Федеральное государственное образовательное бюджетное учреждение высшего образования

«ФИНАНСОВЫЙ УНИВЕРСИТЕТ ПРИ ПРАВИТЕЛЬСТВЕ

РОССИЙСКОЙ ФЕДЕРАЦИИ»

**Колледж информатики и программирования**

ЛАБОРАТОРНАЯ РАБОТА №0

по МДК.02.01 Технология разработки программного обеспечения

по теме: Обзор библиотек javascript.

Выполнила:

студентка группы 4ИСИП-519

Низаметдинова.И.Н.

Проверил:

преподаватель Сибирев И.В.

Москва

2023

Основные библиотеки и фреймворки JavaScript

JavaScript существует уже более 20 лет, и это один из тех языков, которые никогда не перестают развиваться. В последнее время язык быстро развивается, что заставляет меня задаться вопросом, будут ли современные технологии JavaScript для начинающих, которые хорошо известны сегодня, актуальными через пару лет.

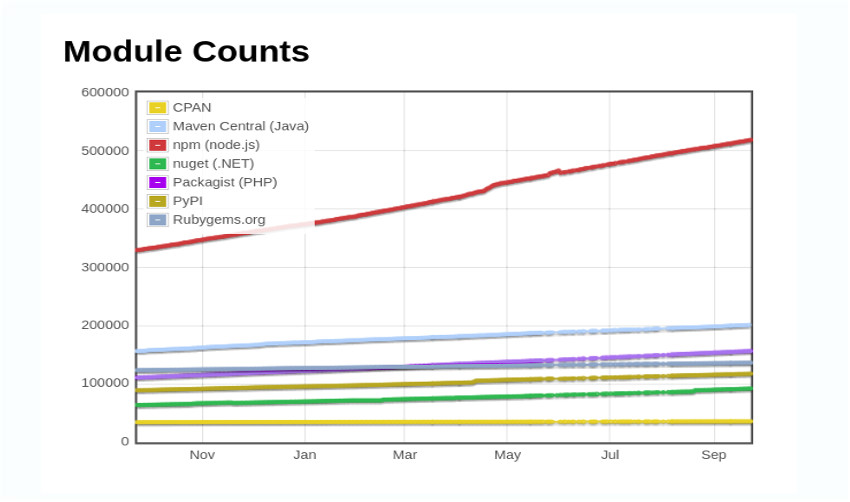
Тем не менее, важно быть впереди, работая с новейшими инструментами и фреймворками для улучшения рабочего процесса разработки.

Введение

Среда JavaScript стала огромной. Она имеет собственную экосистему библиотек, фреймворков, инструментов, менеджеров пакетов и новых языков, которые компилируются в JavaScript. Интересно, что npm, который является де-факто-менеджером пакетов для JavaScript, также является крупнейшим в мире реестром программного обеспечения. Вот выдержка из публикации, опубликованной на Linux.com еще в январе 2017 года.

*«В более чем 350 000 пакетов содержит реестр npm, что более чем вдвое больше, чем следующий наиболее популярный реестр пакетов (который является репозиторием Apache Maven). Фактически, в настоящее время это самый крупный реестр пакетов в мире.»*

Перенеситесь вперед на восемь месяцев, и в настоящее время в реестре npm есть около 500 000 пакетов. Это огромный рост по сравнению с другими репозиториями пакетов.



Будучи разработчиком JavaScript на начальном этапе, важно не отставать от современных инструментов и библиотек JavaScript. Когда технология становится популярной, спрос на нее высок, что, в свою очередь, означает больше заданий по написании кода, где платят самую высокую зарплату в отрасли.

