





Знакомство с программированием. JavaScript Начало



Елена Иванова
@liveldi90
Front-end developer
Artec3D

Содержание

-  Что такое JavaScript и его возможности.
-  Как браузеры обеспечивают безопасность страниц.
-  Подключаем скрипты к странице. Порядок исполнения.
-  Основы синтаксиса: переменные, типы данных, операторы, циклы, функции.

Программа

Последовательность инструкций,
определяющих процедуру решения конкретной
задачи.



Что такое JavaScript и его возможности

Что такое JavaScript?

JavaScript изначально создавался для того, чтобы сделать web-странички «живыми».

Программы на этом языке называются **скриптами**.

В браузере они подключаются напрямую к HTML и, как только загружается страничка – тут же выполняются.



Немного истории

- JavaScript появился на свет в 1995 году.
- Создатель Брендан Эйх.
- JavaScript был написан за 10 дней!
- Язык Java был популярен в то время, поэтому было придумано название JavaScript.
- Последние версии:
ECMAScript 5 и
ECMAScript 6

*JavaScript — это торговая марка,
принадлежит Oracle.
ECMAScript — это спецификация не
зависит от владельцев.*



Преимущества для изучения

- + Программы на JavaScript — это просто текст.
- + Не нужно настраивать никакого дополнительного окружения.
- + Небольшое количество кода = видимый результат.

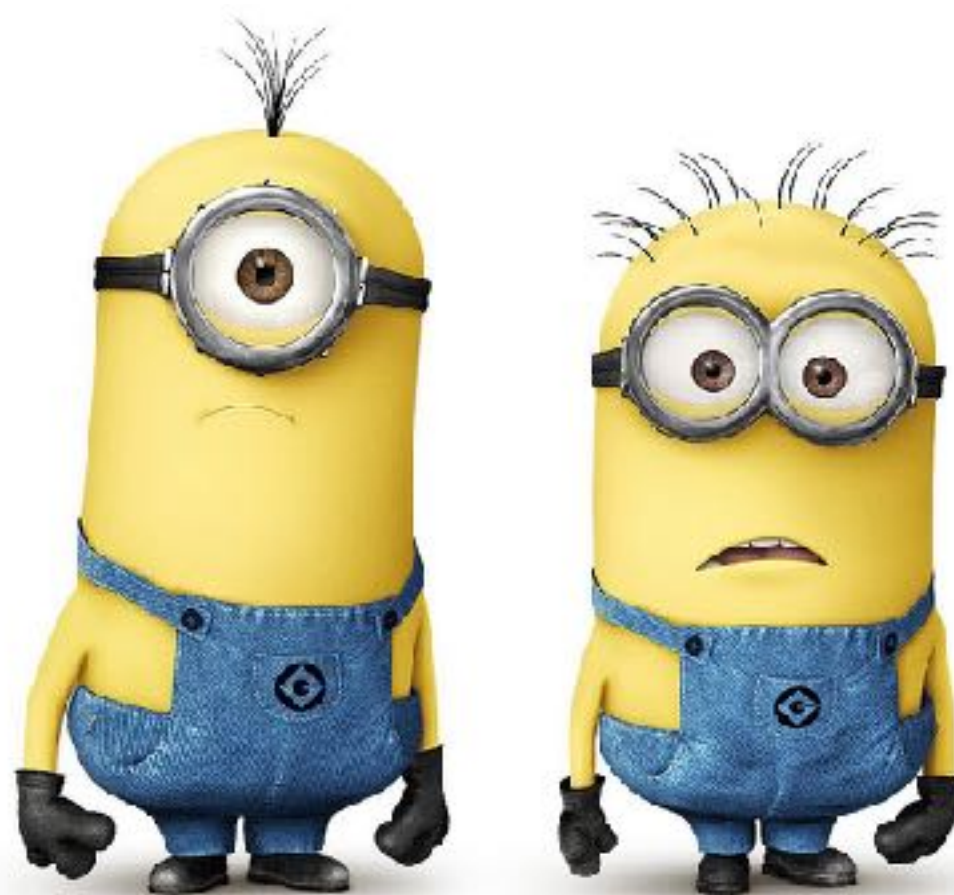


Что умеет JavaScript?

- Создавать новые HTML-теги, удалять существующие, менять стили элементов, прятать, показывать элементы и т.п.
- Реагировать на действия посетителя, обрабатывать клики мыши, перемещения курсора, нажатия на клавиатуру и т.п.
- ...и многое, многое другое!

Как браузеры
обеспечивают
безопасность
страниц

Может ли JavaScript код навредить нашему компьютеру?



Браузер накладывает на его исполнение
некоторые **ограничения**, которые **не**
позволяют злоумышленнику получить
личные данные или как-то навредить
компьютеру!



Что НЕ умеет JavaScript?

