



Введение в Vue.js

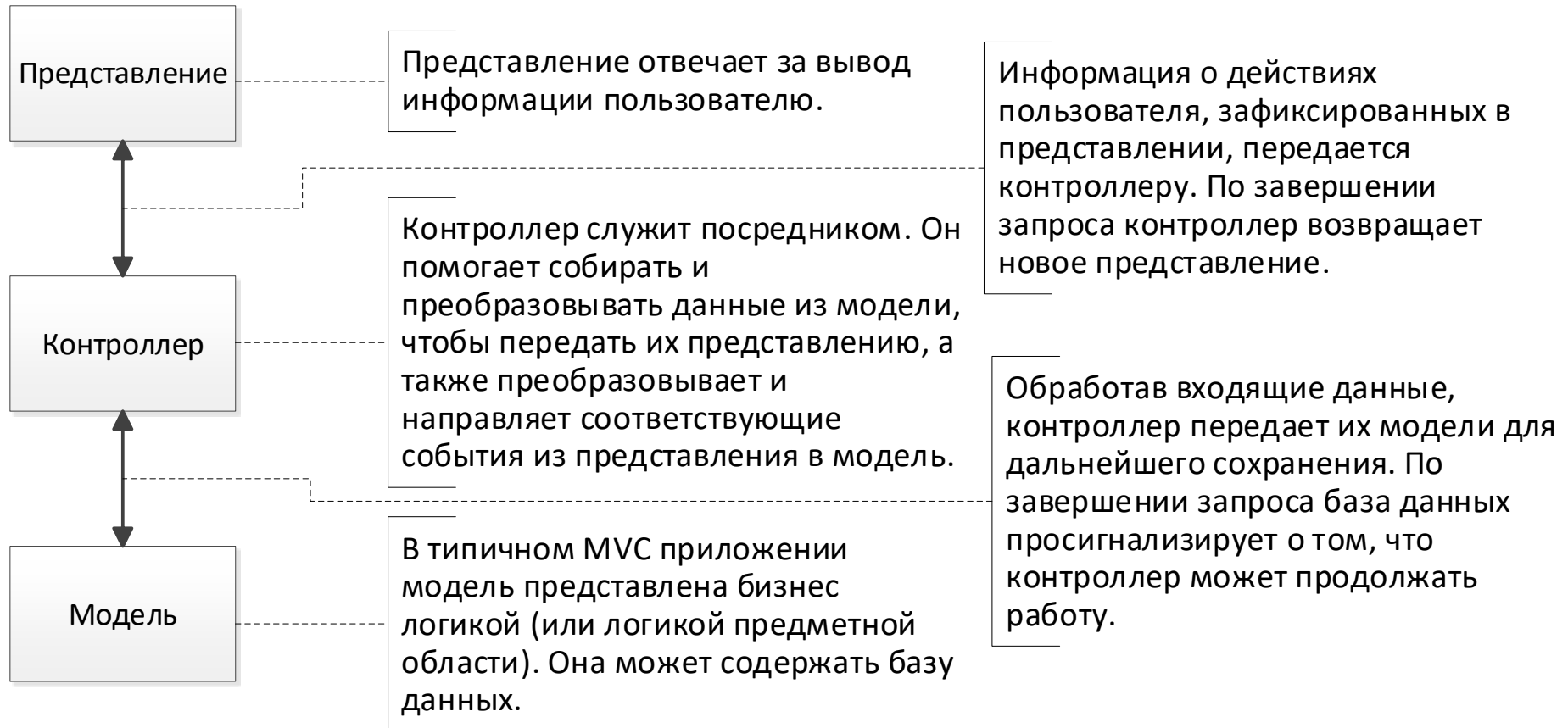
Шаблон проектирования MVC

Клиентский шаблон проектирования «модель - представление - контроллер» (Model - View - Controller, MVC).

MVC применяется для разделения ответственности в приложении. Представление отвечает за вывод информации пользователю - это графический пользовательский интерфейс (graphical user interface, GUI). В центре находится контроллер. Он помогает передавать события из представления в модель, а данные - из модели в представление.