Библиотеки

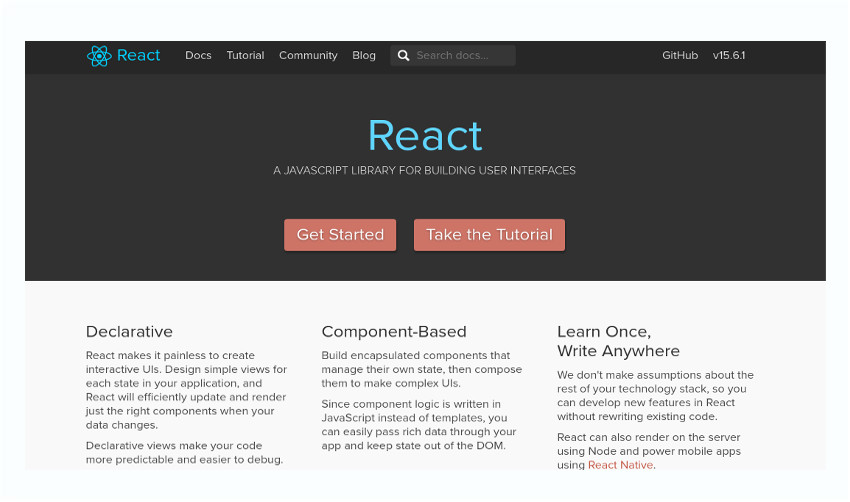
Библиотека является многоразовой частью кода, которая предлагает определенные функции. Это набор функций, объектов и классов, которые вы можете использовать в своем приложении. Библиотека абстрагирует различные слои, так что вам не нужно беспокоиться об их деталях реализации.

Вы можете вызвать библиотечную функцию и передать ей некоторые параметры, и библиотека выполнит ее и вернет вам элемент управления. Тем не менее, это не создает каких-либо структурных ограничений, которые ограничивают использование библиотеки. Популярные библиотеки JavaScript включают:

React

React — это библиотека JavaScript, созданная разработчиками Facebook и Instagram. Согласно опросу Stack Overflow Survey 2017, React был признан самой популярной технологией среди разработчиков. React также имеет честь быть самым популярным проектом JavaScript, согласно количеству звезд на GitHub.

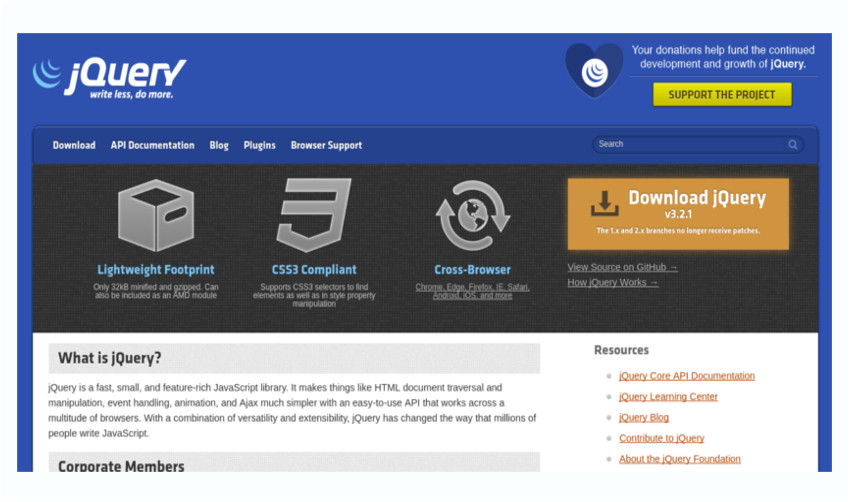
Так почему React получает все внимание? С помощью React можно создать интерактивный интерфейс с использованием декларативного подхода, в котором вы можете контролировать состояние приложения, говоря: «Представление должно выглядеть так». Он использует компонентную модель, в которой компоненты являются повторно используемыми элементами пользовательского интерфейса, и каждый компонент имеет свое собственное состояние.



React использует виртуальную модель DOM, так что вам не нужно беспокоиться о прямом манипулировании с DOM. Другие примечательные особенности React включают однонаправленный поток данных, дополнительный синтаксис JSX и инструмент командной строки для создания проекта React с нуля.

