=	a %= b	a = a % b
<<=	a <<= b	a = a << b
>>=	a >>= b	a = a >> b
>>>=	a >>>= b	a = a >>> b
&=	a &= b	a = a & b
=	a = b	a = a b
~=	a ~= b	a = a ~ b

Операторы

Оператор равенства и идентичности

Равенство (==)

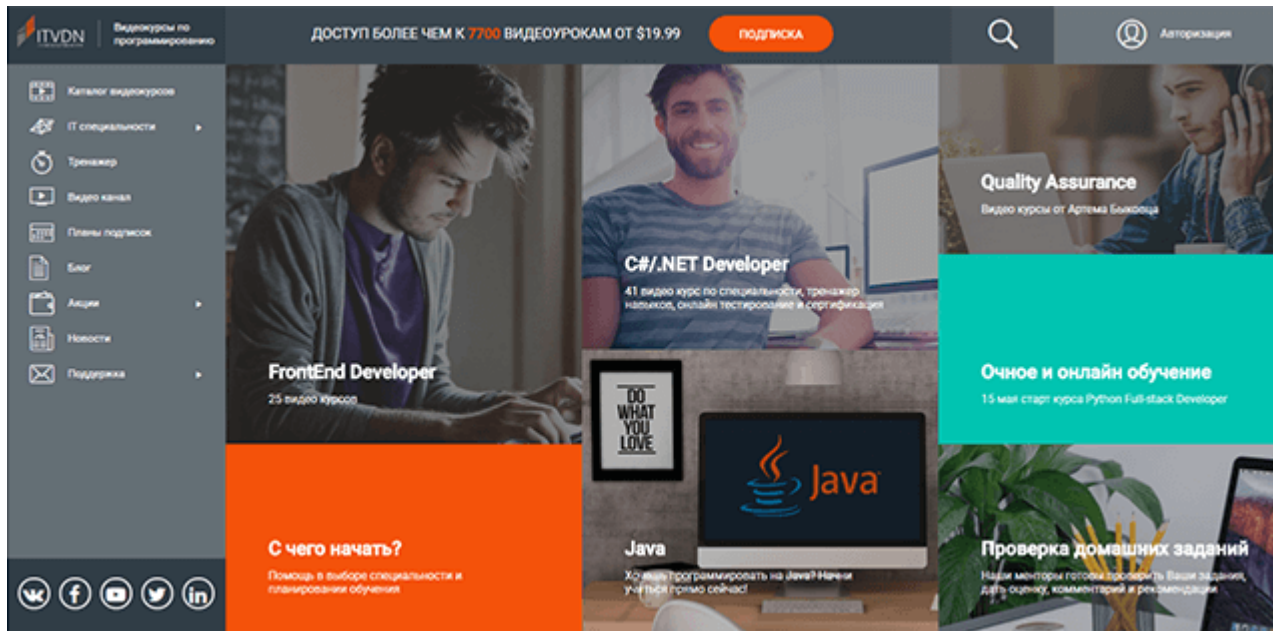
(1==1)	= true
('text'=='text')	= true
(null==undefined)	= true
(1=='1')	= true
(1==true)	= true
(0==false)	= true

Идентичность (===)

(1===1)	= true
('text'=== 'text')	= true
(null===undefined)	= false
(1=== '1')	= false
(1===true)	= false
(0===false)	= false

Смотрите наши уроки в видео формате

ITVDN.com



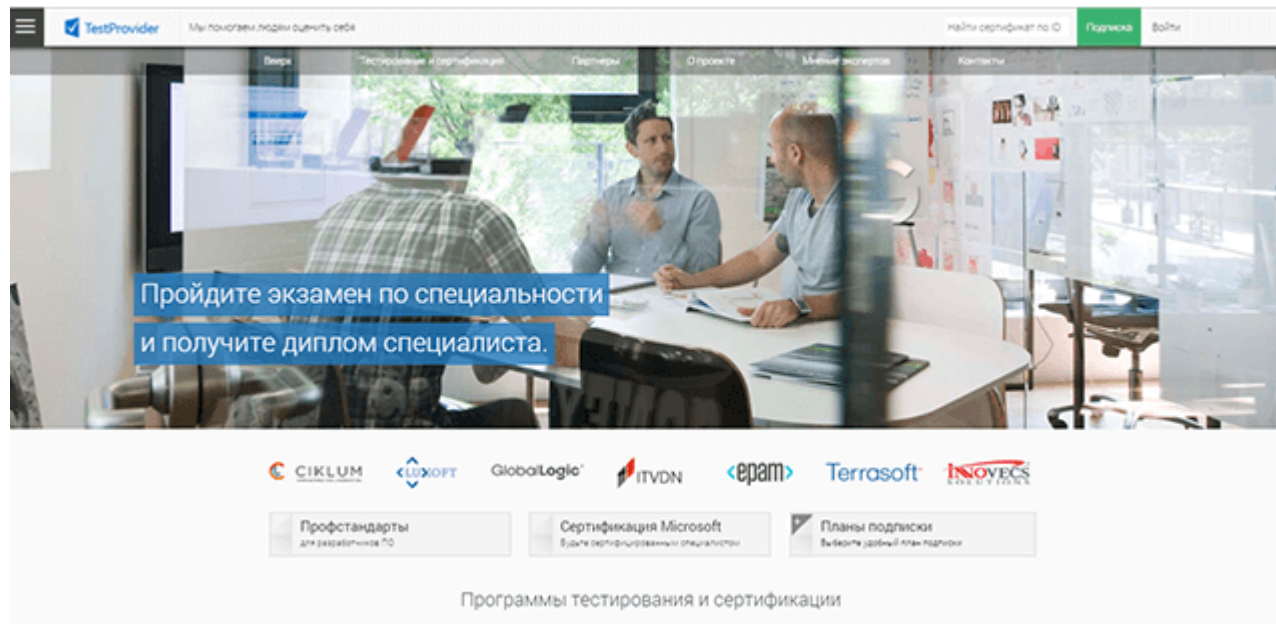
Посмотрите этот урок в видео формате на образовательном портале ITVDN.com для закрепления пройденного материала.

Курсы записаны сертифицированными тренерами, которые работают в учебном центре CyberBionic Systematics и другими высококвалифицированными разработчиками.



Проверка знаний

TestProvider.com



TestProvider – это online сервис проверки знаний по информационным технологиям. С его помощью Вы можете оценить Ваш уровень и выявить слабые места. Он будет полезен как в процессе изучения технологии, так и для общей оценки знаний IT специалиста.

После каждого урока проходите тестирование для проверки знаний на [TestProvider.com](https://testprovider.com)

Успешное прохождение финального тестирования позволит Вам получить соответствующий Сертификат.



JavaScript Essential

Q&A

Информационный видеосервис для разработчиков программного обеспечения

