# **­**[The PHP Practitioner:](https://laracasts.com/series/php-for-beginners)

#### [**Step 1: Get PHP Installed**](https://laracasts.com/series/php-for-beginners/episodes/1)

Окей меня зовут павел курский

И это выглядит так как будто вы хотите выучить php

**Очень круто**

**Давайте сделаем это**

Шаг первый – вам нужно установить PHP верно?

Прямо сейчас существует множество способов сделать это

Так например вы уже можете обнаружить что php уже установлен

Если вы на MAC OS то откройте свое приложение терминала

И далее это может выглядеть также как и у меня

Как альтернатива, если вы имеете немного больше опыта вы можете использовать приложение под названием iterm

~~//обрезать видео~~

Так что мы собираемся напечатать php dash v чтобы понять какая версия php уже установлена

В моем случае как вы можете видеть я уже установил php

У меня установлена версия php 7 которая была выпущена летом 2016

Примерно в то время когда была сделана эта запись

На тот момент это был самый последний релиз

Но возможно у себя вы увидите php 5.5 или 5.6

*и это окей для начала*

Как альтернатива вы конечно же можете установить последнюю версию используя различные приложения

Как вариант вы можете использовать homebrew

Сейчас это предназначено для mac пользователей

Но если вы Windows пользователь – *мы вернемся к вам через минуту*

Вот как homebrew дает вам отличную возможность установить любое количество приложений используя командную строку

**Это действительно круто**

Так что сейчас вы можете просто скопировать эту строку

Вернуться к терминалу или к iterm как я говорил вначале и вставить

**Окей** В моем случае homebrew уже установлен, но для вас это установит homebrew Далее как вы можете видеть вы можете установить любое количество приложений используя этот формат

brew install И далее имя приложения

Также вы можете использовать поиск приложений

Так если вы не знаете имя вы можете просто сказать

**Найди мне что ни будь что связано с php** И вот результат

*Замечу что это может быть немного сложно сначала Но это нормально*

Давайте просто посмотрим на php 7

Так, сейчас это последняя версия на момент этой записи

Также тут множество вещей которые мы можем установить

Такие вещи как mncrypt и memcashed

Вы выучите все это чуть позже

Но не волнуйтесь об этом слишком сильно

Сейчас нас интересует только php 7

Скопируйте эту строчку

Еще раз вы видите что у меня уже установлен php

Но вы должны вернуться в терминал и написать

brew install и затем вставить то что вы скопировали

**И на этом все Brew установит все за вас**

Так что когда он закончит, попробуйте открыть новую вкладку и еще раз напечатайте php dash v

И надеюсь вы увидите последнюю версию php

*Если вы столкнетесь с какими то проблемами при установке Просто задайте вопрос под этим видео*

Далее вам больше не нужно использовать homebrew если вы не хотите

Вы можете использовать приложение под названием mamp

Вы можете думать об этом как о решении все-в одном и это даст все что вам нужно для начала

**Пару лет назад это приложение было невероятно популярно**

Так что это опция для вас если вы на Windows/ MAMP работает и на Windows тоже/ раньше не работало но теперь работает

Но вы также можете использовать приложение под названием WAMP или еще одно под названием XAMP, которое работает на windows linux или mac

Повторюсь это решение все-в-одном и это даст все что вам нужно для начала

Вы свободно можете выбирать любое приложение которое хотите

Но если вам нужен мой совет

Если вы на mac используйте homebrew

Это правда просто

Если вы на Windows используйте любое из этих приложений

Так что вот вам домашняя работа для этого урока

Установите php

Еще раз /Используйте terminal или iterm и затем Напечататйте php dash v чтобы увидеть как минимум php 7 на выходе ­­­Отлично/ действуйте)