JQuery

jQuery - это библиотека, которая сделала JavaScript более доступным а DOM-манипуляцией проще, чем раньше. Плавная кривая обучения jQuery и простой синтаксис породили на стороне клиента новое поколение новых разработчиков. Несколько лет назад jQuery считался прочным решением для создания надежных веб-сайтов с поддержкой кросс-браузерности. Основные функции jQuery, такие как манипулирование DOM на основе селекторов CSS, обработка событий и создание вызовов AJAX, подпитывали его популярность.



Однако все изменилось, и среда JavaScript постоянно развивается. Некоторые функции jQuery были включены в новую спецификацию ECMAScript. Более того, новые библиотеки и фреймворки, используемые сегодня, имеют собственный способ связывания DOM, и поэтому простые методы манипуляции с DOM больше не требуются.

D3: Документы, управляемые данными

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.

D3 работает, связывая данные с DOM и затем преобразуя их в документ. Она также имеет собственную экосистему, которая состоит из плагинов и библиотек, которые расширяют ее базовые функциональные возможности. Библиотека существует с 2011 года.



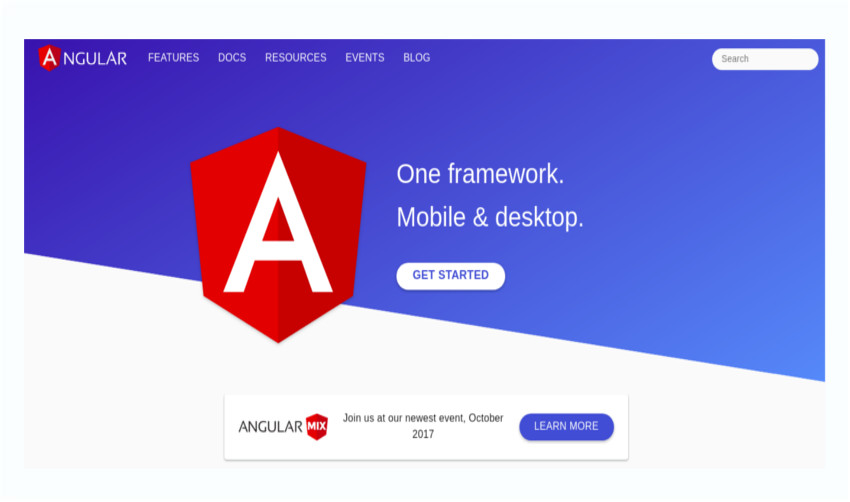
Фреймворки

Фреймворк имеет архитектуру, которая диктует поток управления в вашем приложении. Фреймворк описывает скелет и рассказывает вам, как все должно быть организовано. Вам также предоставляются основные функции, необходимые для запуска вашего приложения. Кроме того, вы обязаны следовать принципам и шаблонам фреймворка. Разница между фреймворком и библиотекой заключается в том, что вы вызываете библиотеку, тогда как фреймворк вызывает вас.

Фреймворк часто включает в себя множество библиотек и имеет более высокий уровень абстракции. Функциональность, такая как обработка событий, создание вызовов AJAX, привязка шаблонов и данных и тестирование, уже встроены в фреймворк.

Angular

AngularJS когда-то была самой популярной технологией JavaScript среди разработчиков интерфейсов. Он был поддержан Google и сообществом частных лиц и корпораций. Несмотря на популярность, у AngularJS была своя доля недостатков. Команда Angular провела два года работы над новой версией Angular, которая была наконец выпущена в сентябре 2016 года.