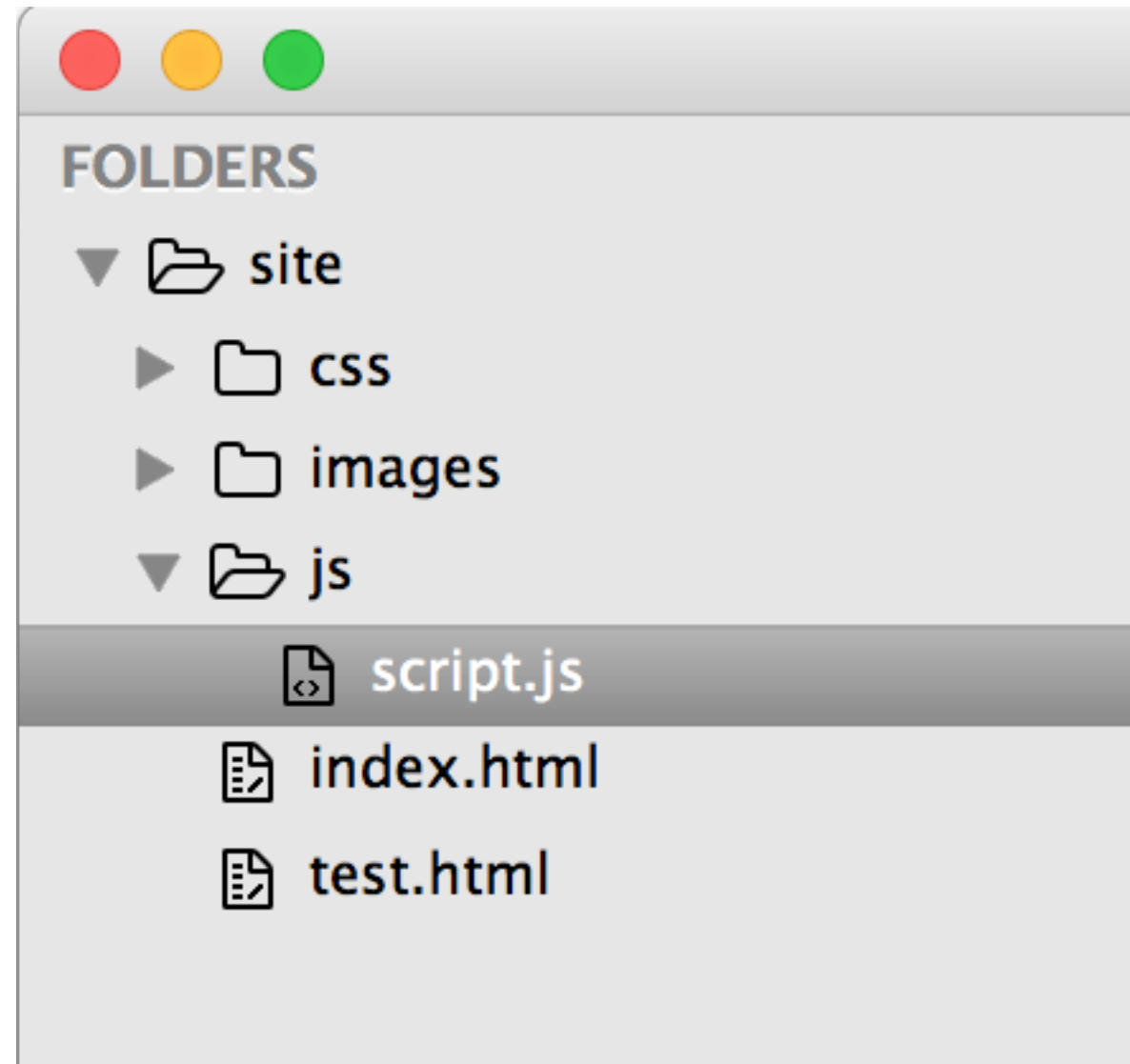
- Читать/записывать произвольные файлы на жесткий диск, копировать их или вызывать программы.
- JavaScript, работающий в одной вкладке, не может общаться с другими вкладками и окнами.
- JavaScript можно легко посылать запросы на сервер, с которого пришла страница. Но на чужие сервера без дополнительных настроек — нет.



Подключаем
скрипты к странице.
Порядок исполнения

Подключение

Создаем файл js -> script.js



Подключение

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <title>Работа с javascript</title>
  </head>
  <body>
    ...

    <script src="/js/main.js" type="text/javascript"></script>
  </body>
</html>
```

Путь к файлу



Тип содержимого



ОСНОВЫ СИНТАКСИСА

Структура кода

// Комментарий

/*

Многострочный комментарий

*/

alert(1);

/*

В конце каждой строки, необходимо
ставить ;

*/

Переменные

— Переменная состоит из имени и выделенной области памяти, которая ему соответствует.

```
var message = 'Hello!';
```

(Сохранили строку в переменную)

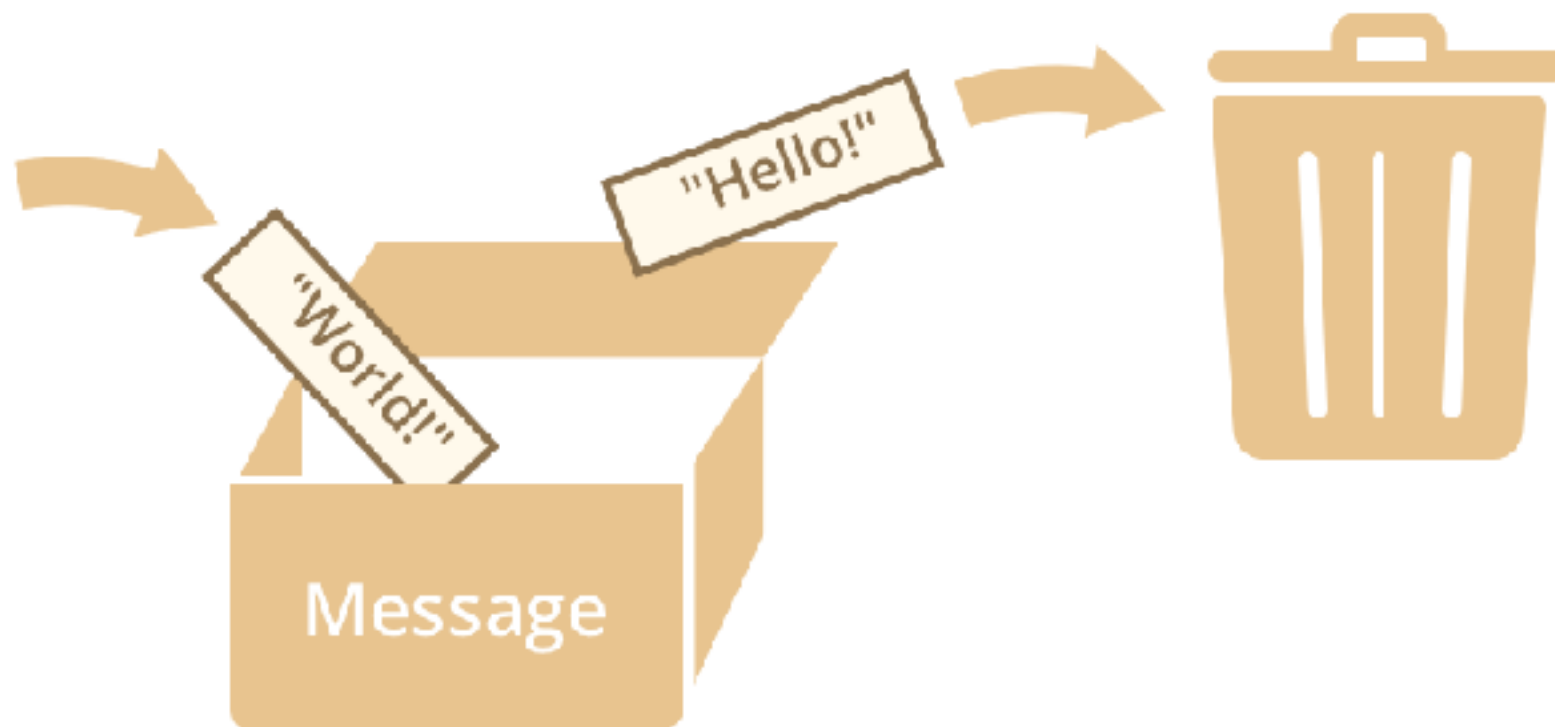


Переменные

