Разработка пользовательского веб-интерфейса

JavaScript

Сергей Геннадьевич Синица

КубГУ, 2020

sin@kubsu.ru

JavaScript

Динамическая типизация (но TypeScript)

Автоматическое управление памятью

Мультипарадигменный (ООП, прототипное, функциональное, событийное)

Интерпретируемый (но JIT, TraceMonkey Mozilla, V8 Google)

TOP1 по популярности в 2019+

JavaScript версии

ECMAScript (ES)

Спецификация стандартизована Ecma International в стандартах ECMA-262 и ISO/IEC 16262

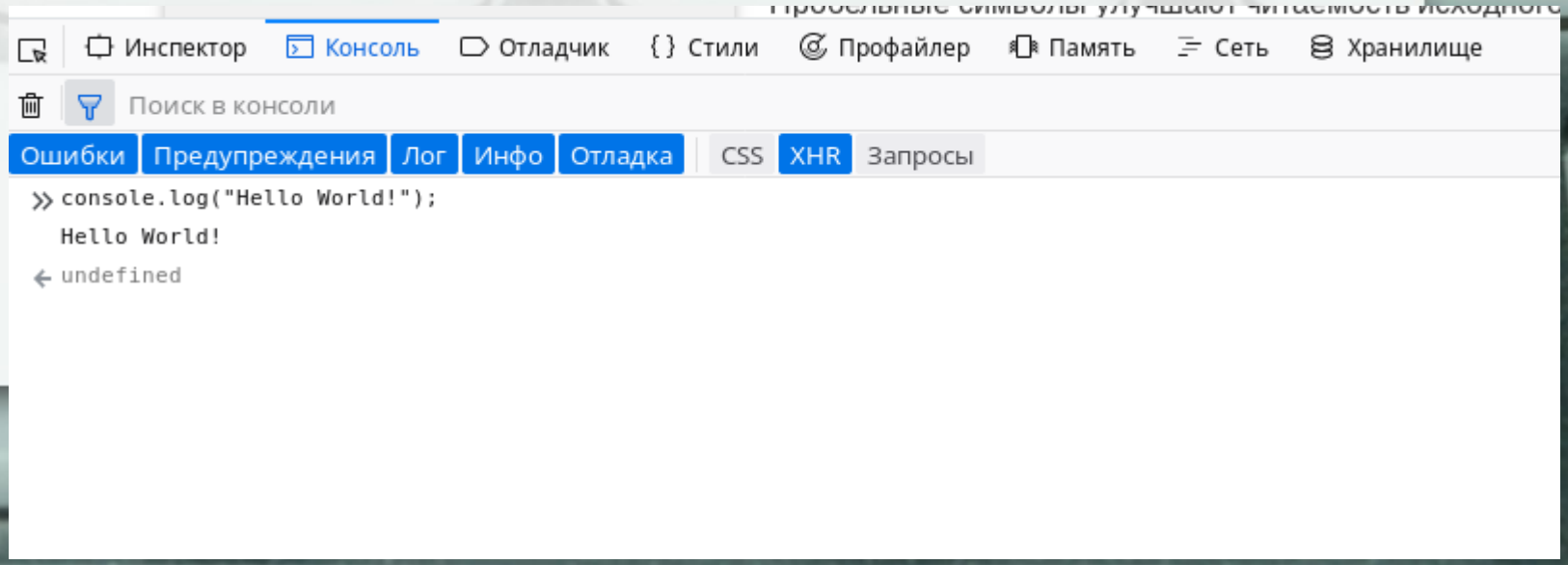
В большинстве браузеров ES2015 (ES6)

Код ES6+ транслируется в ES5 с помощью транспайлера (Babel)

Hello World!

Элемент script содержит программу, которая выполняется после загрузки элемента:

```
<script>
  var str = "hello, world!";
  var стр = "Привет, мир!";
  console.log(str);
  alert(стр);
</script>
```



Переменные

```
var x;
var a, b;
var z = 10;
z = "Переменная меняет тип";
x = z * 1.1;

// x = NaN
// y = undefined
```

Переменные

```
var apples = 5;

if (true) {
  var apples = 10;

  alert(apples); // 10 (внутри блока)
}

alert(apples); // 10 (снаружи блока то же
самое)
```

Переменные ES2015

```
let apples = 5; // (*)

if (true) {
  let apples = 10;

  alert(apples); // 10 (внутри блока)
}

alert(apples); // 5 (снаружи блока значение
не изменилось)
```

Динамическая типизация

```
let a = 3;
let b = "2";

let c = a + b;
console.log(c); // 32

let d = parseInt(a) + parseInt(b);
console.log(d); // 5

let e = a * b;
console.log(e); // 6

let f = a * parseInt(b);
console.log(f); // 6

let g = b * b;
console.log(g); // 4
```