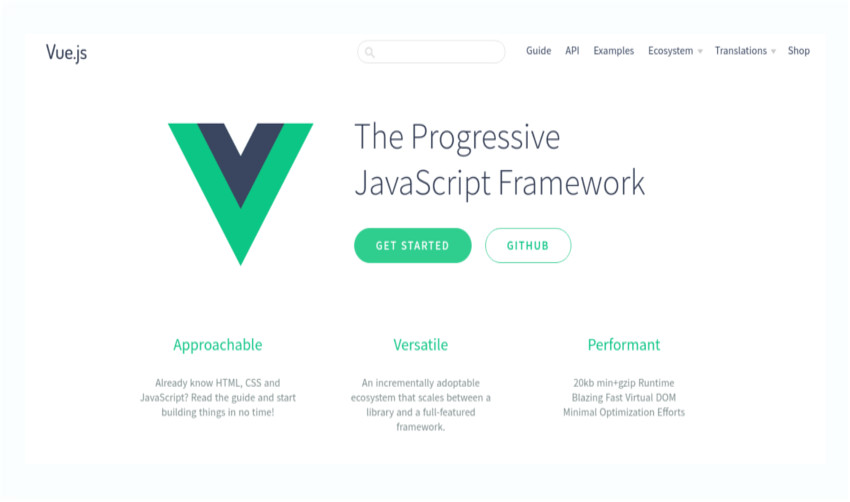
Выпуск Angular 2 был переделкой AngularJS. Некоторые из функций Angular 2 включают:

* TypeScript над JavaScript как язык по умолчанию
* компонентная архитектура
* улучшенная производительность как на мобильных, так и на веб-платформах.
* лучшие инструменты и варианты исходных каркасов

Тем не менее, модернизация от Angular 1.x до Angular 2 является дорогостоящей, потому что Angular 2 - совершенно другой зверь. Это одна из причин, почему Angular 2 не был так сразу же принят, как свой предшественник. Но Angular и AngularJS по-прежнему относятся к числу наиболее часто используемых технологий в соответствии со Stack Overflow (2017). Проект Angular имеет около 28 000 звезд в GitHub.

Vue.js

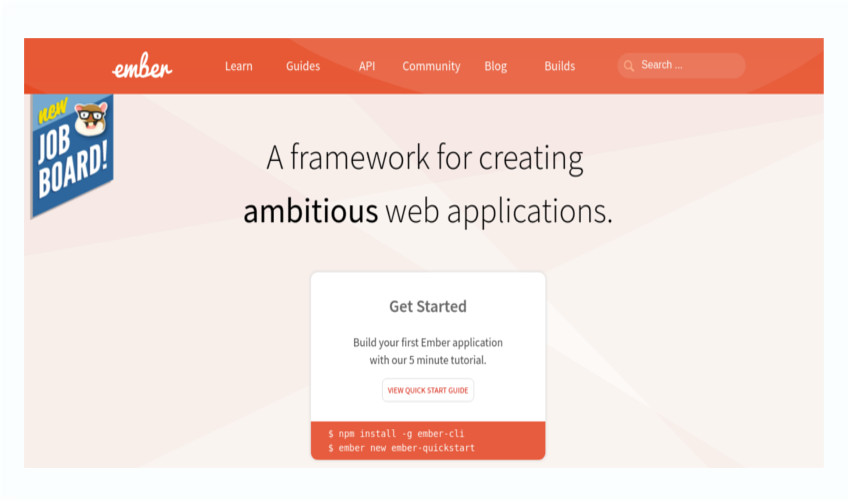
Vue.js - это легкий JavaScript фреймворк, который активно развивается в этом году. Это самый популярный JavaScript фреймворк на GitHub с точки зрения звезд GitHub. Vue утверждает, что это менее упрямый фреймворк и, таким образом более удобный в использовании. Синтаксис шаблона на основе HTML Vue связывает отрисовываемый DOM с данными экземпляра.



Фреймворк предлагает опыт, похожий на React, с его виртуальными DOM и компонентами повторного использования, которые можно использовать для создания как виджетов, так и целых веб-приложений. Кроме того, вы также можете использовать синтаксис JSX для непосредственного написания функций рендеринга. Когда состояние изменяется, Vue.js использует систему реактивности, чтобы определить, что изменилось и перерисовывает минимальное количество компонентов. Vue.js также поддерживает интеграцию других библиотек во фреймворк без особых хлопот.