```
var message = 'Hello!';
```

```
message = 'World!'; // заменили значение
```

```
alert( message );
```



Переменные

```
var message = 'Hello!';  
var copyMessage = message;  
  
// Копируем в новую переменную, значение из старой;
```

Имена переменных

На имя переменной в JavaScript наложены всего несколько ограничений:

- Имя может состоять из: букв, цифр, символов \$ и _.
- Первый символ не должен быть цифрой.
- Нельзя использовать зарезервированные слова: `var`, `class`, `return`, `export`
- Русские буквы использовать можно, но не желательно.
- Следует давать переменным “говорящие” имена

Важно новую переменную начинать с `var`!

Задание

- Объявите переменную: `admin`.
- Запишите в `admin` строку "Василий".
- Выведите `admin` (должно вывести «Василий»).



Типы данных

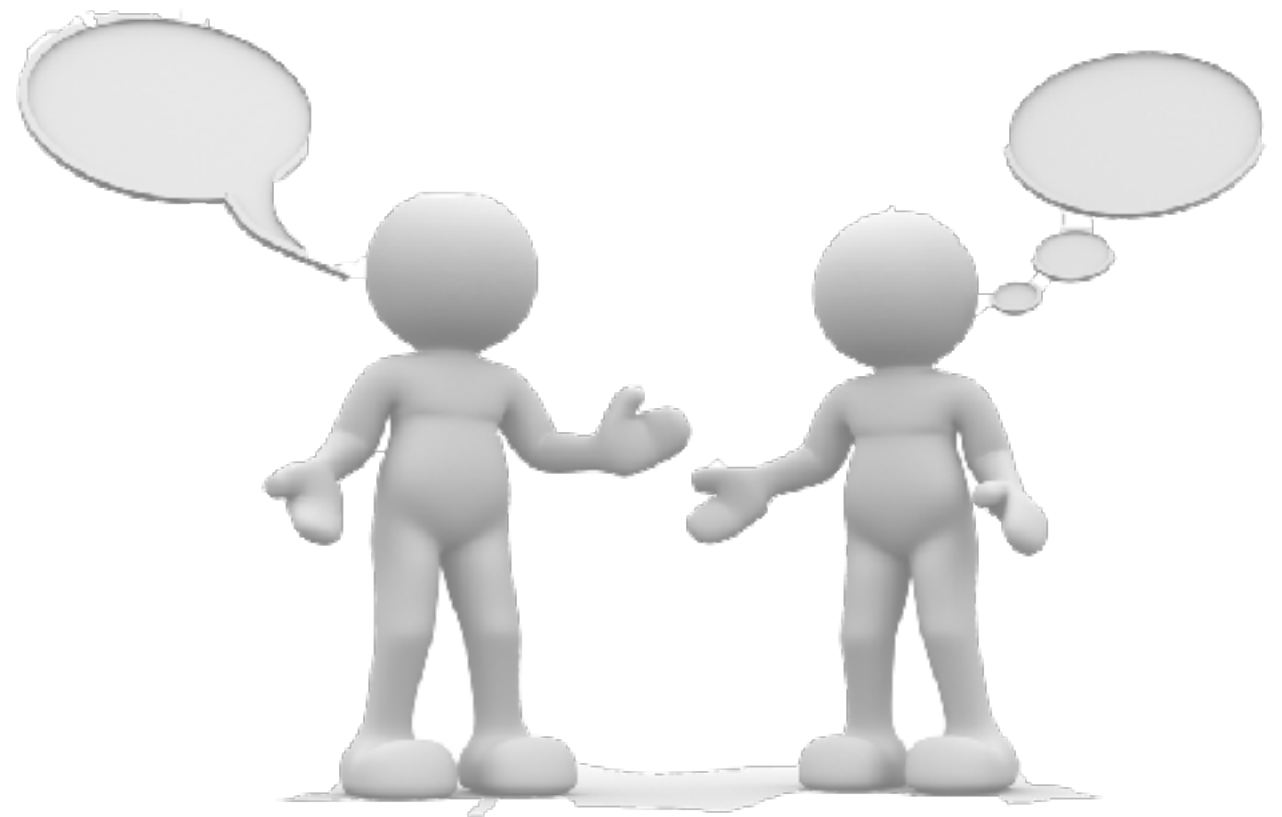
Существует 6 типов данных в JavaScript:

- Число «number»
- Строка «string»
- Булевый (логический) тип «boolean»
- Специальное значение «null»
- Специальное значение «undefined»
- Объекты «object»

Взаимодействие с пользователем

Взаимодействие с пользователем

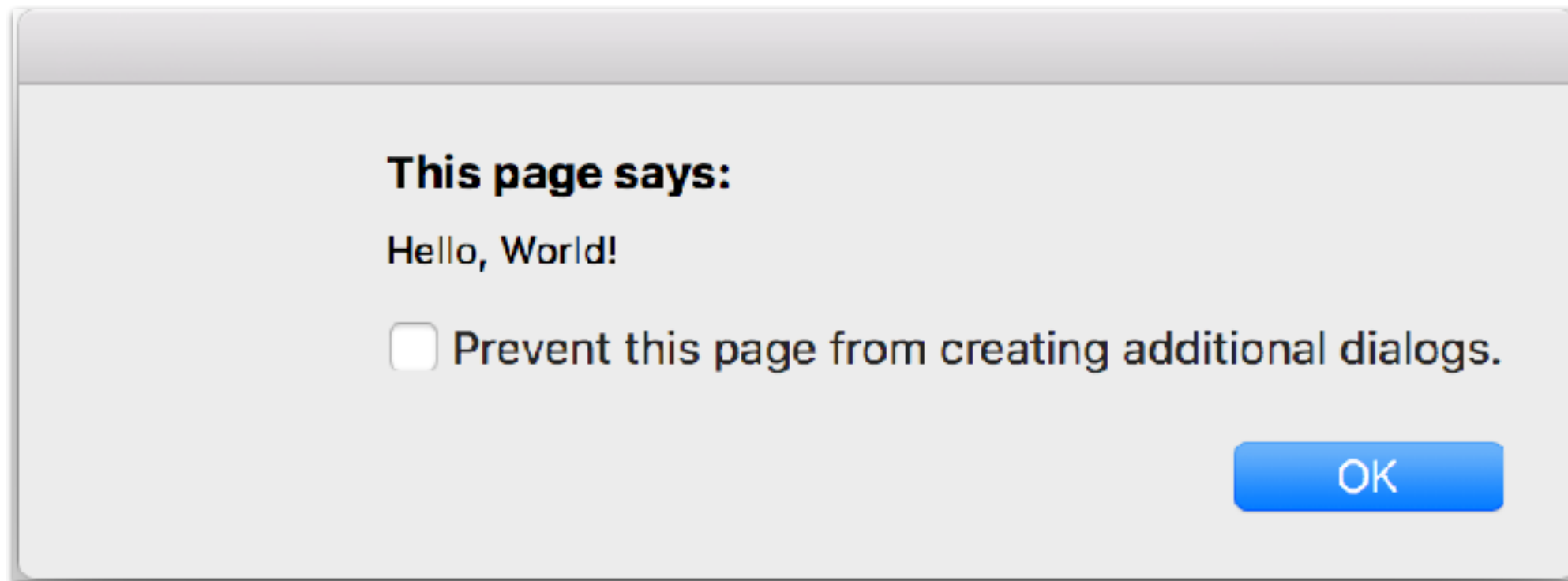
- alert
- prompt
- confirm



alert

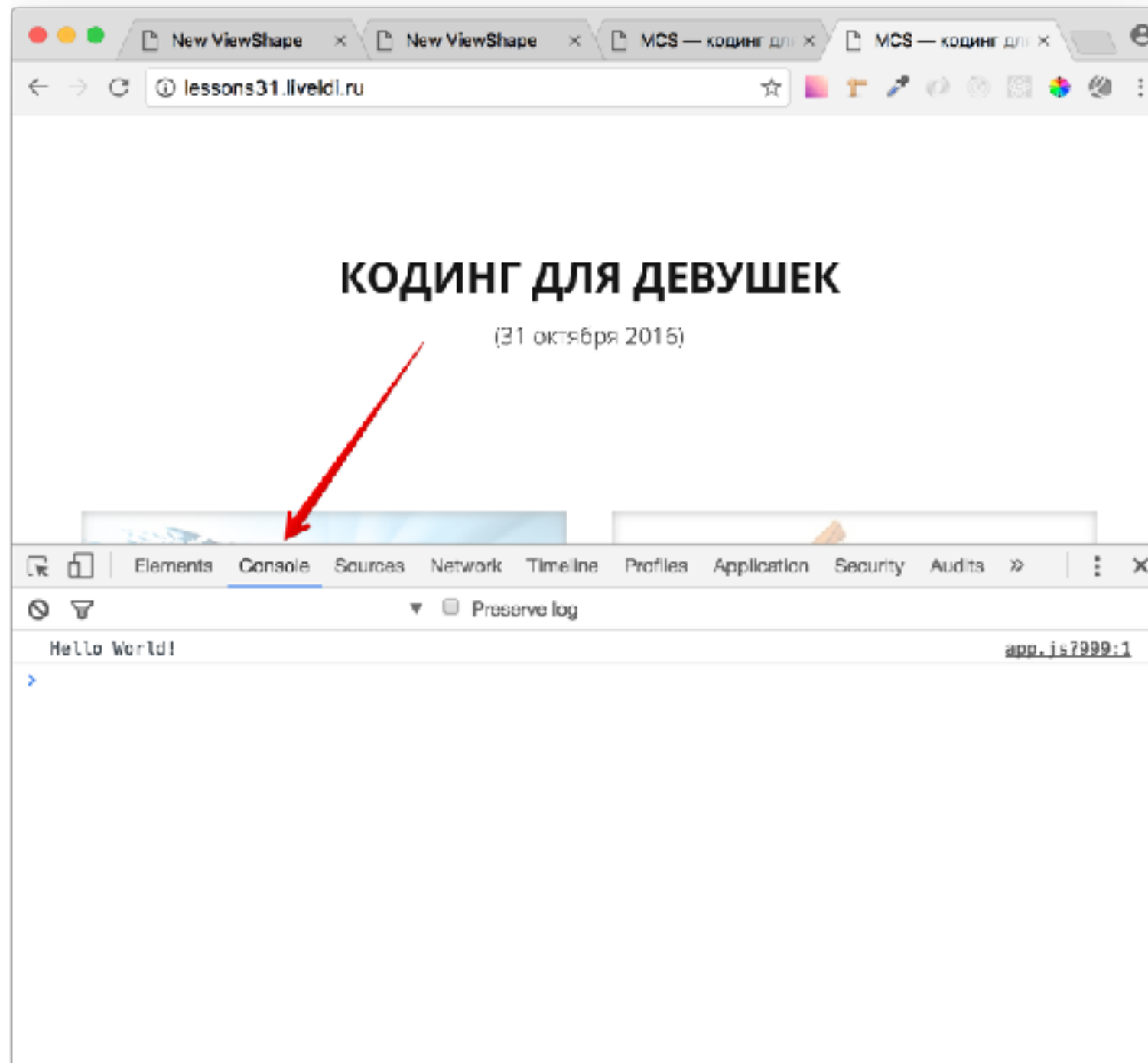
— alert выводит на экран окно с сообщением и приостанавливает выполнение скрипта, пока пользователь не нажмёт «OK».