Ключевые слова ES2015

break
case
catch
class
const
continue
debugger
default
delete
do
else
export
extends
finally
for
function
if
import
in
instanceof
new
return
super
switch
this
throw
try
typeof
var
void
while
with
yield

Разделители

```
<script>
    var name = "Max"
    alert("Рад вас видеть, " +
          name +
          "!");
</script>
```

Комментарии

```
// Однострочный

/*
Многострочный
комментарий
*/
```

Условный оператор

```
let year = prompt('В каком году появилась спецификация ECMAScript-2015?', '');

if (year == 2015) alert( 'Вы правы!' );


let year = prompt('В каком году появилась спецификация ECMAScript-2015?', '');

if (year == 2015) {
  alert( 'Да вы знаток!' );
} else {
  alert( 'А вот и неправильно!' ); // любое значение, кроме 2015
}

let accessAllowed = age > 18 ? true : false;

// Число 0, пустая строка "", null, undefined и NaN становятся false («falsy»).

// Остальные значения становятся true («truthy»).
```

Циклы

```
let i = 0;
while (i < 3) { // выводит 0, затем 1, затем 2
  alert( i );
  i++;
}


let i = 0;
do {
  alert( i );
  i++;
} while (i < 3);


for (let i = 0; i < 3; i++) { // выведет 0, затем 1, затем 2
  alert(i);
}
```

Функции, встроенные объекты, регулярки, null, ===

```
function getVowels(str) {
  var m = str.match(/[aeiou]/gi);
  if (m === null) {
    return 0;
  }
  return m.length;
}

console.log(getVowels('sky'));

// будет 0
```

Пример с формой

werwerwer

werwer wer OK

Пример с формой

```
<script>


function click1() {
  let f1 = document.getElementsByName("field1");
  let f2 = document.getElementsByName("field2");
  let r = document.getElementById("result");
  r.innerHTML = f1[0].value + f2[0].value;
  return false;
}


</script>


<div id="result"></div>
<form>
  <input name="field1" type="text">
  <input name="field2" type="text">
  <button id="button1" onclick="return click1();">OK</button>

</form>
```

DOM

```
<div id="mydiv">Первоначальный текст.</div>

<script>
// Ищем в DOM элемент с идентификатором mydiv
// и сохраняем ссылку на него в переменную mydiv.
var mydiv = document.getElementById('mydiv');

// Меняем свойство CSS background-color.
// Дефис заменяем на Camel Case.
mydiv.style.backgroundColor = 'gray';

// Меняем внутреннее содержимое элемента.
mydiv.innerHTML = 'Новый контент <strong>заменит</strong> старый.'
```

DOM

```
// Создаем новый элемент.
var div = document.createElement('div');
div.classList.add('my-new-div');
div.innerHTML = 'Первый новый элемент.';

// Вставляем его как последний дочерний элемент существующего.
mydiv.appendChild(div);

// Создаем новый элемент.
var div2 = document.createElement('div');
div.classList.add('my-new-div');
div2.innerHTML = 'Втрой новый элемент.';

// Вставляем div2 как дочерний элемент mydiv до div.
mydiv.insertBefore(div2, div);

// Удаляем элемент.
div.remove();

</script>
```

DOM

Window onload

Dom ready

`document.write()`

DOMContentReady

Events

```
<button id="my-button">Кнопка</button>

<script>
function onClick() {
  alert('click');
}

var b = document.getElementById('my-button');

b.addEventListener('click', onClick);

</script>
```

Events

```
<script>
function onClick() {
  alert("click");
}

window.addEventListener('DOMContentLoaded', function (event) {
  console.log("DOM fully loaded and parsed");
  let b = document.getElementById("my-button");
  b.addEventListener("click", onClick);
});
</script>

<button id="my-button">Кнопка</button>
```

Events

Наиболее важные:

load
change
focus
blur
click

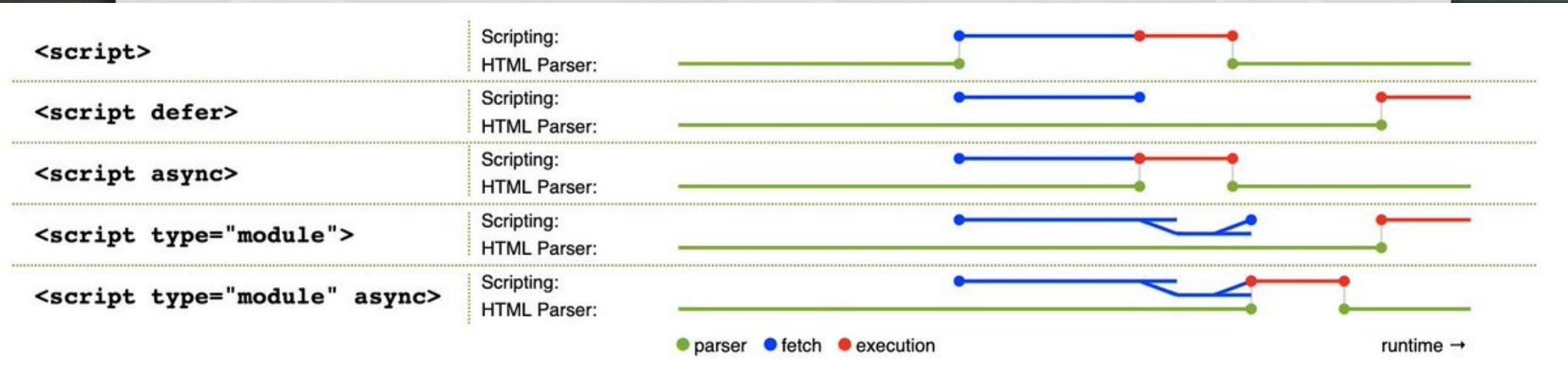
Все:
https://developer.mozilla.org/en-US/docs/Web/Events

Подключение скриптов

Синхронно:
<script src="URL_скрипта"></script>

Отложенно:
<script src="URL_скрипта" defer></script>

Параллельно асинхронно:
<script src="URL_скрипта" async></script>



jQuery

write less, do more.

Your donations help fund the continued development and growth of **jQuery.**

SUPPORT THE PROJECT

Download | API Documentation | Blog | Plugins | Browser Support

Search

Lightweight Footprint

Only 30kB minified and gzipped. Can also be included as an AMD module

CSS3 Compliant

Supports CSS3 selectors to find elements as well as in style property manipulation

Cross-Browser

Chrome, Edge, Firefox, IE, Safari, Android, iOS, and more

Download jQuery

v3.4.1

The 1.x and 2.x branches no longer receive patches.

View Source on GitHub →

How jQuery Works →

What is jQuery?

jQuery is a fast, small, and feature-rich JavaScript library. It makes things like HTML document traversal and manipulation, event handling, animation, and Ajax much simpler with an easy-to-use API that works across a multitude of browsers. With a combination of versatility and extensibility, jQuery has changed the way that millions of people write JavaScript.

Other Related Projects

jQuery user interface | jQuery mobile | QUnit | Sizzle

Resources

- jQuery Core API Documentation
- jQuery Learning Center
- jQuery Blog
- Contribute to jQuery
- About the jQuery Foundation
- Browse or Submit jQuery Bugs

JQuery подключение

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>Document</title>
</head>
<body>


    <!--Подключаем библиотеку-->
    <script src="js/jquery-версия.min.js"></script>
</body>
</html>
```

JQuery подключение

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>Document</title>
    <!--Подключаем библиотеку-->
    <script
src="https://ajax.googleapis.com/ajax/libs/jquery/2.2.0/j
query.min.js"></script>
</head>
<body>

</body>
</html>
```

JQuery подключение

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>Document</title>
</head>
<body>

    <!--Подключаем библиотеку-->
    <script
src="https://ajax.googleapis.com/ajax/libs/jquery/2.2.0/j
query.min.js"></script>
</body>
</html>
```

JQuery $, DOM ready

```
// A $( document ).ready() block.
$( document ).ready(function() {
    console.log( "ready!" );
});
```

Селекторы, события

```
.active {
  color: red;
}

…

<button class="btn-slide">b1</button>
<button class="btn-slide">b2</button>

…

$(document).ready(function(){
    $(".btn-slide").click(function(){
        $(this).toggleClass("active");
    });
});
```

Анимация

```
<div class="pane">
  <button class="delete">Delete</button>
</div>

$(document).ready(function(){

    $(".pane .delete").click(function(){
        $(this).parents(".pane").animate({ opacity:
"hide" }, "slow");
    });