**[Step 2: Install a Proper Code Editor](https://laracasts.com/series/php-for-beginners/episodes/2)**

Итак /На вашей машине установлен PHP и теперь вы хотите написать код/Но очень вероятно, что у вас еще даже нет редактора кода /И если это ваш случай - никаких проблем/ сейчас я вам дам три быстрые рекомендации / Но сначала небольшой дисклеймер/редакторы кода почти как мода и одежда - каждый имеет свое собственное мнение /и на самом деле один не лучше чем другой - все просто сводится к тому, чтобы понять, какой редактор подойдет вам лучше / **так что не позволяйте никому давить на вас в использовании одного вместо другого**, но вот 3 мои рекомендации если хотите - Sublime текст является что то вроде Золотого ребенка в сообществе на момент этой записи, но это не бесплатно / Но есть своего рода демо-версии, которые можно использовать почти бесконечно, так что, возможно, дайте ему шанс, и если вам он понравится конечно / заплатите разработчику за полную версию / он доступен для Windows, Mac или Linux, и кстати в большинстве серий это тот редактор, который я буду использовать / далее вы можете использовать Атом / прямо сейчас он совершенно бесплатный / я хочу, чтобы вы также его попробовали, это созданное сообществом решение / и очень быстро растет / так что Атом представляет собой прекрасный инструмент / и бесчисленное количество разработчиков используют его каждый день / ваш третий вариант - это phpstorm и это то, что мы называем айди и / так что у вас есть несколько редакторов, которые немного проще/ они не пытаются проверить код в той степени, как это делает IDE /так что это главное отличие IDE вроде phpstorm /именно он имеет глубокое распознавание языка PHP/ в то же время вы можете расширить Sublime и добавить ему немного большее понимания языка, но вообще это не их задача/ наверно вы думаете - почему вы не всегда выбираете phpstorm? И ответ в том - что большАя осведомленность о языке это недостаток /это работает немного медленнее как правило, так как большое количество конфигурации может немного давить на производительность / но тем не менее я использовал phpstorm довольно много/но на этот момент в вашем обучении я предлагаю не касаться этого редактора, в пользу того, чтобы попробовать атом или SublimeText/ если я открою Sublime Text **/**вот что я получу /ваш может выглядеть немного по-другому только потому, что вы только что его установили/ но опять, как мода и стиль вы можете настроить свой редактор, как вашей душе угодно и разработчики **делают это все время**/ мы постоянно экспериментируем с новыми цветовыми схемами и это звучит как бред какой-то /это вроде выглядит как будто мы работаем, когда мы действительно просто играемся со схемами/ но тем не менее ваш редактор кода готов чтобы начать/ так что я нажимаю файл>сохранить, и мы сохраняем это как новый PHP файл, и вы увидите, что я установил пустую папку на моем рабочем столе /вы можете поместить его в любом месте где вы хотите/мы собираемся вызвать индекс точка PHP /заметьте это расширение файла/ когда вы хотите чтобы что то интерпретировалось как PHP вы должны добавить это расширение точка php и это наш способ сказать - это PHP файл и таким образом мы также информируем редактор кода, с каким языком мы работаем / таким образом, он сможет обеспечить его правильное выполнение/ так например возможно у меня есть класс под названием report /также Вы заметите, что тут есть специальная подсветка /может тут есть так называемый метод generate и, возможно, он принимает на вход user, который создает report/ в очередной раз вам не нужно знать все это сейчас/ я просто демонстрирую, что с правильным редактором кода вы получите качественную подсветку синтаксиса/ в любом случае/ давайте удалим все это и сделаем это немного проще, давайте выведем что нибудь на страницу - echo привет мир /хорошо, так же просто, как это есть/для сначала давайте Сохраним этот файл и в очередной раз/ вы можете сделать это вручную, но вы знаете, в большинстве редакторов вы просто нажимаете command S на Mac или Control s/ хорошо /далее вы возвращаетесь в ваш терминал, и вы пишете CD/перейдите в каталог, где этот файл находится/ в моем случае он был под названием PHP-learning/хорошо сменим папку/вы могли бы сделать что-то подобное на вашем рабочем столе, а затем сослаться на папку, которую вы создали /хорошо, так если я отображу список файлов с помощью ls/ тут есть мой файл index/ теперь, если мы хотим запустить это я могу напечатать PHP и потом путь к файлу

