Learn Vue 2: Step By Step

Vue является одним из самых ярких дополнений к миру front-end за все эти годы. С его интуитивно понятным API, и тот факт, что он может быть применен к любому типу приложений, это не удивительно, почему люди тяготели к нему так сильно.

Если вы хотите запрыгнуть на борт, позвольте мне показать вам, шаг за шагом, то, что вам нужно знать. Это лучший способ, чтобы узнать Vue!

# [Learn Vue 2: Step By Step:](https://laracasts.com/series/learn-vue-2-step-by-step)Setup Vue Devtools

**EPISODE 2 • RUN TIME 3:03**

На шаге 1 мы изучили привязку данных/и вы впервые взглянули на реактивнсоть которая работатет прямо здесь/что довольно круто/однако/я открою сhrome dev tools выбрав inspect или если вы на маке вы можете нажать shift command c/окей и вы видите эту небольшую подсказку/загрузите vue dev tools для получения лучшего опыта разработки/и почему бы нам не последовать этому совету и не подтянуть это прямо сейчас?/если мы взглянем на это/это является простоым расширением хрома/и мы можем получить его в веб магазине/добавить в хром и все готво/но теперь если я обновлюсь/я все еще не вижу никакиех изменений/и если я кликну на иконку vue это говорит нам что vue js не определен/и это потому что мы испольуем ссылку на файл/мы могли бы запустить небольшой сервер или мы можем просто дать доступ этому расширению к нашему файлу/manage extentions/vue js - предоставить доступ к файлу по ссылке/окей и теперь когда мы обновимся – мы видим что иконка vue изменилась/и давайте прикроем это окно и откроем снова и вот это здесь/и теперь мы видим новую vue вкладку/окей это действительно клево и вам должно быть это понравится/заметьте у нас тут есть корневой элемент где мы имеем один кусок связанных данных – это свойство message/но вот на что я хочу обратить ваше внимание/если я начну менять это/значение которое здесь будет также динамически обновляться/и вот главная вещь которую вам нужно понять в отношении реактивнсоти/любые изменения тут/имею ввиду обновление данных в поле ввода – данные ниже этого обхекта бубудт также обновляться/однако/в точности тоже самое будет происходить в обратном случае/посмотрим/вы видите что этот root был cохраненм в этой временной переменной названной vm ноль/поэтому если я перейду в консоль и напечатаю это/вот/у нас есть парамеры экземпляра/и что еще лучше vue ,eltт автоматически определеять его свойства/так если я например хочу получить доступ к свойству message – вот пожалуйста/но теперь давайте попробуем изменить значение/я напишу я была изменена/и сохраним это/и тут же это обновится в поле ввода/и это то что я называю двухсторонней реактивностью/если я изменил это на уровне даных – это автомаически имениется в представлении/если я изменил это во vue это тут же изменится внутри нашего обхекта данных/это что называется – единственный источник достоверности/взгляните на это – это называется - достоверностью/то чему равен data/и затем просто изменяя это значение это автоматичиески отразится в dom/сравните это с тем как это было в прошлом/например где вы использовали jquery/где для отслеживания вещей вы постоянно обращались к dom/yfйти этот параграф/подтянуть эти данные и затем изменить на что то другое/и затем некоторое время спустя если вам нужно это сделать снова – опять вы обращались к dom получали это и затем изменяли/это очень неудобный способ строить ваши приложения/но с vue или схожим фреймворком – да – мы предоставляем data быть единственный достоверным источником/я надеюсь вы взволнованы и мы продолжим на следующем шаге/

/

Давайте поскорей установим Vue Devtools. Как вы видите, это открывает нам портал к любому экземпляру Vue на странице. Особенно на ранних стадиях, вы получите много пользы от этого расширения Chrome