Ember.js

Ember.js является фронт-енд фреймворком на основе шаблона Model-View-ViewModel (MVVM). Фреймворк использует стандартный подход к конфигурации, который популярен среди серверных фреймворков таких как Ruby on Rails и Laravel. Ember.js включает в себя общие идиомы и лучшие практики фреймворков, чтобы вы могли получать приложение без особых усилий.

Стек Ember обычно включает:

* Ember CLI: предоставляет основные варианты каркаса приложения и поддерживает сотни надстроек.
* Ember Data: библиотека сохранения данных, которая может быть настроена для работы с любым сервером.
* Ember Inspector: расширение доступное для Chrome и Firefox.
* Liquid Fire: дополнение для переходов и анимаций.

Инструменты

Инструмент представляет собой набор подпрограмм, которые помогут вам в процессе разработки. В отличие от библиотеки, инструмент обычно выполняет задачу над клиентским кодом. Он берет ваш код в качестве входных данных, выполняет на нем задачу, а затем возвращает результат. Обычно используемые инструменты включают в себя транспайлеры и инструменты для сборки, сборщики ресурсов и другие.

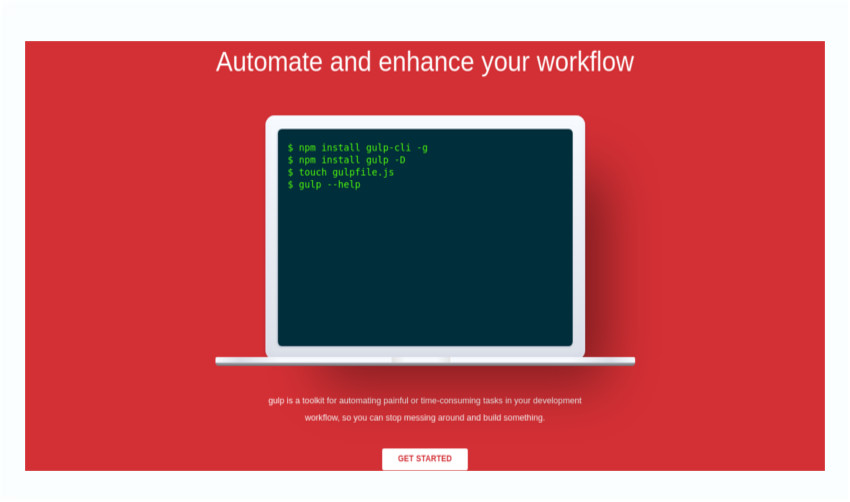
Инструменты: выполнение общих задач

Инструменты, выполняющие общие задачки - это инструменты, используемые для автоматизации определенных повторяющихся задач. В число самых популярных входят:

Gulp

Gulp - это набор инструментов JavaScript, который используется как менеджер задач и как система сборки в веб-разработке. Компиляция, минимизация кода, оптимизация изображений, модульное тестирование, линтинг и т.д. - это повторяющиеся задачи, которые должны быть автоматизированы. Gulp упрощает процесс написания задач даже для людей, которые менее знакомы с JavaScript.

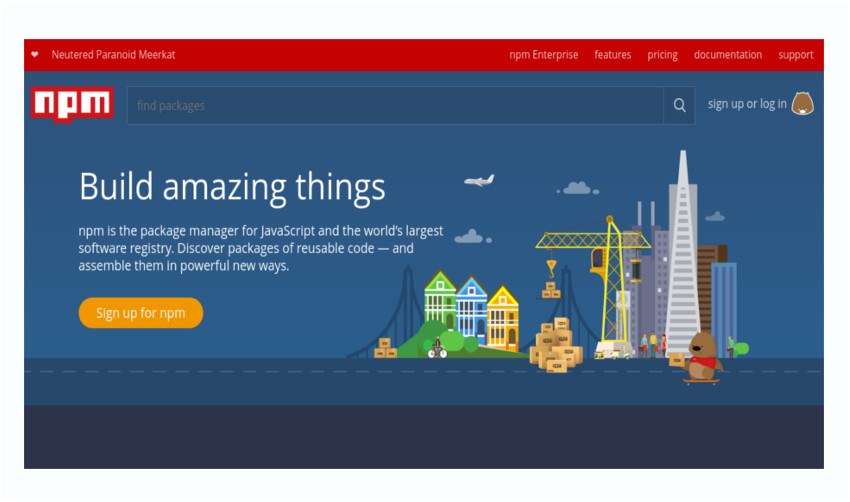
Gulp использует конвейеры для потоковой передачи данных из одного плагина в другой, и конечный результат выводится в папку назначения. Gulp работает лучше по сравнению с Grunt, потому что он не создает временные файлы для хранения промежуточных результатов, что приводит к меньшему количеству вызовов ввода-вывода.

