[ES2015 Crash Course:](https://laracasts.com/series/es6-cliffsnotes)

Если вы чувствуете себя подавленным молниеносным темпом развития сообщества JavaScript, то вы не одиноки. JavaScript и jQuery 2009 года улучшились настолько, что даже отдаленно не сравнятся с современным функционалом. В этой серии я дам вам интенсивный курс по самому современному и лучшему, что этот язык может вам предложить.

[ES2015 Crash Course:](https://laracasts.com/series/es6-cliffsnotes)ES6 Compilation With Laravel Elixir

С возвращением!/ в нашей предыдущей серии мы установили babel и мы обсуждали преобразования и то как заставить это работать для разных браузеров/но сейчас как это все будет выглядеть если вы строите laravel проект?окей я покажу вам начальную установку elixir и мне кажется вам это понравится/и мы могли бы сказать/ laravel new es 6 project и как вы наверно знаете если вы используете laravel я также рекоммендую вам использовать laravel elixir который является чем то вроде оболочки вокруг gulp для упрощения выполнения всех этих задач/поэтому если я cat наш package json aайл вы увидите что мы имеет laravel elixir среди зависимостей/поэтму давайте устатновим все это/окей все так/и вот тетперь если мы взглянем на это/в моем gulp файле я напишу mix.browserfy и укажу ссылку на main js файл и теперь если я попытаюсь выполниь это/ ну это не сработает потому что мы не создали тот файл/и вот тут/да это не может найти файл/так что давайте создадим это/и вот тут я собираюсь создаь в то чности такой же файл который у нас был прежде/construct name назначим имя и давайте оставим это в таком виде/это окей/теперь если я скомпир=лирую это/ gulp /это автоматически позаботиться о компилировании babel для вас/так что вам даже можно не думтаь об этом/если вы сейчас перейдете в public /js/ min js/тут наш преобразованный код который будет работать во всех браузерах насколько это возможно/так что да/это хорошая работа и если вы используете laravel то рассмотрите возможность использоватьlaraevel elixir и вам не придется даже думать обо всех этих штуках babel/только работайте/окей и в следующем уроке мы собираемся начата работать с es 6/

/

Если так сложилось что вы используете Laravel как ваш основной framework тогда вы обрадуетесь узнав, что с Laravel Elixir вам даже не надо думать о Babel. Просто сосредоточьтесь на работе.

/

# [ES2015 Crash Course:](https://laracasts.com/series/es6-cliffsnotes)Default Parameters

Окей следующий довольно простой но я думаю вам это понравится/ecma script 6 сейчас поддерживает параметры по умолчанию/представьте что у вас есть некоторая функция тут которая будет применять скидку/вы передаете ей стоимость чего то и скидку которую хотите применить/и затем предположим стоимость это 100 рублей и вы хотите применить скидку 10 процентов/и затем мы хотим получить ответ в виде стоимости минус стоимость на скидку/и как пример стоимость 100 и умножим на скидку 10 и получим 10 и затем вычтем 10 из ищначальной стоимости и получим 90/но что если мы хотим параметр по умолчанию тут?другими словами если вы не предоставляете скидку тут мы автоматически подставим 10 процентов/хотелось бы что бы мы могли только спросить о скидке и жто автоматически дало 10 процетов/чтож обычно с es 5 вы могли бы сделать что то вроде установкить значение по умолчанию тому что вы передадите/в ином случае это будет равно 10 процентов/и теперь если мы попытаемся применить это appydiscount 100 40 процентов/и мы выведем это/и если я преключусь в браузер получу такое сообщение/60 после применение скидки/но если мы используем значение по умолчанию/удалим это/ и теперь это показывает 90 потому что за минусом 10 процентов получает ся90/отлично/но вот с es 6 нам больше не надо делать ничего пообного/вместо этого прямо как в php или других современных языках мы просто можем указать это тут/ и я все еще буду получать тот же результат/небольшая заметка/это не обязательно должно быть каким то значением параметра/наприер если у вас есть другая функция или метод/которая обозначает ставку скидки по умолчанию и возвращает 10/вы пониамет к чему я клоню/ если вы хотите вы можете вызвать эту функицю прямо тут/и это немного уникально/обновимся тут и получаем все тот же результат/и вот это все о значениях по умолчанию в ecma script 6

/

ES5 не поддерживает параметры по умолчанию. В связи с этим мы все прибегали к менее элегантным альтернативам (как name = name || 'Joe'). К счастью, с ES6 это все работает как вы и хотели. Даже лучше. Теперь вы не ограничены только примитивами. Вы даже можете установить значение по умолчанию равным результату вызова функции!

