

WELCOME TO OUR PRESENTATION

Алгоритмизация и программирование
Урок 1. Введение в JavaScript

Введение в JavaScript

Содержание

- Как появился JavaScript.
- Особенности исполнения кода JavaScript.
- Возможности и ограничения языка.
- Развитие языка. Стандарты.
- Языки поверх JavaScript.
- Экосистема JavaScript.
- Литература



Введение в JavaScript

Как появился JavaScript ?

В 90-е годы Веб нуждался в лёгком скриптовом языке (*или языке сценариев — прим.ред.*), способном работать с DOM, который в те дни не был стандартизирован.



**JAVA – для
инженеров
JAVA – для
дизайнеров**

```
typeof "hello world"  
"string"  
typeof new String('hello world')  
"object"
```

Поспешная разработка стала причиной большого количества архитектурных ошибок, которые могли стать настоящим кошмаром.

**Брендан
Айк**



Введение в JavaScript

Особенности исполнения кода JavaScript

ОСОБЕННОСТИ ИСПОЛНЕНИЯ КОДА JAVASCRIPT



Введение в JavaScript

Особенности исполнения кода JavaScript

<https://es6console.com/>

The screenshot displays the ES6 Console web application. The interface includes a top navigation bar with 'Run', 'Save', and 'Transform' buttons. On the left, there is a sidebar with 'COMMANDS' (save code, run, transform), 'COMPILER' (babel 6, babel 5, traceur, typescript, regenerator, prepack), and 'PRESETS' (es2015, es2016, es2017, react, stage-0, stage-1, stage-2, stage-3). The main editor area contains the following JavaScript code:

```
1 var sym = "sdfgdfgs";  
2 console.log(typeof sym );  
3
```

The console output shows the result of the execution: 'string'. The interface also includes a 'Settings' gear icon in the top right corner.

Введение в JavaScript

Особенности исполнения кода

```
<!DOCTYPE HTML>
<html>
<head>
  <meta charset="utf-8" />
</head>
<body>
Первая программа
<br>
<script>
  document.write("Hello World!");
</script>
</body>
</html>
```

The image shows a browser window displaying the output of the code: "Первая программа" followed by "Hello World!". Below the browser window is the developer console, which shows the execution of the code. The console output is as follows:

```
>> alert("Всплывающее сообщение")
← undefined
>> 2+2
← 4
>> console.log("Вывод нв консоль")
    Вывод нв консоль                                     debugger eval code:1:1
← undefined
```



Введение в JavaScript

Возможности и ограничения языка

ВОЗМОЖНОСТИ И ОГРАНИЧЕНИЯ ЯЗЫКА

Введение в JavaScript

Возможности JavaScript

- **Добавлять различные эффекты анимации.**
- **Реагировать на события - обрабатывать перемещения указателя мыши, нажатие клавиш с клавиатуры.**
- **Осуществлять проверку ввода данных в поля формы до отправки на сервер, что в свою очередь снимает дополнительную нагрузку с сервера.**
- **Создавать и считывать cookie, извлекать данные о компьютере посетителя.**
- **Определять браузер.**
- **Изменять содержимое HTML-элементов, добавлять новые теги, изменять стили.**



Введение в JavaScript

Ограничения JavaScript

- JavaScript не может закрывать окна и вкладки, которые не были открыты с его помощью.
- Не может защитить исходный код страницы и запретить копирование текста или изображений со страницы.
- Не может осуществлять кроссдоменные запросы, получать доступ к веб-страницам, расположенным на другом домене. Даже когда страницы из разных доменов отображаются в одно и тоже время в разных вкладках браузера, то код JavaScript принадлежащий одному домену не будет иметь доступа к информации о веб-странице из другого домена. Это гарантирует безопасность частной информации, которая может быть известна владельцу домена, страница которого открыта в соседней вкладке.
- Не имеет доступа к файлам, расположенным на компьютере пользователя, и доступа за пределы самой веб-страницы, единственным исключением являются файлы cookie, это небольшие текстовые файлы, которые JavaScript может записывать и считывать.

Введение в JavaScript

Развитие языка. Стандарты

РАЗВИТИЕ ЯЗЫКА. СТАНДАРТЫ



Введение в JavaScript

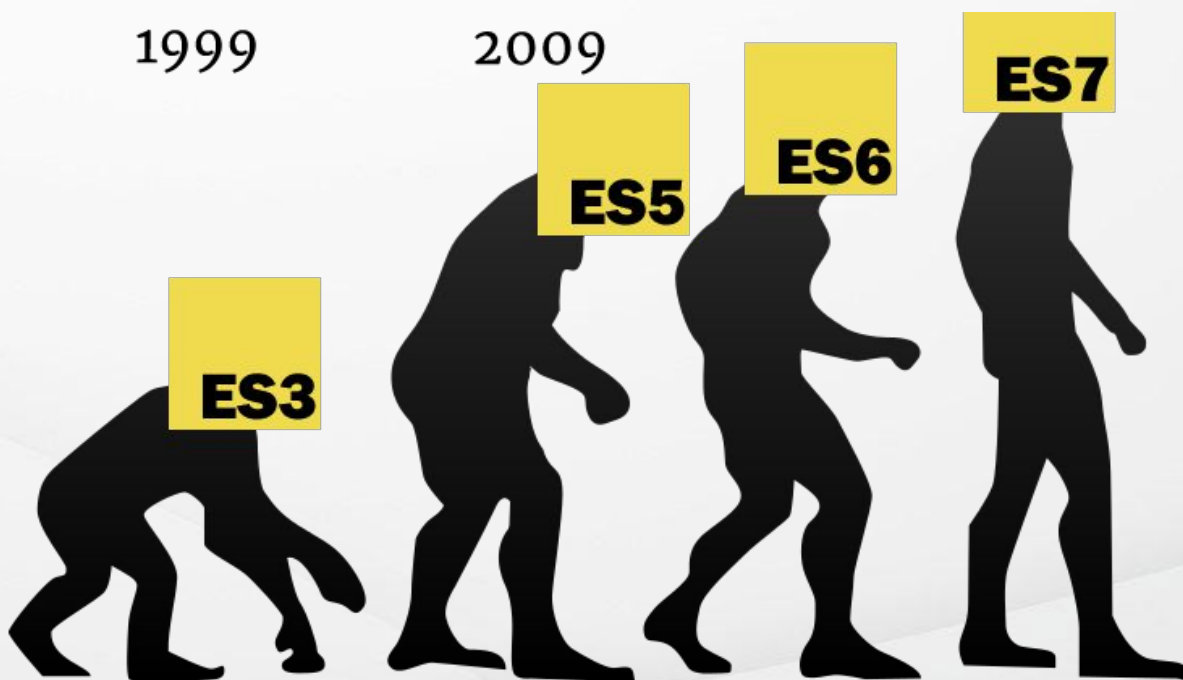
Стандарты языка JavaScript

ECMAScript - это объектно-ориентированный язык программирования с прототипной организацией, имеющий концепцию объекта в качестве базовой абстракции.

«**ES**» – это сокращение от «ECMAScript». Изначально номера версий стандарта обозначались цифрами, прибавляемыми к «ES».

Первые три релиза выходили ежегодно (1997-99гг.), это **ES1**, **ES2** и **ES3**. Четвертый выпуск не состоялся «в связи с политическими расхождениями, касающимися запутанности языка». Следующая версия, **ES5**, увидела свет почти десять лет спустя, в декабре 2009 года.

Начиная с шестой версии было решено возобновить ежегодный выпуск релизов, а вместо порядкового номера добавлять год издания. Таким образом, в 2015 году вышла не **ES6**, а **ES2015**. В течение последующих двух лет вышли **ES2016** и **ES2017**.