Grunt

Grunt - это менеджер задач и инструмент автоматизации JavaScript. Grunt имеет интерфейс командной строки, который позволяет запускать настраиваемые задачи, определенные в файле, называемом Gruntfile. У Grunt есть тысячи плагинов на выбор, которые должны охватывать большинство обычных повторяющихся задач, с которыми вы столкнулись. С помощью Grunt вы можете запускать все задачи под одной командой, что упрощает вашу жизнь.

npm

Gulp и Grunt требуют, чтобы вы потратили время на изучение и освоение нового инструмента, что требует времени. Ввод дополнительных зависимостей в ваш проект можно избежать, выбирая альтернативу, которая уже связана с Node.js. Хотя npm более известен как менеджер пакетов, сценарии npm можно использовать для выполнения основной части вышеупомянутых задач.



Инструменты: тестирование

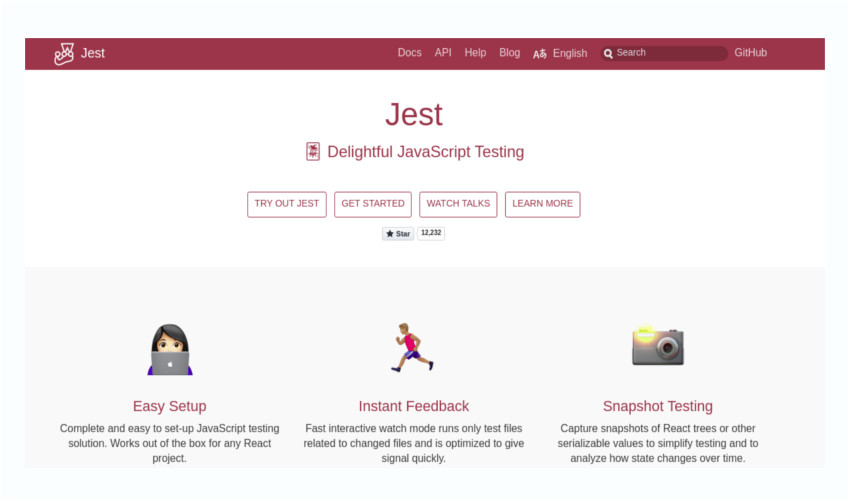
Тестирование - это процесс проверки и валидации того, что приложение соответствует ожидаемым бизнес и техническим требованиям. Подход, основанный на тестах, также помогает обнаруживать ошибки и, следовательно, его следует рассматривать как неотъемлемую часть современного процесса разработки.

Jest

Jest - это относительно новый фреймворк для тестирования, созданный в Facebook и хорошо принятый сообществом React. Существует распространенное заблуждение, что Jest специально разработан для работы с React; однако, согласно документации Jest:

*Несмотря на то, что Jest можно рассматривать как фреймворк тестирования специфичный для React, на самом деле это универсальная платформа тестирования, способная адаптироваться к любой библиотеке или фреймворку JavaScript. Вы можете использовать Jest для тестирования любого кода JavaScript.*