```
alert( 'Hello, World' );
```



console.log

— console.log выводит в консоль разработчика, любое сообщение не блокируя экран.



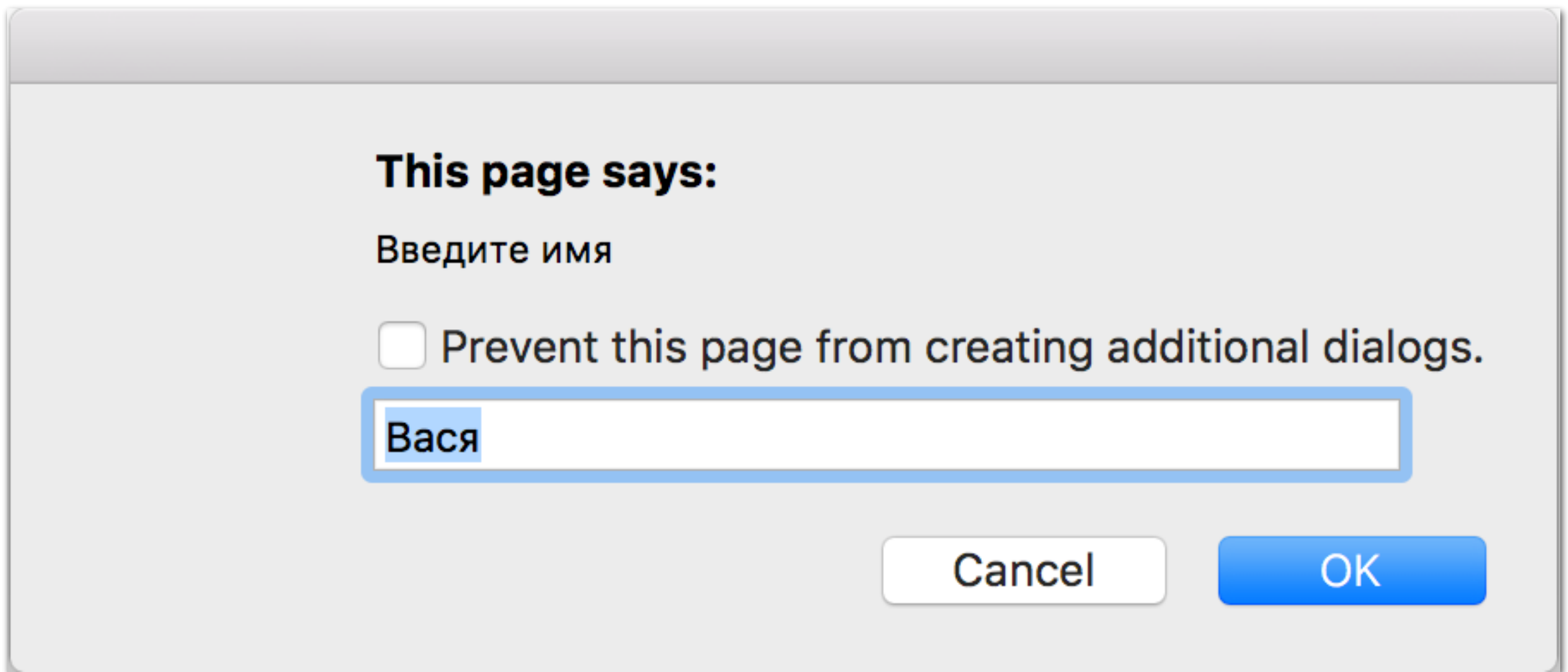
prompt

— ВЫВОДИТ МОДАЛЬНОЕ ОКНО С ЗАГОЛОВКОМ title, полем для ввода текста, заполненным строкой по умолчанию default и кнопками OK/CANCEL.