Бизнес логика может находится как в модели, так и в контроллере, так и частично в представлении.

Шаблон проектирования MVC



Шаблон проектирования MVVM

Шаблон MVVM (Model - View - ViewModel - «модель - представление - модель представления») - это следующий этап в развитии MVC. Его отличительными чертами являются модель представления (ViewModel, VM) и связывание данных.

MVVM позволяет создавать веб-приложения, которые мгновенно реагируют на действия пользователя и позволяют свободно переходить от одной задачи к другой.

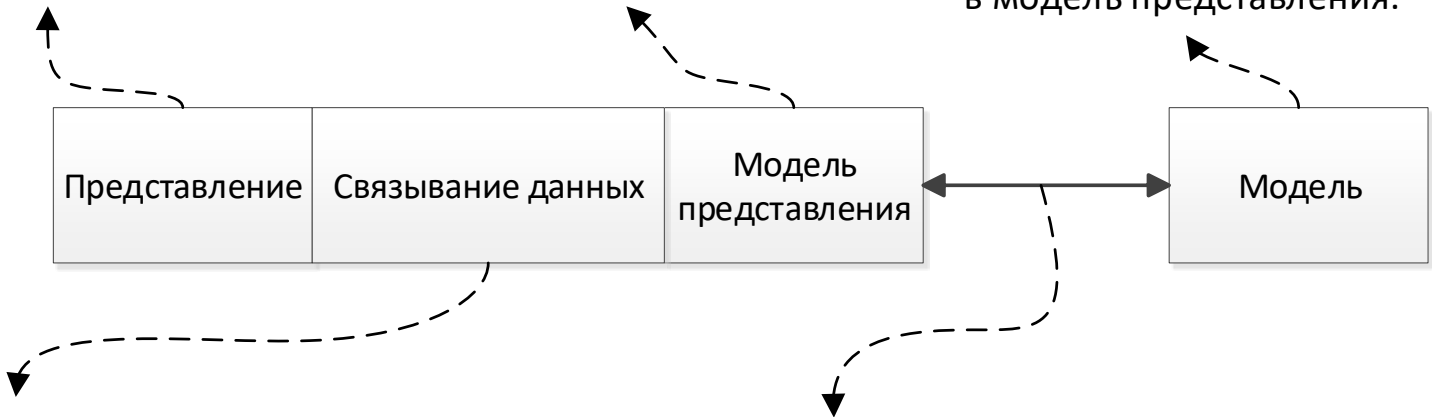
Фундаментальная особенность: при изменении данных внутри VM автоматически обновляются все представления, связанные с этой моделью.

Шаблон проектирования MVVM

Представление по-прежнему отвечаем за то, что происходит на экране, но вся логика принятия решений переместилась в модель представления. Само представление лишь выводит содержимое в зависимости от наличия и количества данных в текущем состоянии приложения.

Модель представления хранит данные приложения в объекте, известном как хранилище. В хранилище находятся все данные, которые могут потребоваться приложению в любой момент времени (то есть состояние приложения).

Модель по-прежнему играет роль постоянного репозитория для данных приложения. Некоторые архитектуры в мире JavaScript используют модель исключительно в качестве хранилища, не налагая никаких ограничений на входящие данные; при этом вся логика принятия решений выносится в модель представления.



Механизм связывания делает данные доступными для вывода в виде свойств. Он также предоставляет методы для работы с данными в хранилище VM, которые вызываются представлениями.

По аналогии с контроллером модель представления отвечает за сохранение данных в модель. Однако подобные транзакции не должны быть синхронными, благодаря чему пользователь могут продолжать работать с приложением.

РЕАКТИВНОЕ ПРОГРАММИРОВАНИЕ

Реактивное программирование — парадигма программирования, ориентированная на потоки данных и распространение изменений.

В императивном программировании присваивание $a = b + c$ будет означать, что переменной a будет присвоен результат выполнения операции $b + c$, используя текущие (на момент вычисления) значения переменных. Позже значения переменных b и c могут быть изменены без какого-либо влияния на значение переменной a .