/

# [Learn Vue 2: Step By Step:](https://laracasts.com/series/learn-vue-2-step-by-step)Components Within Components

**EPISODE 8 • RUN TIME 2:36**

В нашем последнем эпизоде вы создали ваш первый vue компонент/но в настящий момент это просто обертка для элемента списка/так что это не очень полезно/но это позволило нам поговорить о начальной структуре и использовани vue конмонентов/вещи становятся привлекательнее когда вы ссылаетесь на один компонент внутри другого/для примера прямо сейчас у нас есть одиночная задача task /но вы также могли бы иметь список задач/что то вроже этого/и теперь этот компонент список задач будет отвечать за создание под задач/давайте посмотрим/сейчас у нас етсь задача но мы можем создать еще лист задач/и пока давайте захардкодим некоторые данные тут и затем мы поговорим о получении их из вне/или даже о предеачи их как свойства/tasks будет массивом задач/сейчас наш шаблон будет изменен/мы могли бы сказать/я хочу получить элемент задача потому что он был создан прямо тут внизу/но сколько мне нужно? Мне нужен элемент под каждую задачукоторая у меня тут есть/и это свучит так как будто нам нужно использовать v-for/v-for задачу в задачах/и затем нам нужно видеть текст/и это начало выглядеть неплохо/но у нас тут есть одна проблема/давайте перейдем в хром и посмотрим получится ли у него это подтянуть/да я ничего тут не вижу/и если я открою dev tools/то увижу что я не могу использовать v-for компонент на переменном элементе потому что это отображает несколько элементов/и вот что тут важно понимать/когда бы вы не использовали шаблон в компоненте это всегда должно иметь единственный корневой элемент/и в этом случае так как мы используем v-for мы не имеем единственный корневой элемент/у нас тут целая куча близнецов/и в таких случая вы просто можете обернуть все это в div/и между прочем/если вы используете es6 вы можете использовать синтакс шаблонной строки вроде – ковычка открыли/ и закрыли/хорошо/если мы теперь вернемся обратно и обновим то увидим такой результат/список задач/но как это отличается от того что было пару эпизодов назад?ну /тогда мы испольховали неупорядоченный список и элементы списка но теперь у нас полноценный список задач/если мы когда либо заходим его использовать мы просто образаемся к нему/он не содержит элементы списка но содержит отдельные задачи

**Удалить последние 3 секнды**

/

Теперь мы можем не только создавать отдельные компоненты, но и также, можем размещать другие компоненты внутри них как дочерние. В этом эпизоде давайте рассмотрим основной пример такого подхода.

/

**[EPISODE 28](https://laracasts.com/series/learn-vue-2-step-by-step/episodes/28)● RUN TIME 2:30**

#### [Vue Filters](https://laracasts.com/series/learn-vue-2-step-by-step/episodes/28)

Сейчас давайте пройдем дальше и обсудим фильтры в этом эпизоде/мы переходим обратно в наш home vue/и как вы тут видите мы разместили postedON функцию/так мы добавили статус и подтянули дату создания и пропустили это через ,иблиотеку moment чтобы затем мы могли отформатировать это/и если мы вернемся обратно/мы в общем можем сказать что назначение этой фукнции это отформатировать значение внутри нее/но как оказалось мы вместо этого можем использовать синтекс фильтер/вроде - добавим статус create\_at и затем мы может добавить фильтер /это может быть что угодно/дата/относительная дата/или как давно/и затем мы можем определить это как глобально так и в каждом отдельном компоненте/это имеет смысл если применяемый фильтр имеет значение только для отдельного компонента/окей/мы можем уставить фильтр – как давно/и это будет принимаеть значение которое вы передадите/в нашем случае мы хотим чтоы это принимало дату создания/просто назовем это date/jrtй/сейчас мы просто форматируем это/мы вырежем и вставим сюда/и зменим это/что значит что я могу избавиться от этого/давйте сохраним это/скомпирлируем/упс/мы пропустили запятую тут/вт атк лучше/и вот что это делает/мы используем этот символ чтобы указать на фильтер в который должна быть передана эта строка/и если вы думаете о фильтрах – проще говоря думайте об этом как об одиночных функциях/что значит что вы можете использовать их столько сколько хотите/взять эту строку/передать это в фильтер и затем передать это в еще один фильтер если вам нужно/поэтому если вы передете в хром/

/удалить laracasts demo c экрана/

и обновитесь ы получите тоже самое что и было/но да/фильтры наиболее полезны для конкретно этой задачи/и как я сказал если вы хотите продолжить и например хотите сделать все заглавными символами/вы опять же определяете фильтр тут и теперь я хочу вернуть значение в to upper case/и это дожно сработать/и теперь у нас есть месяц назад/но в заглавных буквах/достаточно полезно/правда? И вот это все что касается фильтров/так в подобных случаях вы используете этот символ и вы можете добавлять сколько хотите/далее для каждого фильтра который вы определили/это должно быть размещено внутри компонента filters/и это все

/

В предыдущем эпизоде мы ссылались на простую функцию для отображения относительной даты для каждого потока. Однако, Vue фильтры больше подходят для основных операций и манипуляции над строками таких как эта.

/