Введение в JavaScript

Новое в ES2017

```
//Exponentiation operator
```

```
2 ** 3; //8
```

```
//Array#includes(..)
```

```
let vals = [ "foo" , "bar" , 42, "baz" ];
```

```
if (vals.includes( 42 )) {
```

```
  // found it!
```

```
}
```

```
  //destructured properties
```

```
let o1 = { b: 2, c: 3, d: 4 };
```

```
let { b, ... o2 } = o1;
```

```
console.log(b, o2.c, o2.d); // 2 3 4
```

```
//SIMD (Single Instruction Multiple Data)
```

```
var a = SIMD.Float32x4(1, 2, 3, 4);
```

```
var b = SIMD.Float32x4(5, 6, 7, 8);
```

```
var c = SIMD.Float32x4.add(a,b); // Float32x4[6,8,10,12]
```

```
//Object property and ...
```

```
let o1 = { a: 1, b: 2 }, o2 = { c: 3 };
```

```
let o3 = { ... o1, ... o2, d: 4 };
```

```
console.log(o3.a, o3.b, o3.c, o3.d);
```

```
// 1 2 3 4
```

```
//Async function
```

```
async function main() {
```

```
  let ret = await step1();
```

```
  ret = await step2( ret );
```

```
  ret = await Promise.all([
```

```
    step3a( ret ),
```

```
    step3b( ret ),
```

```
    step3c( ret )]);
```

```
  await step4( ret );
```

```
}
```

```
//object.observe(..)
```

```
let obj = { a: 1, b: 2 };
```

```
Object.observe(obj, function(changes){
```

```
  for (var change of changes) {
```

```
    console.log( change );
```

```
  }
```

```
},
```

```
  [ "add" , "update" , "delete" ]
```

```
);
```

```
obj.c = 3;
```

```
// { name: "c", object: obj, type: "add" }
```



Введение в JavaScript

Языки поверх JavaScript

ЯЗЫКИ ПОВЕРХ JAVASCRIPT



Введение в JavaScript

CoffeeScript

CoffeeScript это небольшой язык, который транслируется в Javascript. Рубистам он кажется похожим на рубли, питонистам он похож на питон, и конечно же, он похож на яваскрипт. CoffeeScript старается упростить использование яваскрипта, сохранив все его сильные стороны.



Assignment:

number = 42
opposite = true

Conditions:

number = -42 if opposite

Functions:

square = (x) -> x * x

Arrays:

list = [1, 2, 3, 4, 5]

Objects:

math =
 root: Math.sqrt
 square: square
 cube: (x) -> x * square x

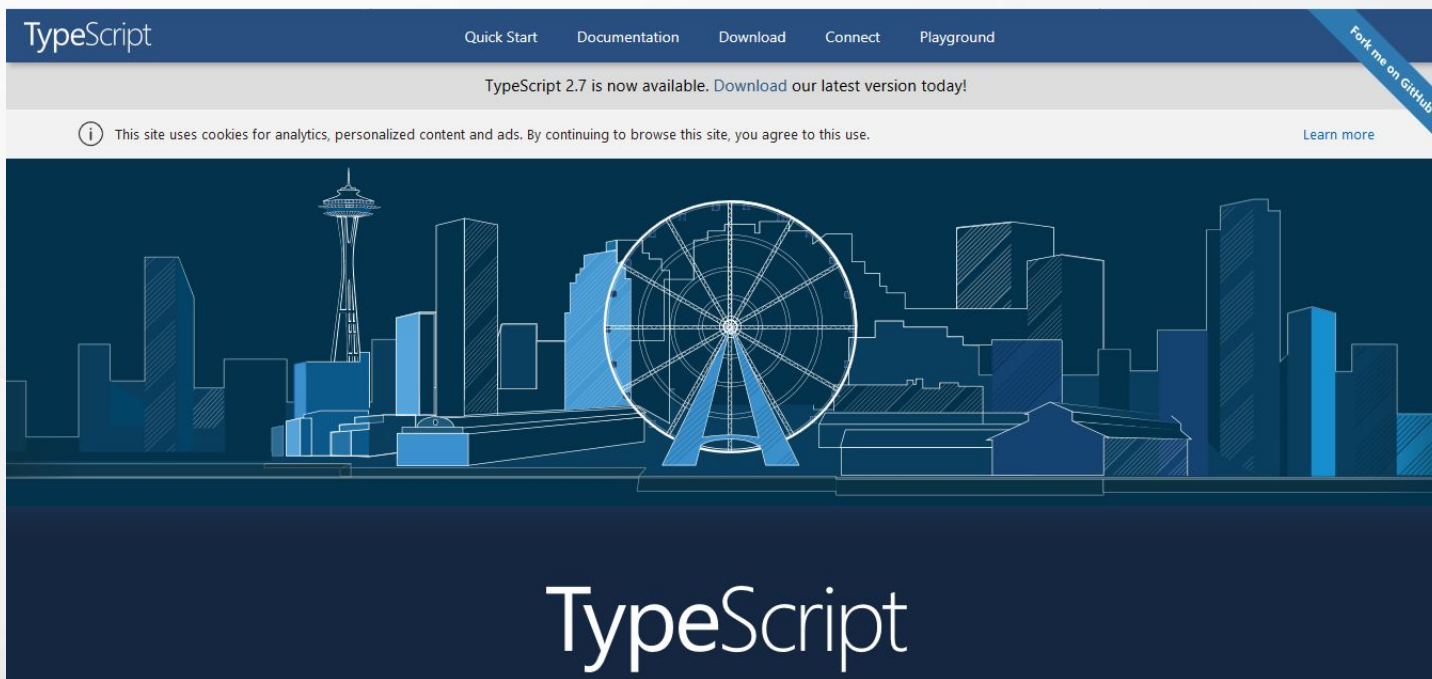
<http://coffeescript.org/>



Введение в JavaScript

TypeScript

TypeScript отличается от JavaScript возможностью явного определения типов (статическая типизация), поддержкой использования полноценных классов (как в традиционных объектно-ориентированных языках), а также поддержкой подключения модулей.



<http://www.typescriptlang.org/>

```
interface Person {  
  firstName: string;  
  lastName: string;  
}
```

```
function greeter(person: Person) {  
  return "Hello, " + person.firstName + " " +  
  person.lastName;  
}
```

