

# ОСНОВЫ JavaScript

Урок 2





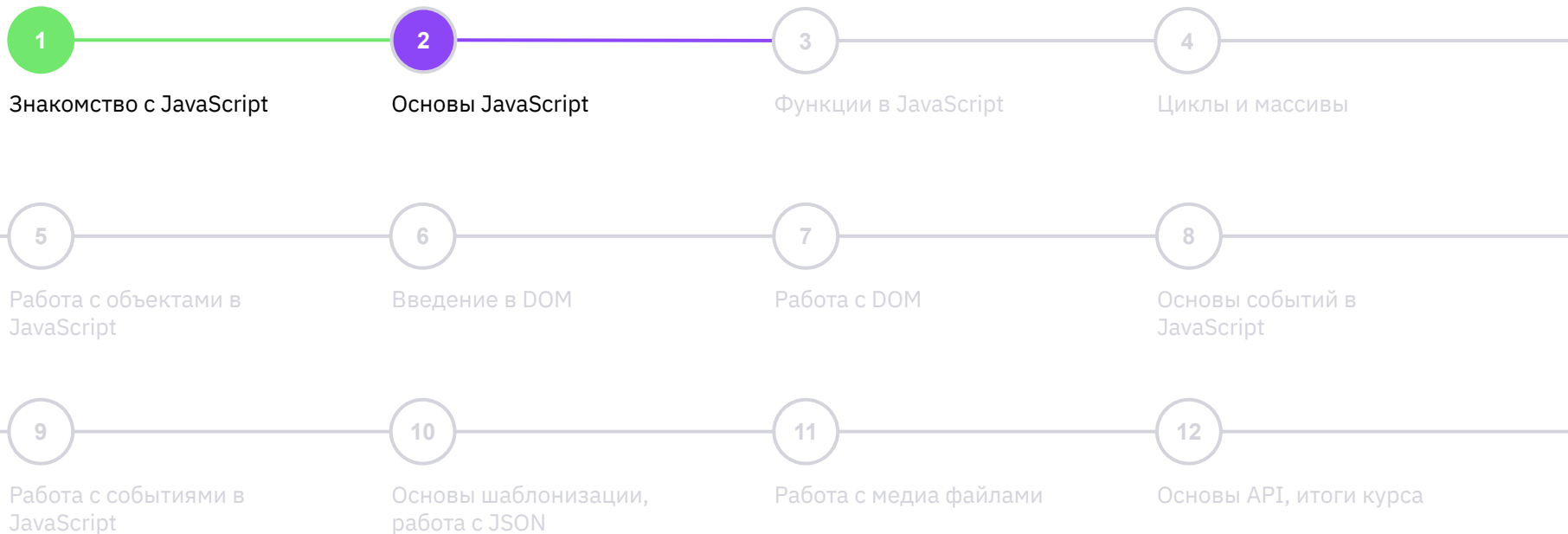
## Кадочников Алексей

Frontend-разработчик

- ☀ Веб-разработчик со стажем более 9 лет
- ☀ Преподаватель GeekBrains с 2015 года
- ☀ Автор курсов по Frontend на портале Geekbrains
- ☀ Работал в таких компаниях, как VK и Wizard-C










# План курса





## Что будет на уроке сегодня

-  Работа с операторами
-  Операторы сравнения
-  Преобразования типов данных
-  Приведение типов
-  Встроенные функции alert, prompt, confirm
-  Работа с отладчиком браузера
-  Логические и арифметические операторы



## Сравнение значений

Существует два типа сравнения значений:

- проверка равенства;
- проверка неравенства.





## Приведение типов



### **Явное**

когда мы сами меняем тип у значения



### **Неявное**

когда движок JavaScript делает это в необходимых местах



## Значения, приводящиеся к true (Truthy) и false (Falsy)

1

### false

- "" (пустая строка);
- 0, -0, NaN (0 и NaN);
- null, undefined;
- false.

