

Основы Javascript

и его работа в браузере

Хворостов В.А. 28.02.2022

Что такое JavaScript?

- Изначально JavaScript был создан, чтобы «сделать веб-страницы ЖИВЫМИ».
- Программы на этом языке называются скриптами. Они могут встраиваться в HTML и выполняться автоматически при загрузке веб-страницы.
- Скрипты распространяются и выполняются, как простой текст. Им не нужна специальная подготовка или компиляция для запуска.
- Сегодня JavaScript может выполняться не только в браузере, но и на сервере или на любом другом устройстве, которое имеет специальную программу, называемую «движком» JavaScript.

Что может JavaScript

- Возможности JavaScript сильно зависят от окружения, в котором он работает. Например, Node.js поддерживает функции чтения/записи произвольных файлов, выполнения сетевых запросов и т.д.
- В браузере для JavaScript доступно всё, что связано с манипулированием веб-страницами, взаимодействием с пользователем и веб-сервером.
- Возможности JavaScript в браузере ограничены ради безопасности пользователя.

Стандарты и спецификации

- Спецификация ECMA-262
<https://www.ecma-international.org/publications/standards/Ecma-262.htm>
- Таблицы совместимости:
 - <http://caniuse.com>
 - <https://kangax.github.io/compat-table>

Scripting engine ↕	Reference application(s) ↕	Conformance ^[46]			
		ES5 ^[47] ↕	ES6 (2015) ^[48] ↕	ES7 (2016) ^[49] ↕	Newer (2017+) ^{[49][50]} ↕
SpiderMonkey	Firefox 94	100%	98%	100%	100%
V8	Google Chrome 95, Microsoft Edge 95, Opera 80	100%	98%	100%	100%
JavaScriptCore	Safari 15	100%	99%	100%	90%

Hello world

```
1  <!DOCTYPE HTML>
2  <html>
3
4  <body>
5
6      <p>Перед скриптом...</p>
7
8      <script>
9          alert( 'Привет, мир!' );
10     </script>
11
12     <p>...После скрипта.</p>
13
14 </body>
15
16 </html>
```

Подключение в HTML

- Для добавления кода JavaScript на страницу используется тег `<script>`
- Скрипт во внешнем файле можно вставить с помощью `<script src="path/to/script.js"></script>`

Синтаксис

- Инструкции **могут** отделяться точкой с запятой.
- Комментарии
//
/* */
- Операторные скобки
{ }

Переменные

- let, const и var
- Имя переменной должно содержать только буквы, цифры или символы \$ и _.
- Первый символ не должен быть цифрой.
- Регистр имеет значение
- Зарезервированные слова

Строгий режим

- "use strict"

Типы данных

- Динамическая типизация
- number, Infinity, NaN, BigInt
- string
- boolean
- null
- undefined
- object, symbol
- typeof
- Преобразование типов: String, Number, Boolean, parseInt, parseFloat, +

Операторы

- бинарные
- унарные
- конкатенация
- инкремент, декремент, постфиксная и префиксная формы
- совмещение с присваиванием: `*=` `+=`
- сравнение: `>`, `<`, `>=`, `<=`, `==`, `!=`, `===`, `!==`
- логические: `||` (ИЛИ), `&&` (И) и `!` (НЕ)

Условия

- `if (a > b) { } else { }`
- `let result = условие ? значение1 : значение2;`

```
1  switch(x) {  
2    case 'value1': // if (x === 'value1')  
3      ...  
4      [break]  
5  
6    case 'value2': // if (x === 'value2')  
7      ...  
8      [break]  
9  
10   default:  
11     ...  
12     [break]  
13 }
```

Циклы

- `while (condition) { }`
- `do { } while (condition);`
- `break, continue`
- `for (let i = 0; i < 3; i++) { }`

Функции

- `function имя(параметры) { ...тело... }`
- `let имя = function(параметры) { ...тело... };`
- `let имя = (параметры) => выражение;`
- локальные и внешние переменные
- параметры по-умолчанию
- возврат значения (`return`)
- колбеки (`callback`)

Объекты

- `let user = new Object();`
- `let user = {};`

```
1  let user = {      // объект
2    name: "John",    // под ключом "name" хранится значение "John"
3    age: 30           // под ключом "age" хранится значение 30
4  };
```

```
1  // получаем свойства объекта:
2  alert( user.name ); // John
3  alert( user.age );  // 30
```

```
// доступ к свойству через переменную
alert( user[key] ); // John (если ввели "name")
```

Методы объектов

```
user.sayHi = function() {  
    alert("Привет!");  
};
```

```
user.sayHi(); // Привет!
```

```
user = {  
    sayHi: function() {  
        alert("Привет");  
    }  
};
```

// сокращённая запись выглядит лучше, не так ли?

```
user = {  
    sayHi() { // то же самое, что и "sayHi: function()"  
        alert("Привет");  
    }  
};
```


Ключевое слово «this»

```
1  let user = {  
2    name: "Джон",  
3    age: 30,  
4  
5    sayHi() {  
6      // this – это "текущий объект"  
7      alert(this.name);  
8    }  
9  
10 };  
11  
12 user.sayHi(); // Джон
```