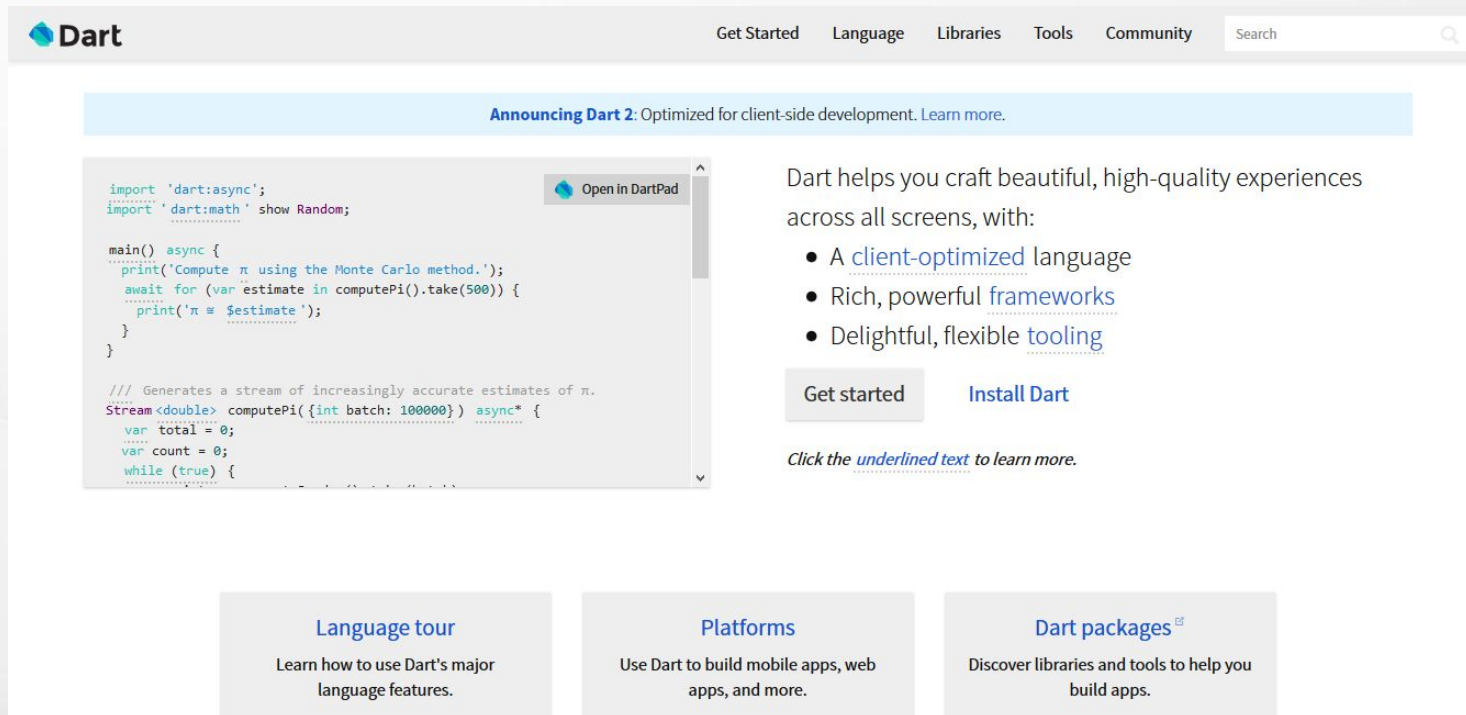
```
let user = { firstName: "Jane", lastName: "User" };  
  
document.body.innerHTML = greeter(user);
```



Введение в JavaScript

Dart

Dart это объектно-ориентированный язык с полноценной системой классов, лексическими скоупами, замыканиями, и опционально — статической типизацией. Dart помогает создавать структурированные веб приложения и легок в изучении для широкого круга разработчиков.



The screenshot shows the Dart website homepage. At the top, there is a navigation bar with links for 'Get Started', 'Language', 'Libraries', 'Tools', and 'Community', along with a search bar. Below the navigation bar, there is a blue banner announcing 'Announcing Dart 2: Optimized for client-side development. Learn more.' The main content area features a code editor on the left with Dart code for calculating pi using the Monte Carlo method. To the right of the code editor, there is a text block stating 'Dart helps you craft beautiful, high-quality experiences across all screens, with:' followed by a bulleted list of features: 'A client-optimized language', 'Rich, powerful frameworks', and 'Delightful, flexible tooling'. Below the list are two buttons: 'Get started' and 'Install Dart'. At the bottom of the page, there are three cards: 'Language tour' (Learn how to use Dart's major language features), 'Platforms' (Use Dart to build mobile apps, web apps, and more), and 'Dart packages' (Discover libraries and tools to help you build apps).

```
class Point {  
  final double x, y;  
  const Point(this.x, this.y);  
  bool get isInsideUnitCircle => x * x + y * y <= 1;  
}
```

<http://www.dartlang.org/>



Введение в JavaScript

...

ЭКОСИСТЕМА JAVASCRIPT



Введение в JavaScript

Экосистема JavaScript

Среда JavaScript стала просто огромной. Она имеет собственную экосистему библиотек, фреймворков, инструментов, менеджеров пакетов и новых языков, которые компилируются до JavaScript.

API - это интерфейс взаимодействия с программой

извне.

Библиотека — это готовый к использованию набор кода, который бежит в контексте приложения, и точно так же выполняет свою работу. То есть библиотека становится при подключении частью приложения.

Фреймворк — его функции, в отличие от библиотеки, не вызываются вами, а наоборот, ваш код вызывается из него. Фреймворк можно представить себе в виде полуфабриката приложения, к которому вы дописываете нужную функциональность сами.

Библиотеки и фреймворки предоставляют

API.

Библиотека может собираться фреймворком
платформы.

API это

паттерн.

Фреймворк состоит из библиотек или является собой паттерн их
соединения.

API может создаваться на базе
фреймворка.

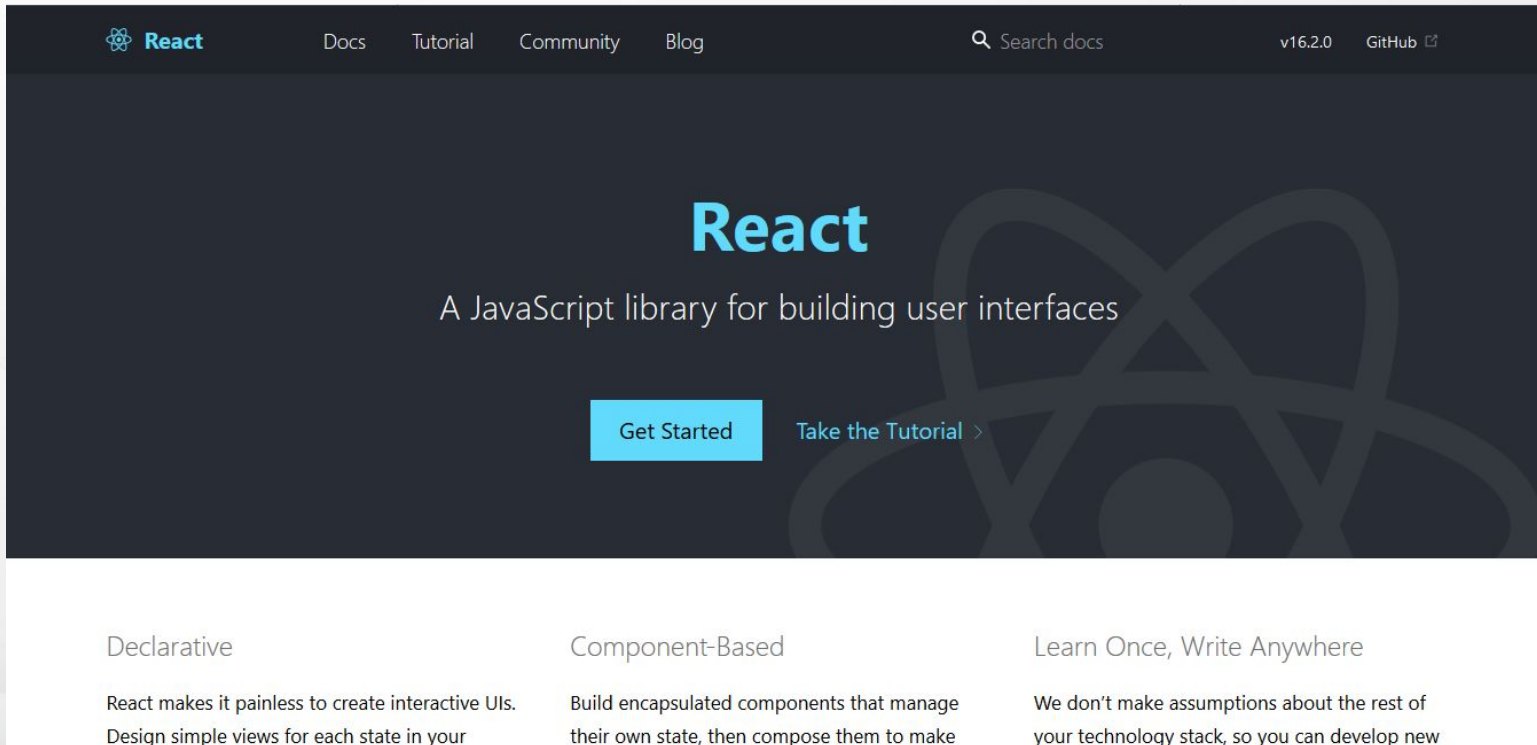


Введение в JavaScript

Библиотека React

React — это библиотека JavaScript, созданная разработчиками Facebook и Instagram. React была признана самой любимой технологией среди разработчиков, согласно опросу Stack Overflow Survey 2017.

