

# JavaScript

Java Collections  
3 уровень, 1 лекция

ОТКРЫТА

— Привет, Амиго!

— Привет, Билаабо!

— Рад тебя видеть. Сегодня у нас небольшое, но очень познавательное занятие. Сегодня я расскажу тебе о языке JavaScript.



— Новый язык? Как интересно...

— JavaScript сейчас популярен благодаря интернету. Дело в том, что это единственный язык, который умеют исполнять все браузеры. Если ты хочешь добавить анимацию или некоторую логику в свою интернет-страницу, то ты можешь сделать это с помощью JavaScript.

— А правда, что JavaScript – самый популярный язык?

— Да, но тут правильнее было бы сказать, что он самый популярный «второй язык». Программисты на C++, Java, C#, PHP вынуждены писать небольшие скрипты на JavaScript, чтобы оживить свои интернет-страницы. В то же время, людей, которые пишут только на JavaScript гораздо меньше.

— А почему у него такое название – JavaScript? Звучит почти как Java.

— На самом деле сначала он назывался LiveScript, но когда Java начала набирать популярность, его переименовали в JavaScript.

Java и JavaScript — это два совершенно разных языка, не стоит их путать.

JavaScript, в отличие от Java, не имеет классов, не поддерживает статическую типизацию, многопоточность и много чего еще. **И если Java – это скорее большой набор строительных инструментов, то JavaScript – это швейцарский нож.** JavaScript создан для решения маленьких задач, а Java – для больших и очень больших.

Вот тебе несколько фактов о JavaScript:

## Факт первый – в JavaScript есть функции, но нет классов

Просто пишешь логику программы в нескольких функциях и все. Пример:

НАЧАТЬ ОБУЧЕНИЕ

JavaScript	Java
<pre>1 function min(a, b) 2 { 3   return a 4 }</pre>	<pre>1 public static int min(int a, int b) 2 { 3   return a 4 }</pre>

Новые функции объявляются с помощью конструкции «**function+имя**».

Еще пример:

JavaScript	Java
<pre>1 function min(a, b) 2 { 3   return a&lt;b ? a: b; 4 } 5 6 function main() 7 { 8   var s = 3; 9   var t = 5; 10  var min = min(s, t); 11 }</pre>	<pre>1 public static int min(int a, int b) 2 { 3   return a&lt;b ? a: b; 4 } 5 6 public static void main() 7 { 8   int s = 3; 9   int t = 5; 10  int min = min(s,t); 11 }</pre>

Факт второй – в JavaScript есть переменные, но у них нет типов

JavaScript – это язык с динамической типизацией. Это значит, что у переменных, по факту, нет типов. Любой переменной можно присвоить значение любого типа (у значений типы есть). Пример:

JavaScript	Java
<pre>1 function main() 2 { 3   var s = "Bender"; 4   var k = 1; 5   var n = s.length; 6 }</pre>	<pre>1 public static void main() 2 { 3   String s ="Bender"; 4   int k = 1; 5   int n = s.length(); 6 }</pre>

Но динамическая типизация увеличивает риск ошибок во время работы программы:

JavaScript	Java
<pre>1 function main() 2 { 3   var s = "Bender"; 4   var k = 1; 5   var n = k.length; 6 }</pre>	<pre>1 public static void main() 2 { 3   String s ="Bender"; 4   int k = 1; 5   int n = k.length(); 6 }</pre>

В примере выше, мы подставили вместо s (строки), переменную k (число). В случае с Java ошибка будет обнаружена на этапе компиляции, в случае с JavaScript – позже: уже во время исполнения этого кода.

В JavaScript очень мало строгих правил и очень много анархии.

Можно объявить функцию с 5 аргументами, а вызвать с двумя – остальные будут равны null. Можно объявить функцию с двумя аргументами, а при вызове передать пять. Три просто отбросятся. Контроль за ошибками, опечатками, изменениями – минимальный.