```
var result = prompt(title, default);
```

prompt

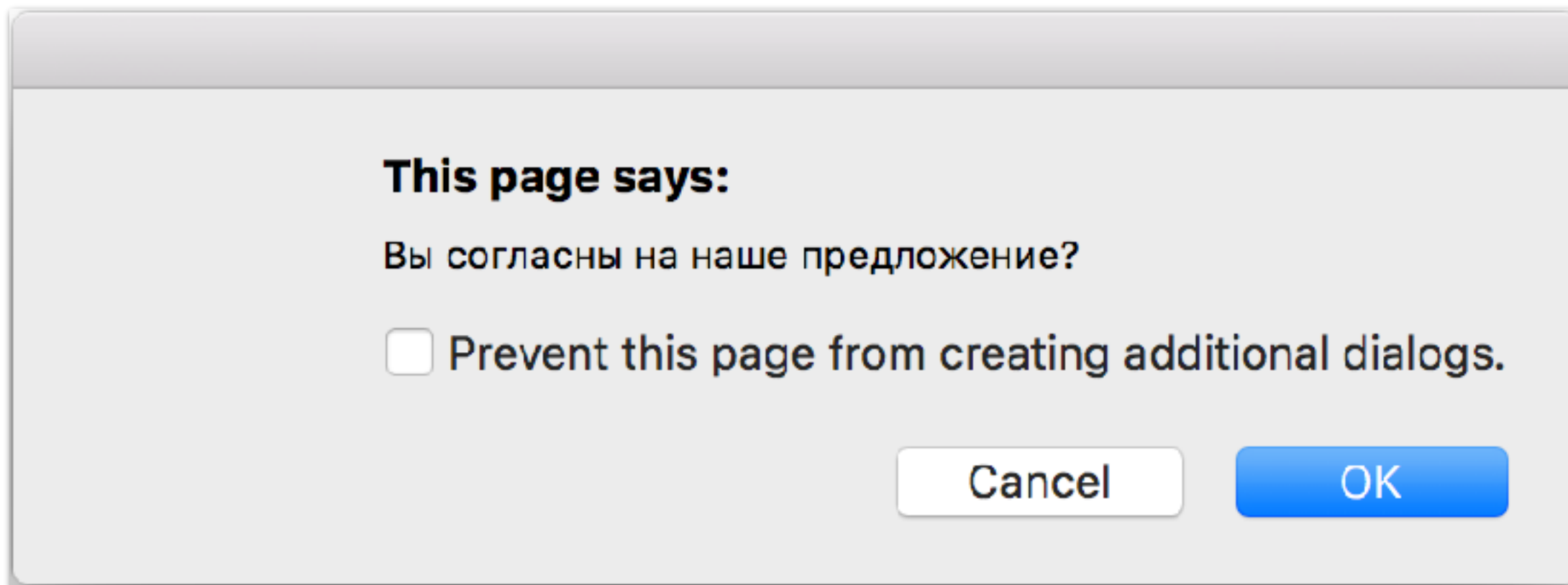
```
var result = prompt('Введите имя', 'Вася');  
  
alert(result); // Если нажали ok – то введеное имя.  
alert(result); // Если нажали cancel – false.
```



confirm

— Выводит окно с вопросом question с двумя кнопками: OK и CANCEL.

```
var question = confirm('Вы согласны на наше предложение?');  
  
alert(question); // Ok – true  
alert(question); // Cancel – false
```



Задание

- Объявите переменную: name.
- Выведите на странице окно с запросом у пользователя ввести свое имя.
- После как пользователь введет его, выведете его на экран.



Типы данных

Всего 6 типов данных

- Число
- Строка
- Булевы значения
- null
- undefined

Число «number»

```
var num = 123;  
var numFloat = 12.456;  
  
alert(1 / 0); // Infinity  
  
alert( "не число" * 2 ); // NaN, ошибка
```

Строка «string»

```
var str = "Мама мыла раму";  
str = 'Одинарные кавычки тоже подойдут';
```

- В JavaScript одинарные и двойные кавычки равноправны.
- Тип символ не существует, есть только строка.

Булевый (логический) тип «boolean»

У него всего два значения: true (истина) и false (ложь).

```
var checked = true;  
checked = false;
```

Специальное значение «null»

Состоит из единственного значения null.

```
var age = null;
```

Смысл: значение неизвестно.

Специальное значение «undefined»

Значение undefined, образует свой собственный тип, состоящий из одного этого значения. Оно имеет смысл «значение не присвоено».

```
var x;  
alert(x); // выведет "undefined"
```

Задание

- Объявите две переменные: `name` и `admin`
- Запишите в `name` строку "Василий".
- Скопируйте значение из `name` в `admin`.
- Выведите `admin` (должно вывести «Василий»).



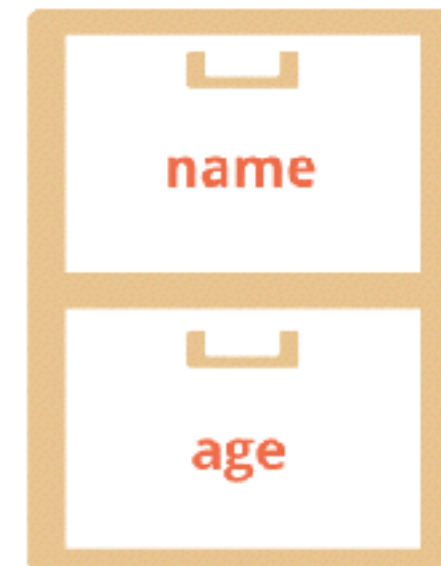
Объекты «object»

Объект может содержать в себе любые значения, которые называются свойствами объекта. Доступ к свойствам осуществляется по имени свойства (иногда говорят «по ключу»).

Объекты «object»

```
var person = {};
```

```
person.name = 'Вася';  
person.age = 25;
```



Можем хранить любой тип данных в ящиках (свойствах объекта).

Объекты «object»