React является самым популярным проектом JavaScript, основываясь на подсчете звезд GitHub.



<https://reactjs.org/>

Можно создать интерактивный интерфейс с использованием декларативного подхода.

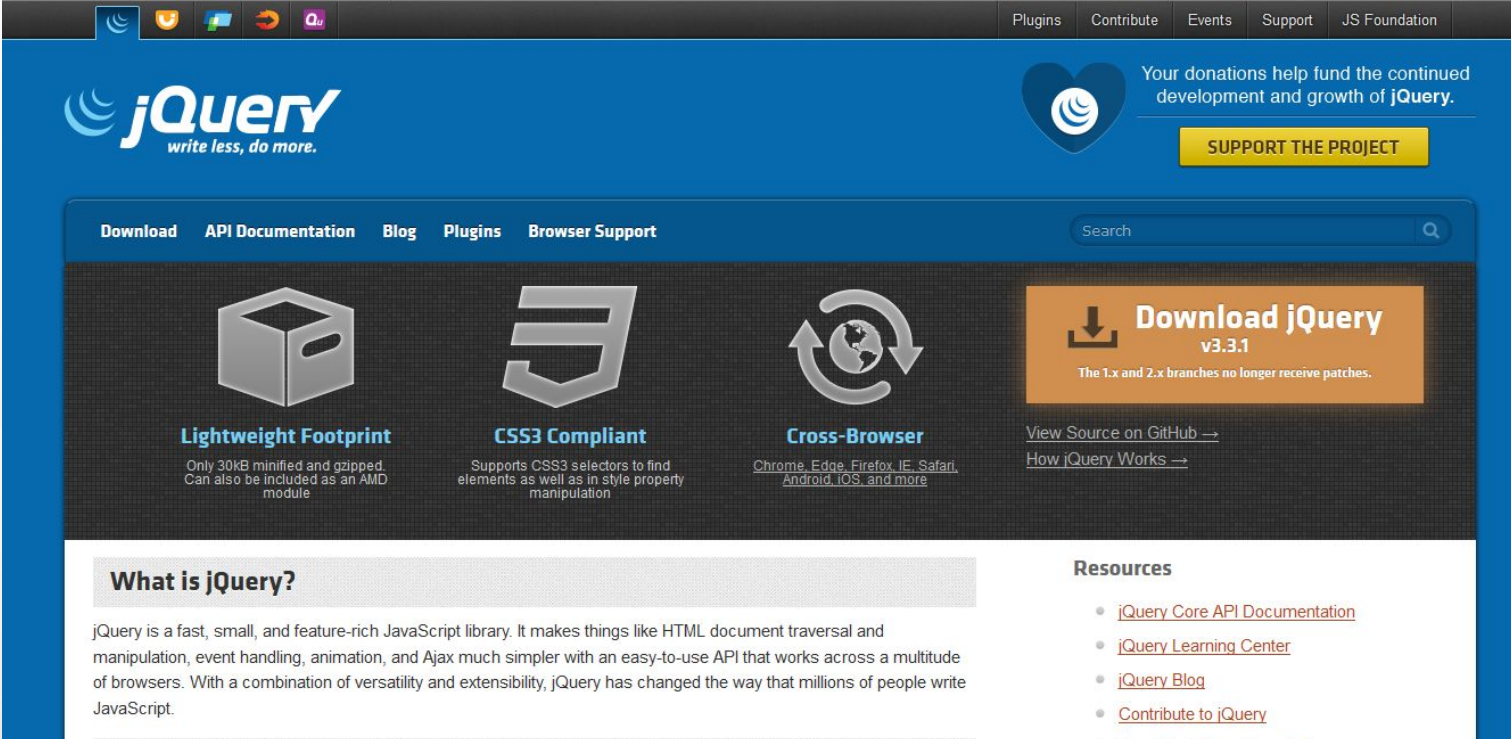
Использование компонентной модели и виртуального DOM.

Введение в JavaScript

Библиотека jQuery

jQuery — это библиотека, которая рассчитана на манипулирование DOM на основе селекторов CSS, обработку событий и создание вызовов AJAX.

<https://jquery.com/>



The screenshot shows the jQuery website homepage. At the top, there's a navigation bar with links for Plugins, Contribute, Events, Support, and JS Foundation. The main header features the jQuery logo with the tagline "write less, do more." and a call to action "SUPPORT THE PROJECT" with a yellow button. Below the header, there's a search bar and a navigation menu with links for Download, API Documentation, Blog, Plugins, and Browser Support. The main content area is divided into four columns: "Lightweight Footprint" (30kB minified and gzipped), "CSS3 Compliant" (supports CSS3 selectors), "Cross-Browser" (supports Chrome, Edge, Firefox, IE, Safari, Android, iOS, etc.), and "Download jQuery v3.3.1" (with a note that 1.x and 2.x branches no longer receive patches). Below this, there's a "What is jQuery?" section and a "Resources" section with links to jQuery Core API Documentation, jQuery Learning Center, jQuery Blog, and Contribute to jQuery.