/

[ES2015 Crash Course:](https://laracasts.com/series/es6-cliffsnotes)Template Strings

Теперь шаблонные строки или как вы можете их звать шаблонные литералы одна из моих любимых вещей/представьте вы резервируете некоторое значение для шаблона и это будет некоторый html/так представьте что вы строите html блок чтобы потом выбросить в dom или например вы используете react js или что то вроде того/скажем у вас есть тут div класса alert и затем внутри него у вас есть параграф и вы закрываете это и наконец вы можете захотеть отформатировать это немного просто потому что это приятнее воспринимать/но это несколько болезненно делать это вручную или вы могли бы использовать обратный слеш что некоторые люди делали в прошлом/или один из подходов который также использовался это строить ваш html как массив/что то вроде того/так что у меня есть открытый div тег и затем мой параграф и затем закрывающий тег и затем вы могли бы gоставить implode или join используя пустою строку как ограничитель/и это собственно распространенный подход который я часто использовал/и вабще то это выглядит достаточно чисто по сравнению с некоторыми альтернативами/но теперь вместе с es 6 у вас есть шаблонные строки/ которые достаточно полезны и не беспокойтесь тут совсем немного что надо понять/это очень просто/и для начала мы можем написать console log и посмотреть как это выглядит/ и если вы используете chrome конечно вы увидите нечно подобное и теперь я удалю этот массив и мы заменим его на эту обратную ковычку/думайте об этом как о просто одиночной кавычке/ и теперь давайте избавимся от этих кавычек/вот и все/сейчас у нас есть мульти линейная строка и фактически вам больше ничего не надо делать тут/и если я вернусь обратно в

**/удалить jefrey chrome/**

хром/обновлюсь и вот результат/именно тот который мы хотели/теперь /что если мы мы хотели сослаться на значение?/например foo это динамическое значение/скажем let равное name Foo и затем мы хотим сослаться на это/и если бы мы это делалли старым способом нам бы пришлось использовать конкатенацию/пришлось бы писать что то вроде p равно значению name/ухх это боль/но теперь с шаблонными строками мы можем использовать уже знакомый синтаксис прямо тут/окей/вернемся и обновим/и мы все еще получаем foo/но мы можем это изменить и теперь это обновится/и вабще то заметьте как мы тут имеем дополнительное пространство и это приходиит нам отсюда/и например если я удалю это – то больше мы его не видим/часто это совсем не важно/но если важно помните вы всегда можете использовать методы для работы со строками/и никаких проблем/вернемся обратно и обновимся и получаем тоже самое/теперь/я всегда использую это при работе с vue js/если вы смотрели мои серии – отлично/если нет-рекоммендую посмотреть/

**3.15 удалить ссылку**

/в любом случае вы часто создаете новый экземпляр vue и указываете шаблон/ часто вы ссылаетесь на этот компонент и вы гли бы использовтаь селектор и это будет подцеплять html/но другая опция – это захардкодить что хорошо работает для простых вещей/и вот например для нашего alert примера мы могли бы сделать нечто подобное/ и снова foobar или вы можете принимать динамический элемент от пользоватлея/это ен важно/но что важно и чтоя часто делаю/я переключаю это на шаблонную строку и я могу отформатировать это немного лучше/вроде того/и это практичный пример того что я делаю постоянно/и это все/ и если в прошлом вы делалил что то вроде того/foo плюс бар плюс еще что то//буээ/вам просто ненужно делать ничего такого/вместо этого используйте одинарные ковычки/

/

Шаблонные строки позволяют нам удалить все хитрости, к которым мы прибегали для создания элегантных мульти линейных строк. Теперь, просто замените все эти одинарные кавычки обратными кавычками, и вы закончили. Даже больше, они также поддерживают замещение переменных. Больше никаких конкатенаций!

/