});
```

Аккордеон

```
<div class="accordion">
  <h3>Question One Sample Text</h3>
  <p>Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Morbi malesuada, ante
at feugiat tincidunt, enim massa
            gravida metus, commodo lacinia massa diam vel eros. Proin eget urna. Nunc
fringilla neque vitae odio. Vivamus vitae
            ligula.</p>
  <h3>This is Question Two</h3>
  <p>Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Morbi malesuada, ante
at feugiat tincidunt, enim massa
            gravida metus, commodo lacinia massa diam vel eros. Proin eget urna. Nunc
fringilla neque vitae odio. Vivamus vitae
            ligula.</p>
  <h3>Another Questio here</h3>
  <p>Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Morbi malesuada, ante
at feugiat tincidunt, enim massa
            gravida metus, commodo lacinia massa diam vel eros. Proin eget urna. Nunc
fringilla neque vitae odio. Vivamus vitae
            ligula.</p>
 ...
    </div>
```

Аккордеон

Question One Sample Text

This is Question Two

Another Questio here

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Morbi malesuada, ante at feugiat tincidunt, enim massa gravida metus, commodo lacinia massa diam vel eros. Proin eget urna. Nunc fringilla neque vitae odio. Vivamus vitae ligula.

Sample heading

Sample Question Heading

Аккордеон

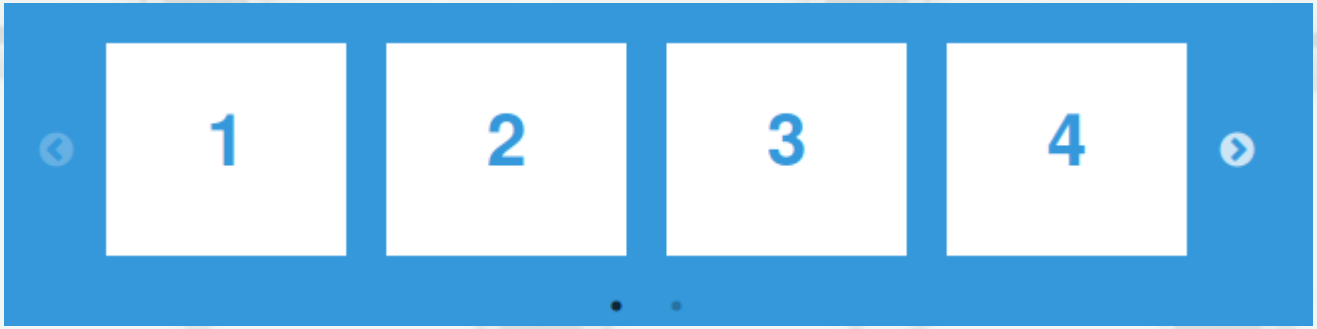
```
$(document).ready(function(){
    $(".accordion h3:first").addClass("active");
    $(".accordion p:not(:first)").hide();

    $(".accordion h3").click(function(){

        $(this).next("p").slideToggle("slow")
        .siblings("p:visible").slideUp("slow");
        $(this).toggleClass("active");
        $(this).siblings("h3").removeClass("active");
     });

 });
```

Плагины

```
<script src="//code.jquery.com/jquery-1.11.0.min.js"></script>
<script src="/slick/slick.min.js"></script>
<link rel="stylesheet" type="text/css" href="/slick/slick.css"/>
<link rel="stylesheet" type="text/css" href="/slick/slick-theme.css"/>
```

…

https://kenwheeler.github.io/slick/




```
<div class="slider responsive">
        <div>
            <h3>1</h3>
        </div>
        <div>
            <h3>2</h3>
        </div>
        <div>
            <h3>3</h3>
        </div>
        <div>
            <h3>4</h3>
        </div>
        <div>
            <h3>5</h3>
        </div>
        <div>
            <h3>6</h3>
        </div>
        <div>
            <h3>7</h3>
        </div>
        <div>
            <h3>8</h3>
        </div>
</div>
```

```
$('.responsive').slick({
  dots: true,
  infinite: false,
  speed: 300,
  slidesToShow: 4,
  slidesToScroll: 4,
  responsive: [
    {
      breakpoint: 1024,
      settings: {
        slidesToShow: 3,
        slidesToScroll: 3,
        infinite: true,
        dots: true
      }
    },
    {
      breakpoint: 600,
      settings: {
        slidesToShow: 2,
        slidesToScroll: 2
      }
    },
    {
      breakpoint: 480,
      settings: {
        slidesToShow: 1,
        slidesToScroll: 1
      }
    }
    // You can unslick at a given breakpoint now by adding:
    // settings: "unslick"
    // instead of a settings object
  ]
});
```

Прототипное ООП в ES5

```
//Пример наследования в прототипном программировании
//на примере языка JavaScript

//Создание нового объекта
var foo = {name: "foo", one: 1, two: 2};

//Создание еще одного нового объекта
var bar = {two: "two", three: 3};

bar.__proto__ = foo; // foo теперь является прототипом для bar

//Если теперь мы попробуем получить доступ к полям foo из bar
//то все получится
bar.one // Равно 1

//Свои поля тоже доступны
bar.three // Равно 3

//Собственные поля выше по приоритету полей прототипов
bar.two; // Равняется "two"
```