Введение в JavaScript

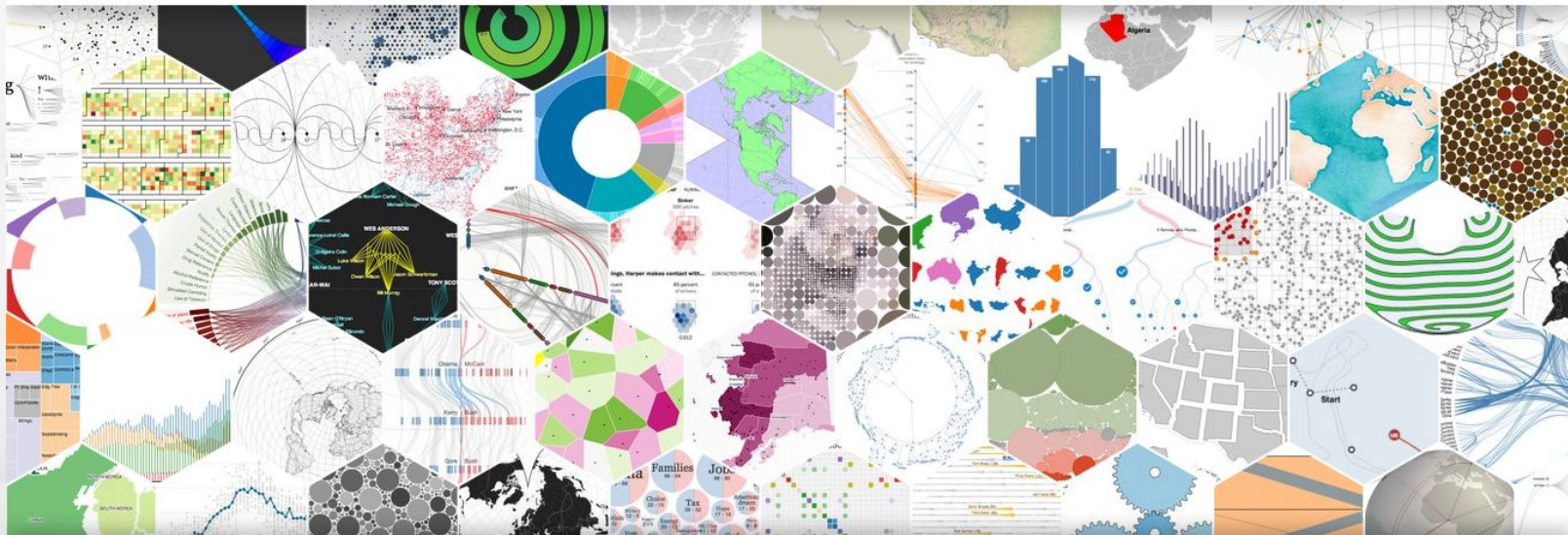
Библиотека D3: Data-Driven Documents

D3 (или D3.js) - мощная библиотека JavaScript для создания интерактивных визуализаций с использованием веб-стандартов, таких как SVG, HTML и CSS. В отличие от других библиотек визуализации, D3 предлагает лучший контроль над окончательным визуальным результатом.

[Overview](#) [Examples](#) [Documentation](#) [Source](#)

Data-Driven Documents

Fork me on GitHub



<https://d3js.org/>

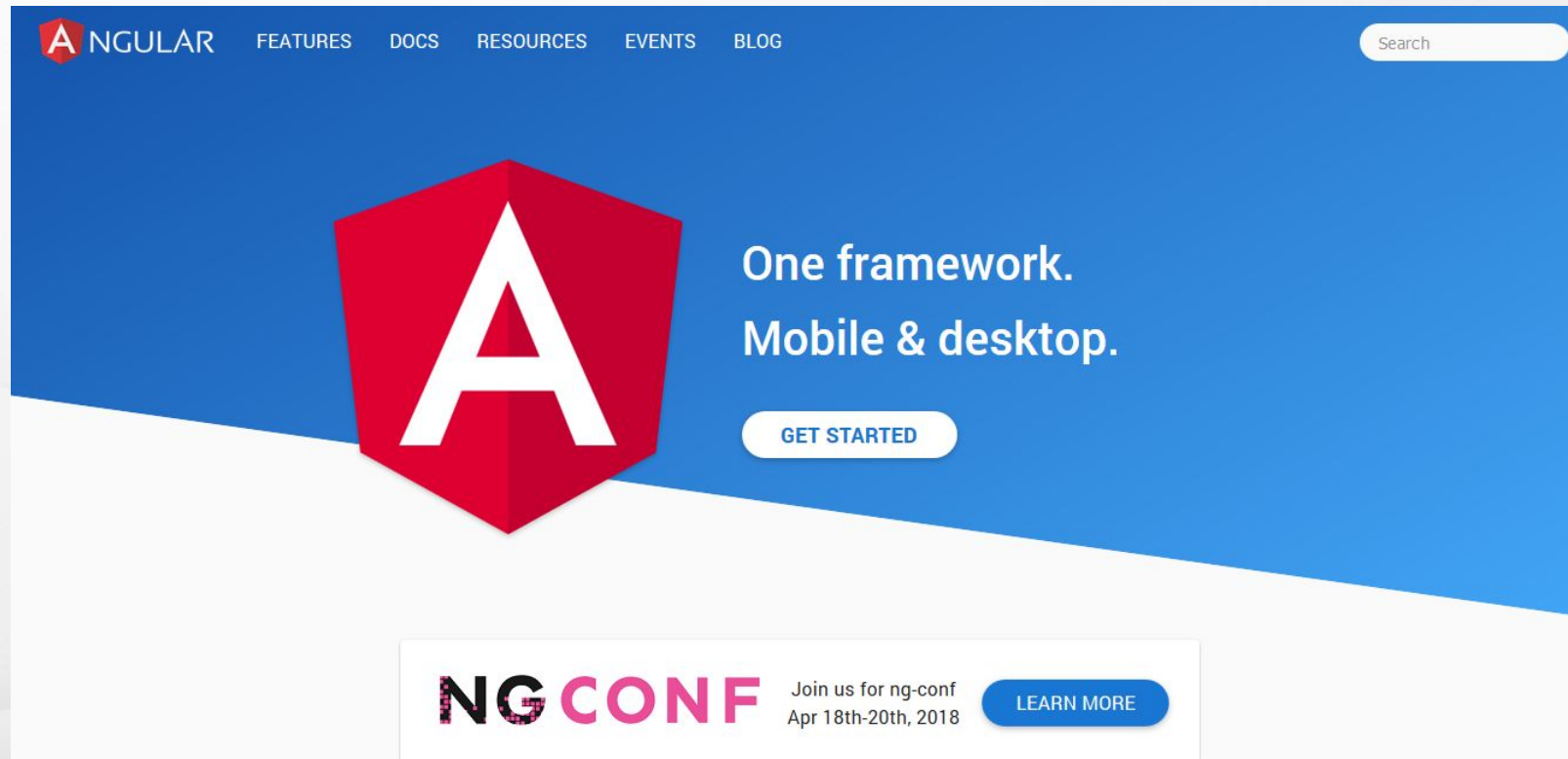
D3 работает, связывая данные с DOM и затем преобразуя их в документ.



Введение в JavaScript

Фреймворк Angular

Angular – фреймворк для разработки мобильных и Desktop-интерфейсов на JavaScript.