```
var person = {  
  name: 'Вася',  
  age: 25  
};  
  
alert(person.name);  
alert(person['name']);
```

Задание

- Создайте объект `admin` с информацией о вас: имя, возраст, `email`, номер телефона.
- Используйте корректные имена свойств.
- Выведите через `console.log` каждое свойство.



ОПЕРАТОРЫ

Операторы

```
var x = 1 * 2 + 10 / 5 - 4;
```

```
var fullName = 'Вася ' + 'Петров'; // Вася Петров
```

Приоритет	Название	Обозначение
14	умножение	*
14	деление	/
13	сложение	+
13	вычитание	-
....
3	присваивание	=

Задание — калькулятор

- С помощью метода `prompt` попросите пользователя ввести одно число, запишите в переменную `number1`;
- Затем второе – `number2`;
- Умножьте эти числа и выведете в `alert` результат умножения:

Например:

Результат умножения = 3;



Инкремент/декремент: ++, --

- Инкремент ++ увеличивает на 1.

```
var i = 2;  
i++;      // более короткая запись для i = i + 1.  
alert(i); // 3
```

- Декремент -- уменьшает на 1.

```
var i = 2;  
i--;      // более короткая запись для i = i - 1.  
alert(i); // 1
```

Логические операторы

- Больше/меньше: $a > b$, $a < b$.
- Больше/меньше или равно: $a \geq b$, $a \leq b$.
- Равно $a == b$ ($a = b$ — это присваивание).
- «Не равно». В математике он пишется как \neq , в JavaScript – знак равенства с восклицательным знаком перед ним $!=$.

```
alert( 2 > 1 ); // true, верно  
alert( 2 == 1 ); // false, неверно  
alert( 2 != 1 ); // true, верно
```


Условные операторы

Иногда, в зависимости от условия, нужно выполнить различные действия. Для этого используется оператор `if`.

```
var year = prompt('В каком году появился JavaScript', '');  
  
if (year != 1995) {  
    alert('А вот и неправильно!');  
}
```

Можно использовать `else`, для дополнительного вывода, если `if` не сработал

```
if (year != 1995) {  
    alert('А вот и неправильно!');  
} else {  
    alert('Вы супер умный!');  
}
```

Задание — вход пользователя

Вопросы задаются через prompt



Блок-схема программы

ЦИКЛЫ

Циклы

— Для многократного повторения одного участка кода – предусмотрены циклы.

```
var i = 0;  
while (i < 3) {  
    alert( i );  
    i++;  
}
```

Задание

Какое последнее значение выведет этот код? Почему?

```
var i = 3;  
  
while (i != 0) {  
    i--;  
    alert(i);  
}
```



Циклы

— Для многократного повторения одного участка кода – предусмотрены циклы.

```
var i = 0;  
while (i < 3) {  
    alert( i );  
    i++;  
}
```

Циклы

— Более короткий вариант цикла

```
for (var i = 0; i < 10; i++) {  
    console.log(i);  
}
```

Циклы

— Цикл для перебора свойств объекта

```
var person = {  
    name: "Катя",  
    age: "15"  
}  
  
for (var key in person) {  
    console.log(person[key]);  
}
```


Задание

- Создайте объект `admin` с информацией о вас: имя, возраст, `email`, номер телефона.
- Используйте корректные имена свойств.
- Выведите каждое свойство используя цикл (`for in`) и `console.log`



ФУНКЦИИ

Функции

— Чтобы не повторять один и тот же код во многих местах, придуманы функции. Функции являются основными «строительными блоками» программы.

```
function showMessage() {  
    alert( 'Привет всем присутствующим!' );  
}
```

```
showMessage();  
showMessage();
```

Функции

— передаем любые переменные в функцию

```
function sayHi(name) {  
    alert( 'Привет, ' + name + '!' );  
}
```

```
sayHi( 'Леша' ); // Привет, Леша!
```

```
sayHi( 'Лена' ); // Привет, Лена!
```

Функции

— Используем переменные извне, объявляем внутренние.

```
var userName = 'Леша';

function showMessage() {
    var message = 'Привет, я ' + userName;
    alert(message);
}

showMessage(); // Привет, я Леша
```

Функции

— В функции можно возвращать значение и присваивать его переменной.

```
function sum(x, y) {  
    return x + y;  
}  
  
var sumNumbers = sum(5, 4);  
alert(sumNumbers);
```

Задание — калькулятор функции

- Создайте функцию `calculator`, на вход которой передаются два числа, как параметры и третьим параметром, то что необходимо с ними сделать:
 - `'multiple'` (умножить),
 - `'add'` (сложить),
 - `'divide'` (разделить),
 - `'subtract'` (вычесть)
- Внутри функции, используя оператор `if`, верните нужный результат.
- Присвойте результат переменной и выведите ее через `alert`.

Задание

- Напишите функцию `min(a, b)`, которая возвращает меньшее из чисел `a` и `b`.

Пример вызовов:

```
var result = min(2, 5);  
alert(result); // 2
```



Функции

— В функции можно возвращать значение и присваивать его переменной.

```
function sum(x, y) {  
    return x + y;  
}  
  
var sumNumbers = sum(5, 4);  
alert(sumNumbers);
```

МАССИВЫ

Массивы

— способ хранения упорядоченных данных.