Массивы

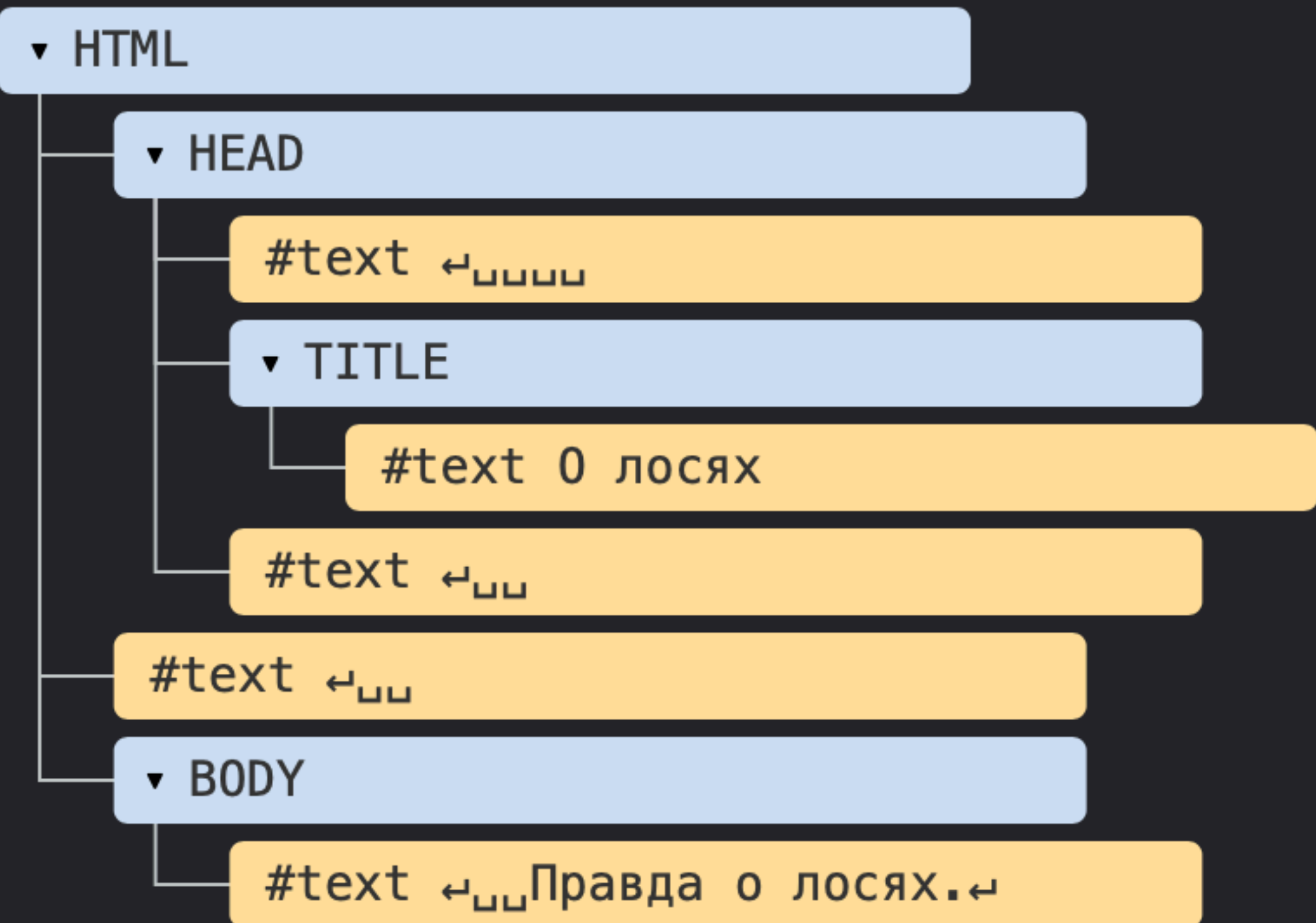
- `let arr = new Array();`
- `let arr = [];`
- Элементы массива нумеруются, начиная с нуля.
- Мы можем получить или заменить элемент, указав его номер в квадратных скобках:
`alert(arr[1]);`
`arr[2] = 'значение';`
- свойство `.length`, методы `.pop()`, `.push()`, `.shift()`, `.unshift()`

Методы массивов

- splice, slice, concat
- forEach
- indexOf/lastIndexOf и includes
- find и findIndex
- filter, map, reduce
- sort, reverse
- split, join

DOM

```
1 <!DOCTYPE HTML>
2 <html>
3 <head>
4   <title>0 лосях</title>
5 </head>
6 <body>
7   Правда о лосях.
8 </body>
9 </html>
```



getElement*, querySelector*

- document.getElementById()
- document.querySelectorAll(css-селектор)
- document.querySelector()
- elem.closest()

Codewars

<https://www.codewars.com/>

- Convert a Number to a String!
<https://www.codewars.com/kata/5265326f5fda8eb1160004c8/javascript>
- How good are you really?
<https://www.codewars.com/kata/5601409514fc93442500010b/javascript>