Так/ вы только что написали свой первый PHP-скрипт /это очень просто, и вы уже на верном пути,/ так что у нас получилось /мы видим Привет мир на экране, потому что мы сказали терминалу это сделать, echo это что то вроде печати, когда вы говорите echo по сути вы имеете ввиду - покажи это на странице если у вас есть браузер или через консоль- если вы в терминале /теперь насчет ковычек и насчет точки с запятой /это основные строительные блоки PHP /Для всего что вы хотите чтобы интерпретировалось как PHP вы изначально начинаете с этого набора/ это наш способ сказать, хорошо это все PHP /и это выглядит немного странно на первый взгляд, напечатайте - знак меньше вопросительный знак PHP/ хорошо не забудьте это сделать так как много новичков забывают, и потом они не понимают, что происходит/ например, если я попытаюсь сделать это - ничего не сработает, видите/ в этом случае он просто выплевывает все содержимое на страницу, это никак не было обработано, если бы я написал тут какуюто чушь, и снова запустил это/ да /нет никакой обработки/ однако если мы теперь вернем PHP то он будет интерпретирован и на самом деле в моем редакторе есть некоторые настройки/ так что очень вероятно, что Вы не увидите этого/ и это нормально/ вы узнаете больше о редакторе в будущем /в моих сериях есть Sublime text эпизоды, так что я думаю, что вы узнаете много когда вы будете готовы/ в любом случае/ редактор показывает нам что тут творится что-то странное и если я выполню это/ мы увидим ошибки/ мы не подготовились правильно и PHP становится раздраженым/справедливо/ так что теперь мы знаем, что любой код в файле, который должен быть опознан как PHP должен начинаться с этой последовательности/ дальше вы знаете, что echo что то типа – напечатай это на экране/ Двигаемся дальше/ тут мы ссылаемся на строку/ так что строка - просто основной текст/это не команда это не функция, это не переменная /это просто набор букв/ Теперь мы обозначаем строку, обернув ее в одинарные или двойные кавычки, и есть очень небольшое различие между ними, но пока не волнуйтесь об этом/ вы можете использовать их как взаимозаменяемые/наконец, точка с запятой/ так мы показываем конец определенной последовательности или определенной команды, и это то, что вы будете использовать всегда /как правило, когда вы пишете команду, даже такую простую вещь, как эта, вы заканчиваете ее с точкой с запятой, но давайте представим, что Вы забыли сделать это /давайте представим , что вы написали echo привет мир/ в очередной раз можно видеть из моего редактора – ему это не нравится, так что если я запускаю это снова/ Да/ мы вызвали ошибку, неожиданный мир, и вот в чем дело/ Hello, может быть функцией, может быть процедурой или переменной, или ничем, но мы действительно хотели сказать - нет, ничего подобного, это просто набор букв, и это может быть любая строк, так что мы должны обернуть ее в кавычки, чтобы сказать/да, это просто текст/ не беспокойся о том что там написано/ просто покажи это на странице/ теперь, если мы запустим - это все равно не сработает и еще раз, это Потому, что мы забыли точку с запятой/ так что мы добавим ее обратно, запустим снова, и теперь все работает нормально/ так что давайте оставим это как есть/ вот ваше домашнее задание /это не имеет значения, какой редактор кода вы выбирете, но выберите один и установите его на вашем компьютере/ Хорошо, после того как ваш редактор установлен, повторите то, что мы сделали в этом уроке/ создайте новую папку на рабочем столе откройте новый файл и назовите его index точка PHP/ затем выведите любую строку какую вы хотите /а затем, наконец, откройте терминал и напечатайте PHP, а затем имя файла, который вы создали/ как только у вас это заработает, и вы почувствуете себя комфортно - вы готовы к шагу 3