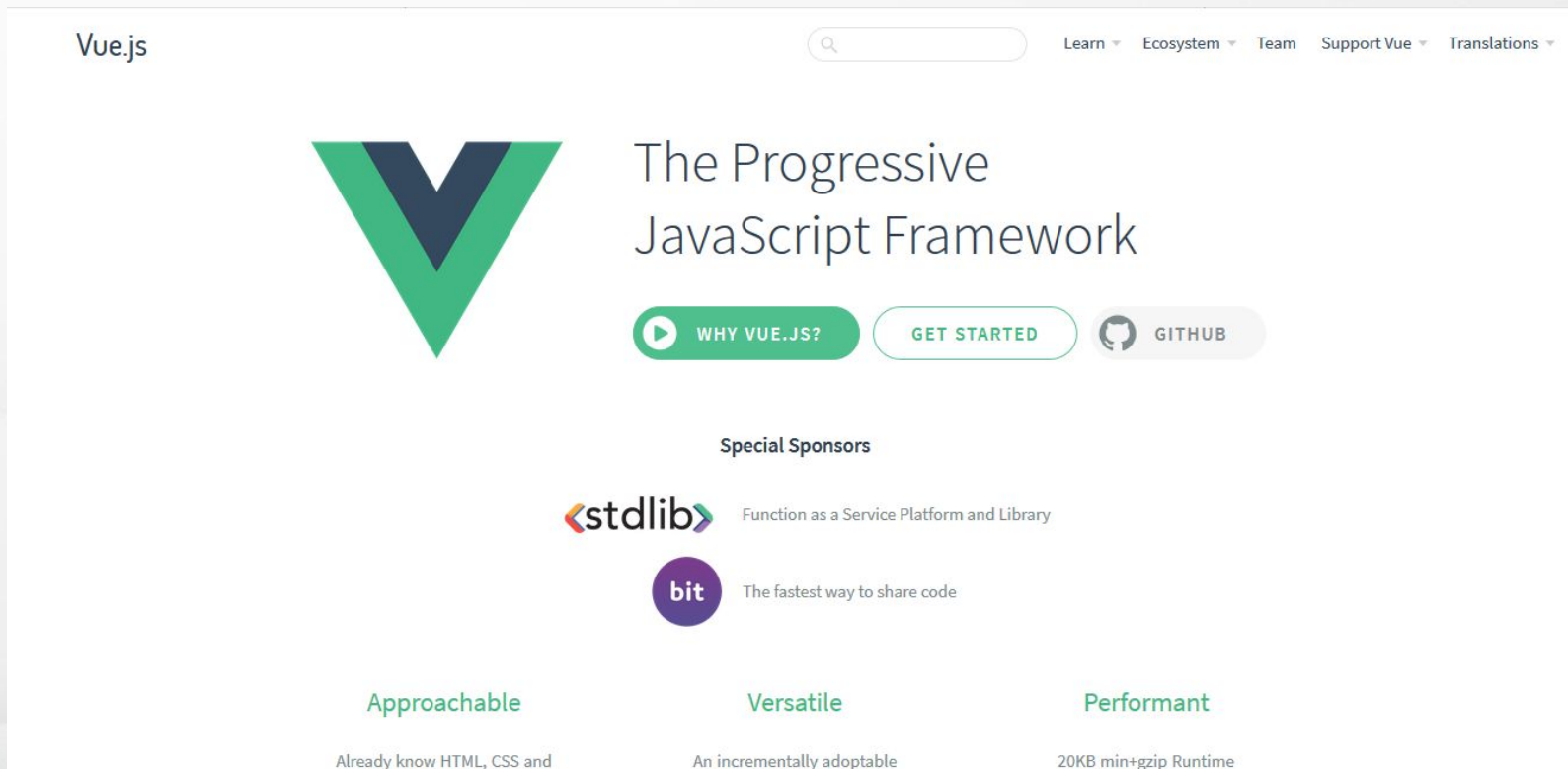
<https://angular.io/>

Некоторые из функций Angular 2 включают:

- TypeScript поверх JavaScript как язык по умолчанию
- компонентная архитектура
- улучшенная производительность как на мобильных, так и на веб-платформах
- лучшие инструменты и варианты скаффолдинга

Фреймворк Vue.js

Vue.js – фреймворк предлагает опыт, похожий на React, с его виртуальными DOM и компонентами многократного использования, которые можно использовать для создания как виджетов, так и целых веб-приложений.



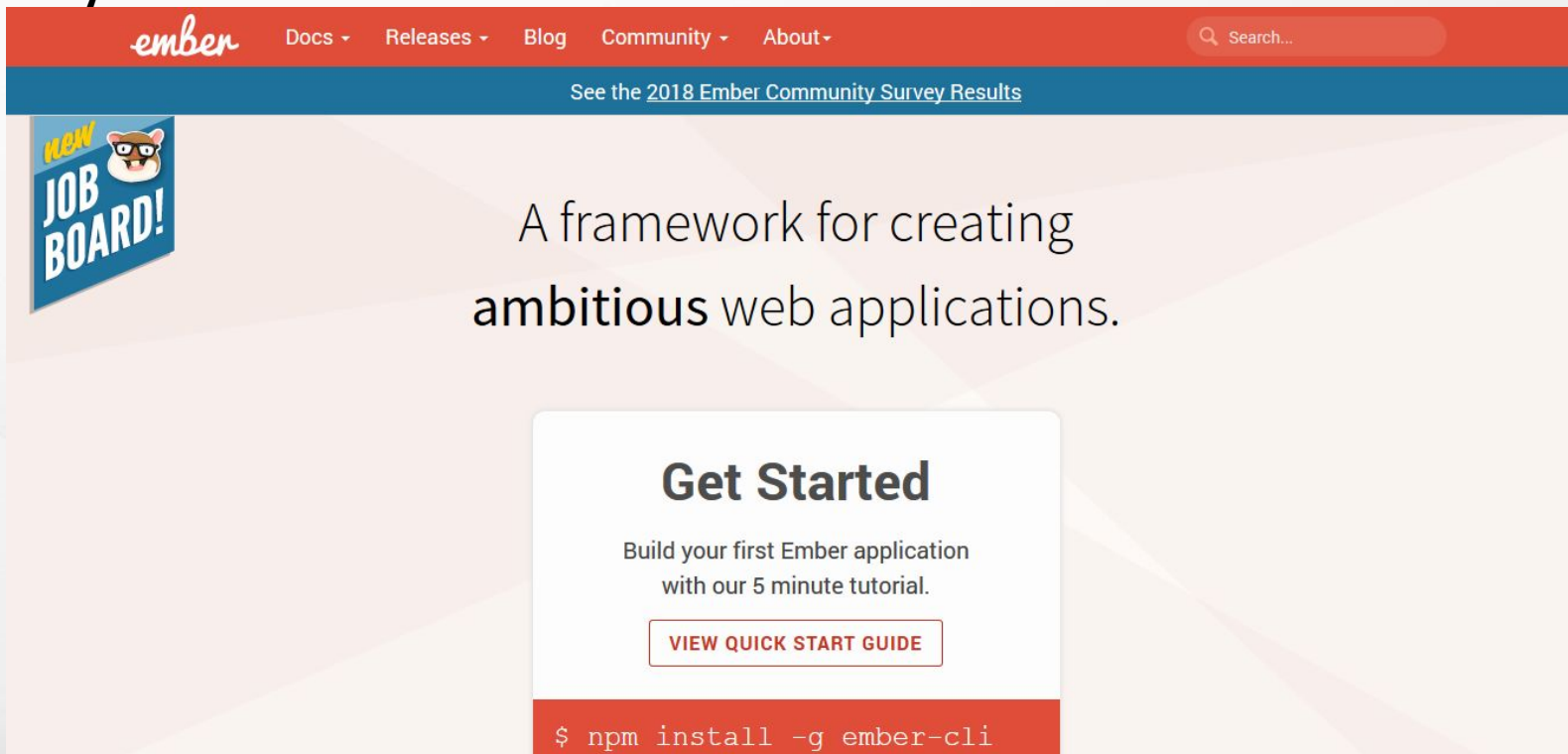
<https://vuejs.org/>

Vue.js использует систему реактивности, чтобы определить, что изменилось и перераспределяет минимальное количество компонентов.

Введение в JavaScript

Экосистема JavaScript

Ember.js является интерфейсом на основе шаблона Model-View-ViewModel (MVVM). Он следует за стандартным подходом к настройке, который популярен среди сторонних фреймворков, таких как Ruby on Rails и Laravel.



The screenshot shows the Ember.js website homepage. At the top, there is a navigation bar with the Ember logo and links for Docs, Releases, Blog, Community, and About. A search bar is also present. Below the navigation bar, there is a blue banner with the text "See the 2018 Ember Community Survey Results". On the left side, there is a "NEW JOB BOARD!" badge featuring a cartoon animal. The main content area features the text "A framework for creating ambitious web applications." and a "Get Started" section with a button labeled "VIEW QUICK START GUIDE". At the bottom of the "Get Started" section, there is a terminal command: `$ npm install -g ember-cli`.