2

### true

- "Hello" (непустые строки);
- 42;
- true;
- [ ], [ 1, "2", 3 ] (массивы);
- { }, { a: 42 } (объекты);
- function foo() { .. }  
(функции).



## Равенство

Есть четыре оператора проверки на равенство:

- `==` (нестрогое равенство)
- `===` (строгое равенство)
- `!=` (нестрогое неравенство)
- `!==` (строгое неравенство)

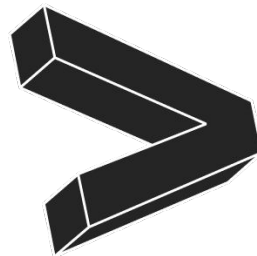
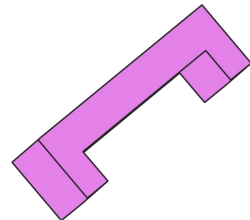
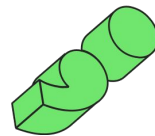




## Сравнения

Обычно их используют для сравнения чисел или строк.

- $<$
- $>$
- $<=$
- $>=$





# Разминка

## Задание

Что выведет консоль браузера?

→ `1 === "1"`

→ `1 == false`

→ `null !== 1`



# Унарные и бинарные операторы



## **Унарный оператор**

применяется к одному операнду



## **Бинарный оператор**

применяется к двум операндам



## У некоторых операторов есть свои особые названия

- **Инкремент** – увеличение операнда на установленный фиксированный шаг (как правило – единицу). Он же  $a++$  или  $a+1$ .
- **Декремент** – обратная инкременту операция.  $a--$  или  $a-1$ .
- **Конкатенация** – сложение строк. Обратной операции нет.





Частые примеры приведения  
типов и проверок





## Частые примеры приведения типов и проверок

- Проверка, есть ли значение
- Приведение строки к числу
- Приведение строки к числу
- Приведение числа в строку
- Приведение к булеву типу

## Все операторы в наглядной таблице

Оператор	Описание
<code>.[]()</code>	Доступ к полям, индексация массивов, вызовы функций и группировка выражений.
<code>++ -- ~ ! delete new typeof void</code>	Унарные операторы, тип возвращаемых данных, создание объектов, неопределённые значения.
<code>* / %</code>	Умножение, деление, деление по модулю.
<code>+ - +</code>	Сложение, вычитание, объединение строк.
<code>&lt;&lt; &gt;&gt; &gt;&gt;&gt;</code>	Сдвиг бит.
<code>&lt; &lt;= &gt; &gt;= instanceof</code>	Меньше, меньше или равно, больше, больше или равно, instanceof.
<code>== != === !==</code>	Равенство, неравенство, строгое равенство, строгое неравенство.
<code>&amp;</code>	Побитовое И.

## Все операторы в наглядной таблице (продолжение)

Оператор	Описание
<code>^</code>	Побитовое исключающее ИЛИ.
<code> </code>	Побитовое ИЛИ.
<code>&amp;&amp;</code>	Логическое И.
<code>  </code>	Логическое ИЛИ.
<code>?:</code>	Условный оператор.
<code>= OP=</code>	Присваивание, присваивание с операцией (например <code>+=</code> и <code>&amp;=</code> ).
<code>,</code>	Вычисление нескольких выражений.





## Методы alert, prompt, confirm

Первое что нужно помнить про эти методы

- Расположение определяется браузером.
- Внешний вид этих окон мы не можем поменять.





## alert

- Всплывающая подсказка, внешний вид зависит от вашего браузера. Достаточно старый способ отображения ошибок или подсказок на странице.



## prompt()

- **message** — это строка текста, которая показывается пользователю. Этот параметр является необязательным и может быть пропущен если в окне prompt ничего не показывать.
- **default** — это строка, содержащая значение по умолчанию, отображаемое в поле ввода текста. Это необязательный параметр. Обратите внимание, что в Internet Explorer 7 и 8, если вы не укажете этот параметр, строка undefined будет значением по умолчанию.