### Факт третий – в JavaScript есть if, for, while

В JavaScript есть if, for, while и это – хорошая новость. Смотрим примеры:

JavaScript	Java
<pre>1 function main() 2 { 3   var s = "Bender"; 4 5   var result = ""; 6 7   for(var i=0;i&lt;s.length;i++) 8   { 9     result += s[i]+ " "; 10  } 11  if(result.length&gt;10) 12  { 13    alert (result); 14  } 15  else 16  { 17    while(result.length &lt;= 10) 18    { 19      result += " "; 20    } 21    alert(result); 22  } 23 }</pre>	<pre>1 public static void main() 2 { 3   String s = "Bender"; 4   char[] s2 = s.toCharArray(); 5   String result = ""; 6 7   for(int i=0;i&lt;s.length();i++) 8   { 9     result += s2[i]+ " "; 10  } 11  if(result.length()&gt;10) 12  { 13    System.out.println(result); 14  } 15  else 16  { 17    while (result.length() &lt;= 10) 18    { 19      result += " "; 20    } 21    System.out.println(result); 22  } 23 }</pre>

— Довольно похоже. Думаю, я смог бы разобраться, как работает код, написанный на JavaScript.

— Оптимизм – это хорошо.

### Факт четвёртый – в JavaScript есть try-catch-finally

В JavaScript есть исключения (**Error**) и это хорошо. **checked** исключений нет, только **unchecked** – аналог **RuntimeException**. **try-catch-finally** работает так же, как и в Java. Пример:

JavaScript	Java
<pre>1 function main() 2 { 3   try 4   { 5     var s = null; 6     var n = s.length; 7   } 8   catch(e)</pre>	<pre>1 public static void main() 2 { 3   try 4   { 5     String s = null; 6     int n = s.length(); 7   } 8   catch(Exception e)</pre>

НАЧАТЬ ОБУЧЕНИЕ

11	}
12	}

11	}
12	}

При попытке узнать длину строки будет выкинуто исключение, т.к. переменная ссылается на null.

## Факт пятый – в JavaScript есть массивы

Хорошая новость. В JavaScript есть массивы. Плохая – там нет ни списков, ни коллекций.

Еще одна хорошая новость – массивы могут динамически растягиваться, при добавлении новых элементов и уменьшаться при их удалении. Больше похоже на гибрид массива и списка.

Пример:

JavaScript	
1	function main()
2	{
3	var m = [1,3,18, 45, 'c', "roma", null];
4	alert(m.length); //7
5	
6	m.push("end");
7	alert(m.length); //8
8	
9	for (var i=0;i<m.length;i++)
10	{
11	alert(m[i]);
12	}
13	}

Java	
1	public static void main()
2	{
3	List m = Arrays.asList(1,3,18, 45,'c', "roma");
4	System.out.println(m.size());//7
5	
6	m.add("end");
7	System.out.println(m.size());//8
8	
9	for (int i=0;i<m.size();i++)
10	{
11	System.out.println(m.get(i));
12	}
13	}

— А что это за квадратные скобки при объявлении массива?

— Это и есть «объявление массива». Чтобы объявить массив надо написать квадратные скобки, а между ними перечислить элементы массива. Пустой массив объявляется просто парой скобок.

Пример
var m = [];

## Факт шестой – в JavaScript есть объекты

В JavaScript есть объекты. На самом деле все в JavaScript является объектом, примитивные типы тоже. Каждый объект представлен в виде набора пар «ключ-значение». Грубо говоря, каждый объект JavaScript – это аналог HashMap в Java. Пример объявления объекта:

JavaScript	
1	function main()
2	{
3	var m = {
4	first_name : "Bill",
5	last_name: "Gates",
6	age: 54,
7	weight: 67,
8	children: ["Emma", "Catrin"],

Java	
1	public static void main()
2	{
3	HashMap m = new HashMap();
4	m.put("first_name", "Bill");
5	m.put("last_name", "Gates");
6	m.put("age", 54);
7	m.put("weight", 67);
8	

НАЧАТЬ ОБУЧЕНИЕ

11121314151617181920212223

last\_name: "Gates",  
age: 45,  
}  
};  
  
alert(m.first\_name); // Bill  
alert(m.age); // 54  
alert(m.wife.first\_name);// Melinda  
  
m.age = 45;  
m.age++;  
m["first\_name"] = "Stive";  
m["wife"] = null;

1112131415161718192021222324252627

HashMap wife = new HashMap();  
wife.put("first\_name", "Melinda");  
wife.put("last\_name", "Gates");  
wife.put("age", 45);  
m.put("wife", wife);  
  
System.out.println(m.get("first\_name"));  
System.out.println(m.get("age"));  
  
HashMap w = ((HashMap)m.get("wife"));  
System.out.println(w.get("first\_name"));  
  
m.put("age", 45);  
m.put("age", ((Integer)m.get("age"))+1);  
m.put("first\_name", "Stive");  
m.put("wife", null);

Чтобы создать новый объект достаточно написать две фигурные скобки «{}».