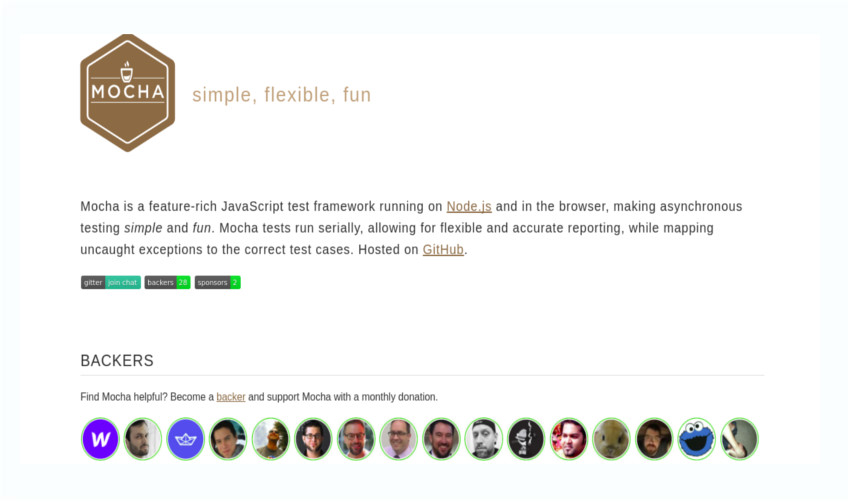
Самое большое преимущество использования Jest над другими наборами тестов заключается в том, что вам требуется нулевая или минимальная конфигурация для того, чтобы начать писать тесты. Фреймворк имеет встроенную библиотеку утверждений и поддерживает использование мок функций или шпионов.



Jest имеет функцию под названием snapshot testing, которая позволяет гарантировать, что пользовательский интерфейс приложения неожиданно не изменится. Разработчики в Facebook и другие участники недавно вложили много работы в этот проект, поэтому не удивительно, что Jest окажется самой популярной тестовой платформой для JavaScript в ближайшие годы.

Mocha

Mocha - это фреймворк для тестирования JavaScript, который предоставляет поддержку браузера, асинхронные вызовы, включая обещания, отчеты о тестовом покрытии и JavaScript API для запуска тестов. Mocha часто используется с библиотекой утверждений, такой как Chai, should.js, expect.js или better-assert, потому что у нее нет собственной библиотеки утверждений.



Jasmine

Jasmine - это фреймворк для тестирования JavaScript, ориентированный на поведение. Jasmine стремится стать браузером, платформой и независимым от платформы набором тестов. У Jasmine есть своя собственная библиотека утверждений, называемая matchers, которая дает ему чистый и легко читаемый синтаксис. Jasmine не имеет встроенной команды для выполнения тестов, и вам, возможно, придется использовать какую-нибудь общую, например, Karma.



Вывод

JavaScript, язык Интернета, остался актуальным со времен его создания еще в 1995 году. Вероятно, он останется таким же, пока браузеры не решат отказаться от него в пользу какого-нибудь другого языка. Несмотря на то, что существует множество других языков, которые скомпилируются с использованием JavaScript, нет другого языка сценариев, который заменит JavaScript в обозримом будущем. Почему? Потому что JavaScript стал слишком популярным, чтобы его можно было заменить.

Среда JavaScript определенно развивается, что видно из текущих тенденций в веб-разработке. Старые библиотеки и фреймворки были заменены новыми технологиями. jQuery, которая когда-то была наиболее популярной библиотекой JavaScript, испытывает спад с точки зрения привлекательности, использования и популярности. Новое поколение фронт-енд библиотек, фреймворков и инструментов набирает силу и получает всеобщее признание.

Привыкание к новым тенденциям в технологии также имеет свои преимущества. Задачи по написанию кода, требующие знания React, имеют одни из самых высоких зарплат в отрасли со средней зарплатой в размере 105 000 долларов США в соответствии с Stack Overflow (2016). Поэтому вам нужно продолжить обучение и экспериментировать с новейшими инструментами и фреймворками, чтобы максимально использовать JavaScript.