```
var arr = [1, 5, 6, 7, 8];  
var fullArr = ['string', true, 6, { name: 'Валя' }];
```

```
arr[0] // 1  
arr[1] // 5  
arr[2] // 6  
arr[3] // 7  
arr[4] // 8
```

Перебор значений массива

— у массивов есть метод `.length`, который возвращает длину массива: `arr.length` // = кол-во элементов

```
var arr = [5, 7, 2, 4];  
  
for (var i = 0; i < arr.length; i++) {  
    console.log(arr[i])  
}
```

Задание

- Напишите функцию `getRandom` принимающее на вход минимальное и максимальное значения `min` и `max`, и возвращающее случайное число в этом диапазоне
- Используйте формулу:
$$x * (max - min) + min,$$
где x — случайное число от 0 до 1



Задание

- Создайте массив с различными значениями:
`var food = ['гренки', 'картошка', 'блинчики', 'йогурт'];`
- Используя функцию `getRandom`, достаньте случайное значение из этого массива.
- Сохраните значение в переменную и выведите на экран.



ДАТА & ВРЕМЯ

Просмотр

```
var nowDate = new Date(); // Создание текущей даты.  
console.log(nowDate); // Mon Nov 21 2016 15:58:37 GMT+0300 (MSK)  
  
nowDate.getFullYear() // Получить полный год (4 цифры)  
  
nowDate.getMonth() // Получить месяц (цифра от 0 до 11)  
  
nowDate.getDate() // Получить число месяца от 1 до 31
```

```
nowDate.getHours(),  
nowDate.getMinutes(),  
nowDate.getSeconds(),  
nowDate.getMilliseconds()
```


Задание

Необходимо вывести текущую дату в формате **21 нояб. 2016 г.**, используя объект `data`.

Подсказка: создайте массив со всеми месяцами и используйте его для получения текстового значения месяца.



МАТЕМАТИЧЕСКИЕ ФУНКЦИИ

Список математических функций

`Math.sqrt(x)` // – корень числа x

`Math.pow(base, exponent)` // – число (base) в степени (exponent)

`Math.max(1,3,4,1)` // – возвращает максимальное число (4)

`Math.min(5,6,1,7)` // – возвращает минимальное число (1)

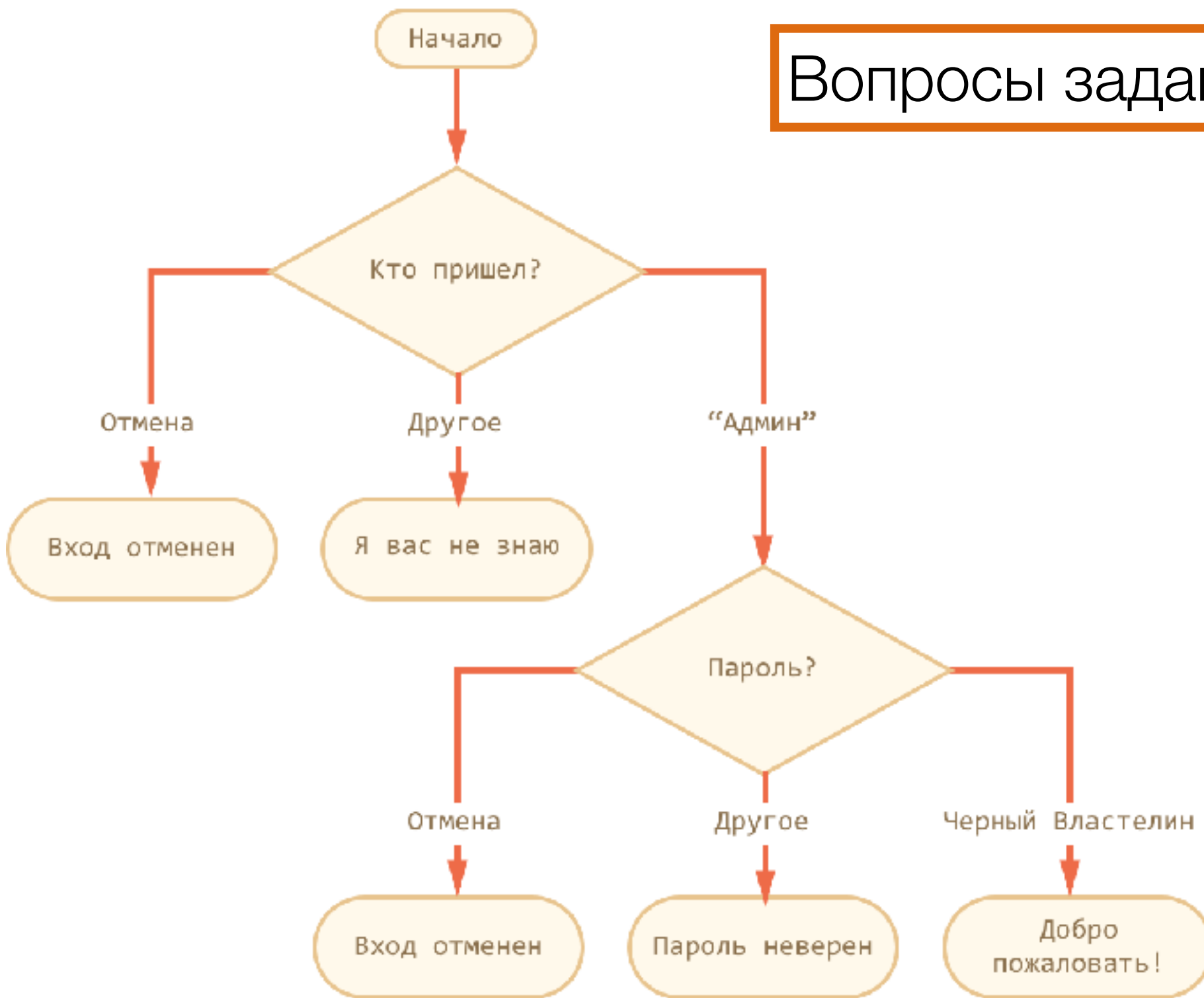
`Math.random()` // возвращает случайное число от 0 до единицы.

Полный список

Домашнее задание

Реализовать полностью блок-схему

Вопросы задаются через prompt



Блок-схема программы

Вопросы?



Автор курса: Елена Иванова
lessons7.liveldi.ru