Внутри скобок можно указать данные объекта в формате «ключ, двоеточие, значение, запятая».

К полям объекта можно обращаться двумя способами:

Эквивалентные записи	
1	m.age = 45;
1	m[“age”] = 45;

Если указанного поля нет, оно создается.

Что-то мой газовый пузырь переполнился, думаю надо сделать перерыв.

## Факт седьмой — JavaScript создан, чтобы работать внутри web-страниц.

[< Предыдущая лекция](#)

[Следующая лекция >](#)

−

+105

+

Комментарии (87)

популярныеновыестарые

JavaCoder

Введите текст комментария

Алексей СУровень 30, Россия

24 июля, 14:41

...

Смотрю на джава скрипт( не первый раз, я уже делал одно приложение на нем) и понимаю насколько он проще в написании, но ненадежнее.

Ответить

−

0

+

НАЧАТЬ ОБУЧЕНИЕ

Бесплатные курсы по JS:  
[https://ru.hexlet.io/courses/introduction\\_to\\_programming\\_](https://ru.hexlet.io/courses/introduction_to_programming_) // введение в программирование на JS  
<https://ru.hexlet.io/courses/js-basics> // основы JS

Ответить

0

const

Уровень 38

17 мая, 02:56

Оптимизм - это хорошо )

Ответить

+2

Джу

Уровень 35

9 декабря 2021, 20:35

Здравствуй, здравствуй, нежный ДжаваСкрипт

Ответить

0

Jalkon

Уровень 37, Москва

30 ноября 2021, 01:12

В JavaScript есть классы

Ответить

0

2DaMoon

Уровень 41

18 октября 2021, 22:08

Есть одна поговорка: самый востребованный навык для java программиста - знание javascript

Ответить

+1

Лиза Воренувкина

Уровень 43, Кривой Рог, Ukraine

24 сентября 2021, 18:33

Этот язык не смотря на простоту очень сложный .

Ответить

+4

Edil Kalmamatov

Уровень 35

14 сентября 2021, 20:05

"Факт седьмой – JavaScript создан, чтобы работать внутри web-страниц." - эту фразу Билаабо говорил уже из комнаты для освобождения от газов?

Ответить

+5

Лиза Воренувкина

Уровень 43, Кривой Рог, Ukraine

24 сентября 2021, 18:33

Красиво сказано ,,,,

Ответить

0

Лизунов Сергей

Уровень 41, Пенза, Россия

26 августа 2021, 17:08

Хм, чем-то Python напоминает. Только там, например, не надо писать var, а просто m = [1,3,18, 45, 'с', "roma", null]. function = def, только фигурных скобок не надо. Динамическая типизация. Интересно, может и пригодится мой небольшой опыт изучения Python.

Ответить

0

Darth Nihilus

Разработчик спокойствия в Rage&Flame Industrie

21 мая 2021, 01:39

эти лекции точно нужно актуализировать)

Ответить

+8

Показать еще комментарии

ОБУЧЕНИЕ

- Курсы программирования
- Курс Java
- Помощь по задачам
- Подписки
- Задачи-игры

СООБЩЕСТВО

- Пользователи
- Статьи
- Форум
- Чат
- Истории успеха
- Активности

КОМПАНИЯ

- О нас
- Контакты
- Отзывы
- FAQ
- Поддержка

НАЧАТЬ ОБУЧЕНИЕ

JavaRush — это интерактивный онлайн-курс по изучению Java-программирования с нуля. Он содержит 1200 практических задач с проверкой решения в один клик, необходимый минимум теории по основам Java и мотивирующие фишки, которые помогут пройти курс до конца: игры, опросы, интересные проекты и статьи об эффективном обучении и карьере Java-девелопера.

ПОДПИСЫВАЙТЕСЬ

ЯЗЫК ИНТЕРФЕЙСА

Русский

▼