## confirm()

- message — опциональная (необязательная) строка, которая будет отображена в диалоговом окне.



## Работа с переменными и методами



### Шаблонные литералы

Звучит сложно, но на практике это популярные и достаточно простые в использовании ``` косые кавычки ```. Контент, который вы хотите отобразить внутри метода или в консоли вы можете поместить в косые кавычки.





## Принципы ветвления, визуализация, блок-схемы

- В программном коде, как и в жизни, множество решений зависят от внешних факторов. И зависимость эта выражается в вербальном виде «Если случится событие А, то я выполню действие Б». Именно по такому принципу начинает строиться ветвление во всех языках программирования.



## Операторы if, if-else



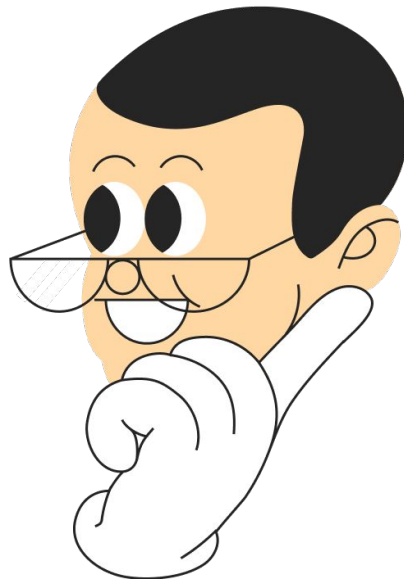
Рассмотрим пример ветвления, когда в случае истины мы выполним одно действие, а иначе – другое.

```
1 if (Условие) {  
2     // Действие1;  
3 } else {  
4     // Действие2;  
5 }  
6
```



## Тернарный оператор

**Тернарный оператор** – это операция, возвращающая либо второй, либо третий операнд в зависимости от условия (первого операнда)







## Разминка

**Задание** Что выведет консоль браузера?

→ `1 === "1"`

→ `1 == false`

→ `null !== 1`



## Консоль в chrome

Чтобы открыть консоль в Chrome, используйте сочетание клавиш:

- на Windows `Ctrl` + `Shift` + `J`
- на macOS `Option` + `Command` + `J`





## Пошаговая отладка кода

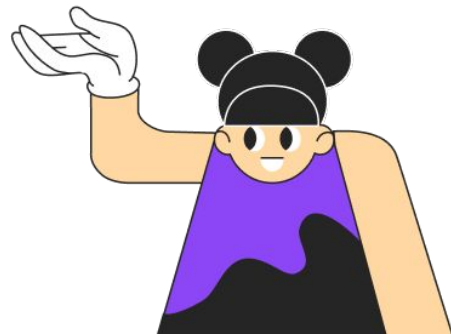
1. В данном примере мы видим, что выбрана вкладка Sources.
2. Выбираем необходимый файл.
3. Выставляем точку остановки.
4. Обновляем страницу в браузере.
5. Двигаемся по шагам.
6. При каждом шаге программа показывает какие значения принимает.
7. Также вы всегда можете выделить переменную или операцию в коде, чтобы узнать, чему она равняется на данный момент времени.



## Введение в браузерные события











Давайте для начала разберёмся, что такое браузерные события. Мы уже знаем что html — это контент сайта, а css — это стили для данного контента, но когда мы заходим на сайт мы очень часто выполняем какие-то действия на странице, например, открываем или закрываем окна, вводим значения в поле ввода, нажимаем на любые элементы. Именно это и называется **браузерными событиями**.





## Итоги урока

-  Рассмотрели какие бывают операторы
-  Рассмотрели операторы сравнения
-  Разобрали преобразования типов данных
-  Рассмотрели приведение типов
-  Рассмотрели встроенные функции `alert`, `prompt`, `confirm`
-  Узнали как работать с отладчиком браузера
-  Разобрали логические и арифметические операторы

**Спасибо**   
**за внимание**