В реактивном программировании значение a будет автоматически пересчитано, основываясь на новых значениях переменных b и c .

Источник:

https://ru.wikipedia.org/wiki/Реактивное_программирование

РЕАКТИВНОЕ ПРОГРАММИРОВАНИЕ

Однако в веб-приложениях парадигму реактивного программирования начали использовать относительно недавно, в основном благодаря наличию фреймворков Vue, React и Angular.

Чтобы веб-приложение можно было назвать реактивным, оно должно выполнять следующие функции:

- ❖ отслеживать изменения в своем состоянии;
- ❖ распространять уведомления об изменениях для всех своих компонентов;
- ❖ автоматически отрисовывать представления в ответ на изменения состояния;
- ❖ своевременно реагировать на действия пользователей.

ЧТО ТАКОЕ Vue.js?

Vue.js — фреймворк, написанный на языке JavaScript, с открытым исходным кодом для создания пользовательских интерфейсов в парадигме реактивного программирования.

Первый релиз фреймворка увидел свет в феврале 2014 года. Его текущей версией >2.

Vue.js поддерживается всеми браузерами, которые совместимы с ECMAScript 5 (это все современные браузеры).

Официальный сайт фреймворка - <https://vuejs.org/>

Руководство к Vue.js на русском языке:

<https://ru.vuejs.org/v2/guide/>

ПОЧЕМУ Vue.js?

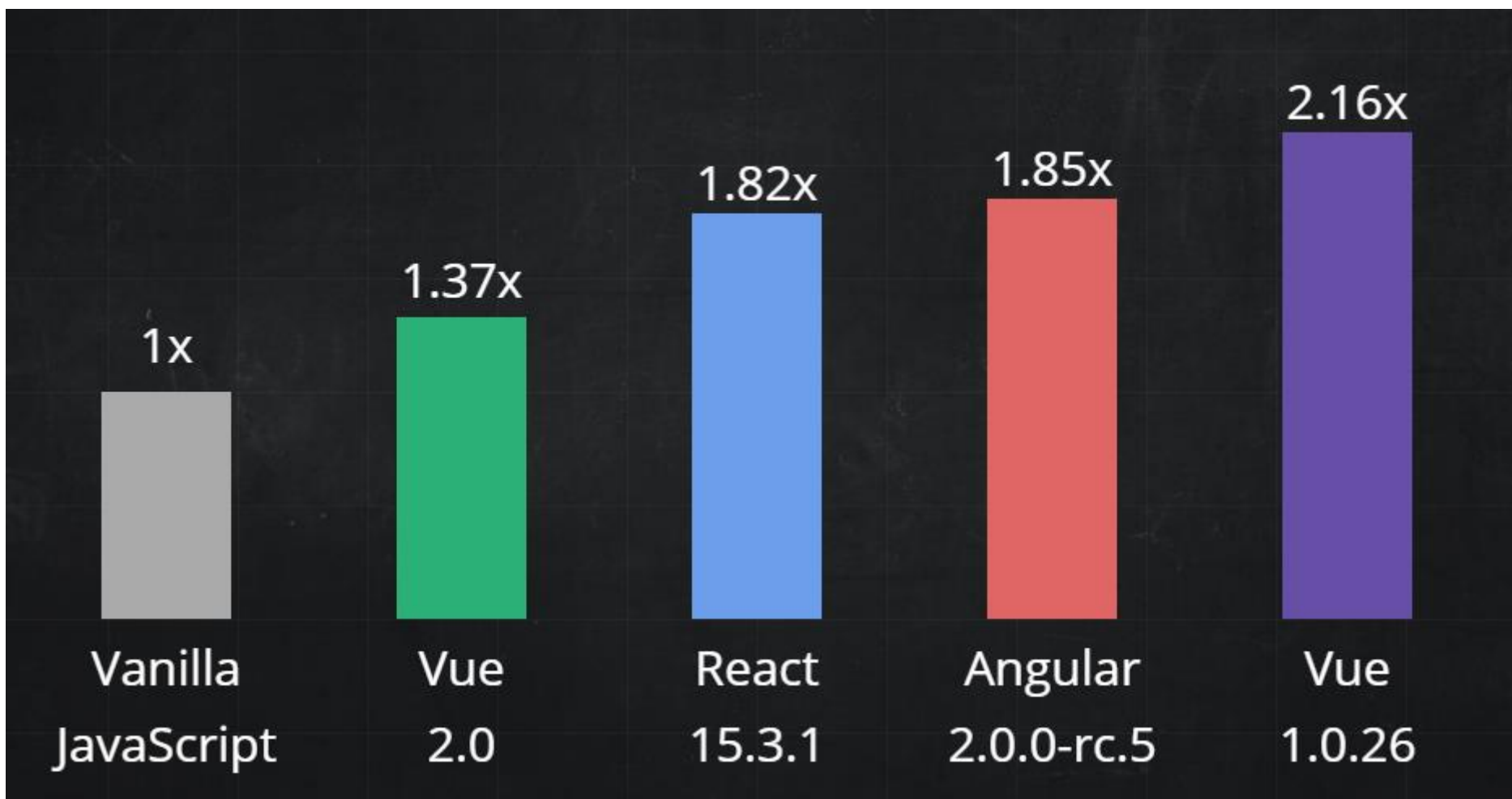
Vue.js имеет небольшой размер:



Источник: <https://habrahabr.ru/post/329452/> 9

ПОЧЕМУ Vue.js?

Vue.js обладает хорошей производительностью



Источник: <https://habrahabr.ru/post/329452/> ¹⁰

ВИРТУАЛЬНЫЙ DOM Vue.js

Для взаимодействия с DOM применяется JavaScript. Но когда мы пытаемся манипулировать большим количеством html-элементами с помощью JavaScript, то мы можем столкнуться со снижением производительности.

Виртуальный DOM представляет легковесную копию обычного DOM. Если приложению нужно узнать информацию о состоянии элементов или их изменить, то происходит обращение к виртуальному DOM. А потом в случае необходимости Vue выбирает минимальный набор компонентов, для которых надо выполнить изменения на веб-странице, чтобы реальный DOM соответствовал виртуальному. Благодаря виртуальному DOM повышается производительность приложения.

ПОЧЕМУ Vue.js?

Другие критерии:

- ❖ Библиотека Vue.js не требует никаких специальных знаний. Любое Vue-приложение пишется с помощью HTML, CSS и JavaScript. Используемый реактивный подход упрощает написание программ и работу с ними. Уменьшается объем шаблонного кода, связанного с интерфейсом. Отсутствие необходимости кодировать все взаимодействия с пользователем позволяет сосредоточиться на моделировании данных и внешнем виде страницы, а также связывании этих двух аспектов, что благодаря Vue.js очень просто сделать.
- ❖ При внедрении Vue.js в существующие проект, не мешает ранее написанным компонентам, используемым библиотекам и другим фреймворкам.
- ❖ Не содержит сложных инструментов сборки. Разработка прототипа от начала до конца занимает одну-две недели.
- ❖ Позволяет реализовывать уникальные и нестандартные возможности. Vue-приложения изначально проектируются модульными и расширяемыми. Их компоненты поддерживают многократное использование, наследование и добавление новых функций с помощью примесей.
- ❖ Vue.js легко интегрируется с многими популярными фреймворками для сборки (Webpack и др.) и тестирования (Jasmine и др.), а значит может быть интегрирован в существующий процесс сборки, тестирования и развертывания.
- ❖ Обширное и активное сообщество Vue.js, русскоязычная документация и расширяющаяся экосистема.

ДОПОЛНИТЕЛЬНЫЕ ССЫЛКИ

Сравнение Vue.js с другими фреймворками:

<https://ru.vuejs.org/v2/guide/comparison.html>

Руководство по Vue.js (онлайн учебник)

<https://metanit.com/web/vuejs/>