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

Стек Ember обычно включает:

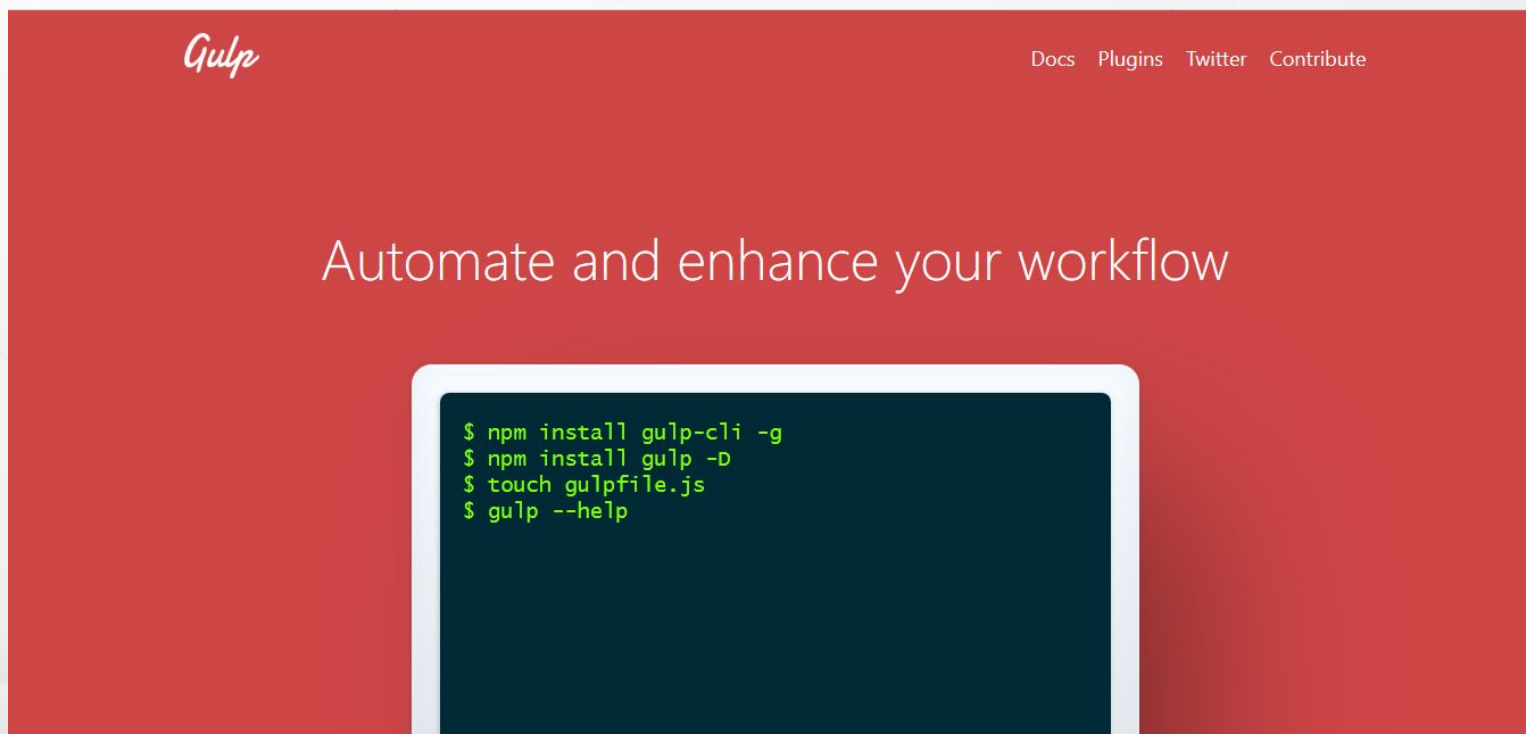
- Ember CLI: предоставляет основные варианты скаффолдинга и поддерживает сотни надстроек.
- Ember Data: библиотека сохранения данных, которая может быть настроена для работы с любым сервером.
- Ember Inspector: расширение доступно для Chrome и Firefox.
- Liquid Fire: дополнение для переходов и анимации.



Введение в JavaScript

Инструмент автоматизации Gulp

Gulp - это инструментарий JavaScript, который используется как исполнитель задач и как система сборки в веб-разработке. Компиляция, минимизация кода, оптимизация изображения, модульное тестирование, листинг и т.д. - это повторяющиеся задачи, которые должны быть автоматизированы.



<https://gulpjs.com/>

Gulp использует конвейеры для потоковой передачи данных из одного плагина в другой, и конечный результат выводится в папку назначения.



Введение в JavaScript

Инструмент автоматизации Gulp

Babel.JS – это транспайлер, переписывающий код на ES-2015 (и выше) в код на предыдущем стандарте ES5.

Он состоит из двух частей:

- Собственно транспайлер, который переписывает код.
- Полифилл, который добавляет методы `Array.from`, `String.prototype.repeat` и другие.



<https://babeljs.io/>

Транспайлер (Transpiler) - это тот же транслятор с той лишь разницей, что у результата примерно тот же уровень абстракции, что и у исходного текста.

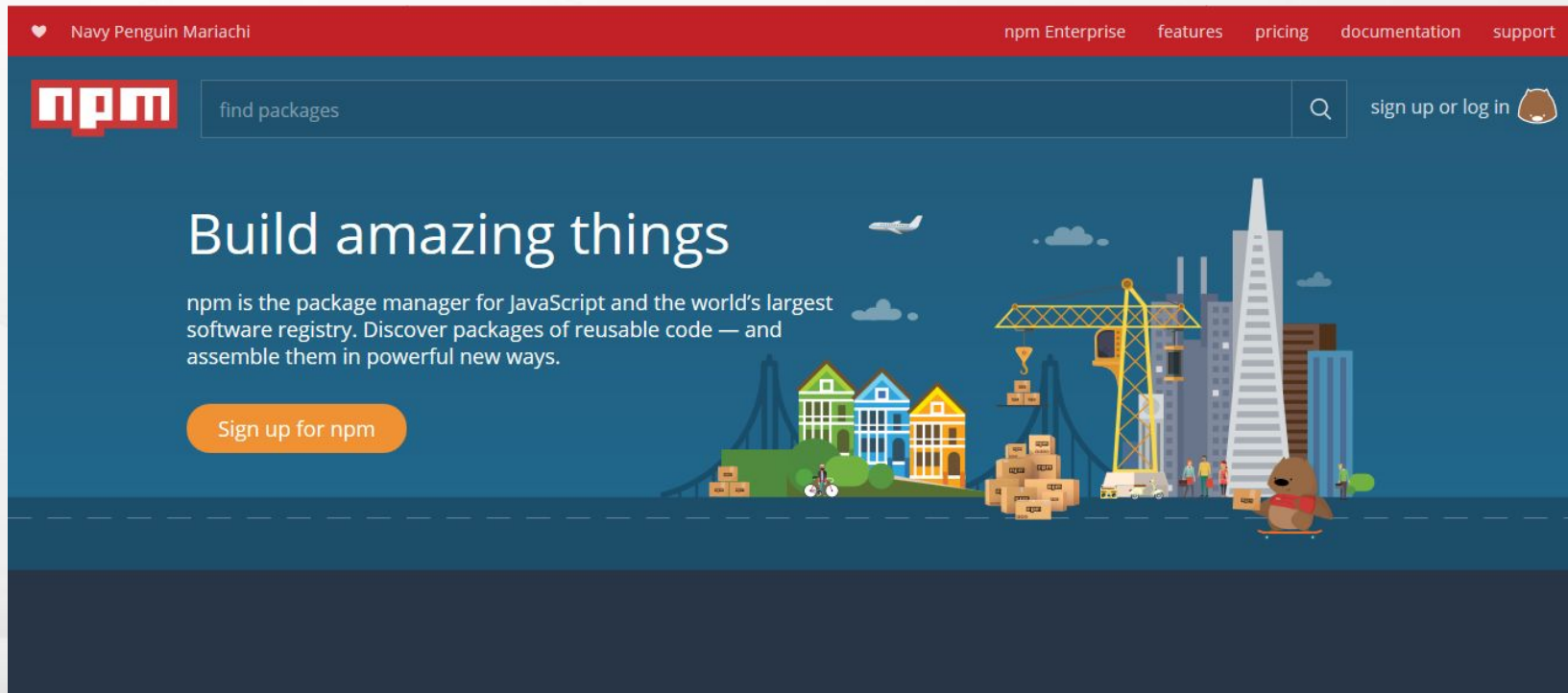
«Полифилл» (англ. polyfill) – это библиотека, которая добавляет в старые браузеры поддержку возможностей, которые в современных браузерах являются встроенными.



Введение в JavaScript

Пакетный менеджер NPM

NPM. Gulp требует, чтобы вы потратили время на изучение и освоение нового инструмента, что займет у вас так же время. Ввода дополнительных зависимостей в ваш проект можно избежать, выбирая альтернативу, которая уже связана с Node.js. Хотя npm больше известен как менеджер пакетов, сценарии npm можно использовать для выполнения основной части вышеупомянутых задач.

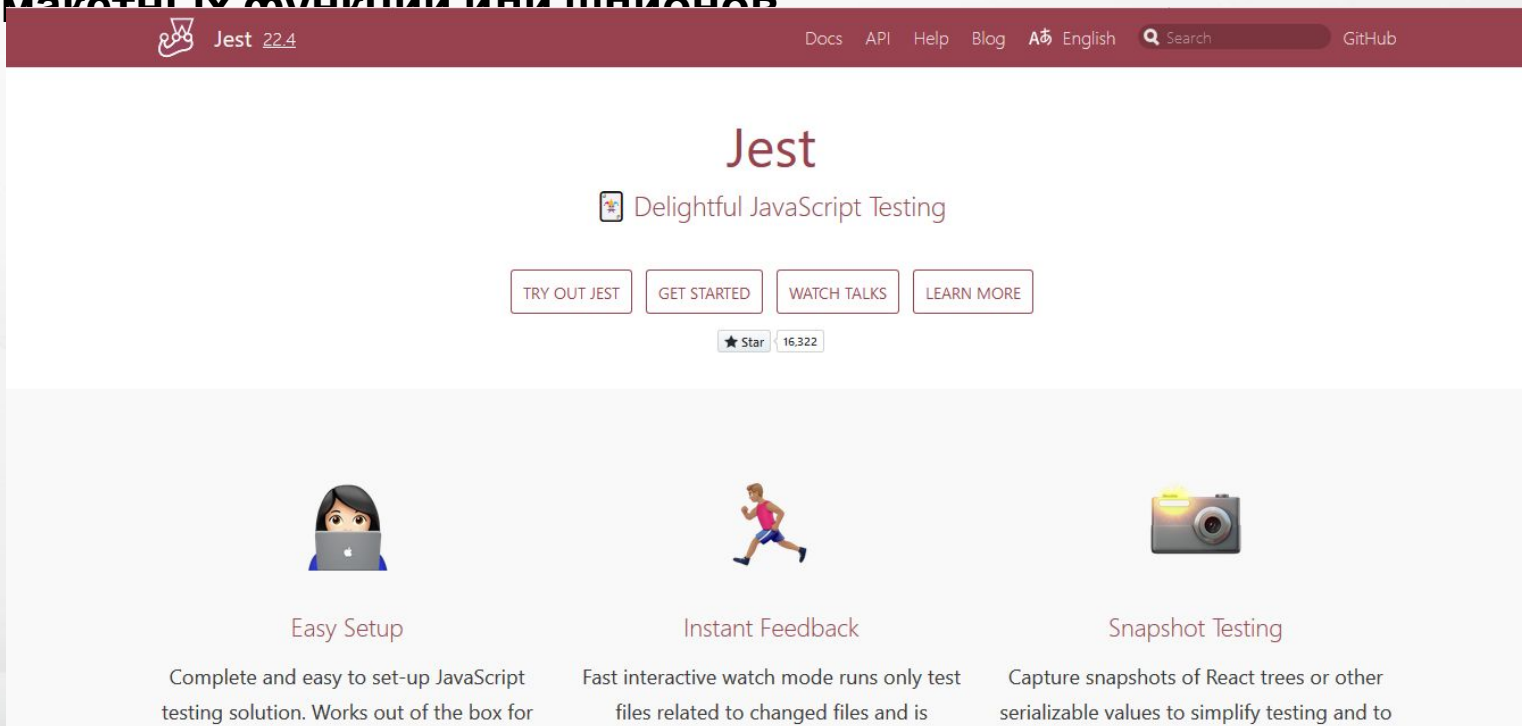


<https://www.npmjs.com/>

Введение в JavaScript

Инструмент тестирования Jest

Jest - это относительно новая платформа тестирования, созданная Facebook и хорошо принятая сообществом React. Самое большое преимущество использования Jest перед другими наборами тестов заключается в том, что вам требуется нулевая или минимальная конфигурация для начала написания тестов. Фреймворк имеет встроенную библиотеку утверждений и поддерживает использование макетных функций или шпионов.



The screenshot shows the Jest website homepage. At the top, there is a navigation bar with links for Docs, API, Help, Blog, a language selector set to English, a search bar, and a GitHub link. The main heading is "Jest" with the tagline "Delightful JavaScript Testing". Below this are four buttons: "TRY OUT JEST", "GET STARTED", "WATCH TALKS", and "LEARN MORE". A star icon indicates 16,322 stars on GitHub. The page features three columns of benefits, each with an emoji icon and a title:

- Easy Setup** (woman at laptop emoji): Complete and easy to set-up JavaScript testing solution. Works out of the box for
- Instant Feedback** (runner emoji): Fast interactive watch mode runs only test files related to changed files and is
- Snapshot Testing** (camera emoji): Capture snapshots of React trees or other serializable values to simplify testing and to

<https://facebook.github.io/jest/>

Jest имеет функцию под названием «snapshot testing», которая позволяет гарантировать, что пользовательский интерфейс приложения не изменится неожиданно.



Введение в JavaScript

Экосистема JavaScript

Mocha - это платформа тестирования JavaScript, которая имеет поддержку браузера, поддержку асинхронного обслуживания, включая перспективы, отчеты об охвате тестирования и JavaScript API для запуска тестов. Mocha часто соединяется с библиотекой утверждений, такой как Chai, should.js, expect.js, потому что у нее нет собственной библиотеки утверждений.



simple, flexible, fun

Mocha is a feature-rich JavaScript test framework running on [Node.js](#) and in the browser, making asynchronous testing *simple* and *fun*. Mocha tests run serially, allowing for flexible and accurate reporting, while mapping uncaught exceptions to the correct test cases. Hosted on [GitHub](#).

gitter join chat backers 64 sponsors 18

BACKERS

Find Mocha helpful? Become a [backer](#) and support Mocha with a monthly donation.

<https://mochajs.org/>



Экосистема JavaScript

Jasmine - это ориентированный на поведение фреймворк JavaScript. Жасмин стремится стать браузером, платформой и независимым от фреймворка набором тестов. У Jasmine есть своя собственная библиотека утверждений, называемая *matchers*, которая дает инструменту чистый и легко читаемый синтаксис. Jasmine не имеет встроенного тестового исполнителя, и вам, возможно, придется использовать общий тестовый исполнитель, например, Karma.

<https://jasmine.github.io/>



Введение в JavaScript

Литература

Интернет-ресурсы:

1. Современный учебник по Java Script - <https://learn.javascript.ru/books>
2. Выразительный JavaScript - https://www.gitbook.com/book/karmazzin/eloquentjavascript_ru/details
3. Ресурс с кратким изложением основ - <http://bonsaiden.github.io/JavaScript-Garden/ru/>

Печатные издания:

1. Kyle Simpson, You Don't Know JS: ES6 and Beyond, O'Reilly, 2015
2. <http://es6-features.org/>
3. Марк Бейтс. CoffeeScript. Второе дыхание JavaScript: пер. с англ. А.Кисилев. – М.: ДМК Пресс, 2012. – 312 с





**THANKS FOR
COMING**

See you soon