Увидимся

Step 3: Variables

На нашем уроке вы на самом деле узнали больше, чем вы думаете/ вы узнали, как создать файл PHP, как вызвать его из командной строки, а затем вы и узнали немного о строках и то как закончить обьявление любых команд или операции точкой с запятой/ так что теперь давайте перейдем к переменным/ но прежде чем мы это сделаем/ смотрите/до этого момента вы делали это через терминал, но вы могли задуматься/ я хочу сделать сайт / я хочу увидеть это в браузере/ справедливо/ давайте посмотрим/ пишем PHP черточка эйч/ эта команда покажет нам справку/ это просто список вариантов как можно запустить сервер из командной строки/ я собираюсь найти server/ **хорошо**/ заметьте, если я добавлю тире, заглавная буква S а затем адрес - Я могу запустить встроенный веб сервер/ **круто**/ давайте попробуем PHP dash S и мы дадим ему название localhost и мы хотим, чтобы порт был 8888 /на самом деле любые 4 цифры подойдут/ но просто используем это для начала/ **окей/** так с помощью PHP 7, мы сейчас прослушиваем этот адрес, так что вы можете скопировать и вставить его в Chrome или FireFox, или если вы используете iterm вы щелкаете command click/ хорошо/так/ теперь браузер всплывает и мы видим веб-представление кода который мы написали/ **круто** /я знаю, что это немного веселее, чем делать это в консоли так что никаких проблем/ давайте вернемся в sublime текст/ следующий шаг это переменные/ и переменные пугают немного вначале/ я помню, когда я впервые учился, я думал про себя/ **переменные - круто**, но почему я вабще должен использовать их/ почему вместо этого я не могу просто ссылаться непосредственно на значение?/ и как вы понимаете тут много причин чтобы сделать это/ поэтому первым шагом, когда мы создаем переменную мы делаем следующее/ добавляем знак доллара, а затем любую строку/ мы делаем это путем ввода знака доллара, а затем что бы дать название любое слово которое хотите для этого сообщения или приветствия /например greeting/ хорошо /назовем ее greeting и затем мы делаем эту переменную равной/еще раз/здесь есть множество разных вещей которые нужно понять, но сейчас вы просто знаете о строках/ поэтому мы собираемся вставить это здесь/ еще раз /мы закончили с нашей маленькой операцией, поэтому мы завершаем ее точкой с запятой/ хорошо/ **это ваша первая переменная/ хорошая работа**/ так что теперь, если вы хотите чтобы echo сработало мы можем заменить все это именем переменной greeting/ хорошо, сохраним его с control или сommand S и переключимся обратно на хром/ и если вы обновите вы увидите в точности то же самое, но теперь вы используете переменную/ **отличная работа**/ а теперь представьте, что вы хотите изменить это на Hello Universe/ нет проблем вы обновляете значение переменной и это автоматически отразится здесь/вы можете нажать command tab - это позволяет очень быстро переключаться между недавно использованными приложениями/ так что вы увидите как я постянно это делаю /хорошо/ в любом случае, если я обновлюсь /я нажму command R, что делает то же самое, не трогая эту маленькую кнопку/ в любом случае мы видим как произошло обновление/ хотя вы все еще можете думать про себя/нуокей, я не понимаю, почему это лучше чем то, что было раньше?? и ответ в этом случае - если мы изменили значение переменной то мы сразу же можем вывести ее значение на экран/ так что возможно это кажется на этом примере лишним, но потом вы поймете, что в реальной жизни вы будете изменять эти значения все время/ например, представьте, что у вас есть ваше имя хорошо создайте переменную/ не забудьте всегда использовать знак доллара в начале /это то, как мы описываем, что это переменная/ присвойте ей значение любого имени например Джеффри или типа того/ так/ далее мы собираемся написать echo Привет/а затем имя переменной/ хорошо, но вы **не мОжете** сделать это таким образом/ если мы выполним это то увидим имя переменной, а не значение, связанное с ней/ и если вы помните один или два урока назад я заметил, что есть небольшая разница между одинарными и двойными кавычками, так что вот разница - в двойных кавычках вы можете использовать переменные/ так что мы можем интерполировать такие вещи автоматически, но только если у вас есть двойные кавычки/ так что если я обновлсь/ сейчас вы увидите, Привет Джеффри/ ок/ но в одинарных кавычках, Вы не можете этого сделать/ вот еще один способ как вы можете сделать это/ вы можете использовать конкатенацию, так что я могу использовать формат периода здесь, а затем имя переменной, и вы знаете, что период - это запутанно/ много других языков используют что то более вразумительное, как например плюс/ но в нашем случае PHP хочет, чтобы мы использовали период, чтобы понять это/что я хочу вывести на экран - "привет" запятая пробел а затем объединить или добавить к этому значение этой переменной name/ и вот как это выглядит/ но в этом случае да/ я часто делаю что-то подобное Hello name и далее окружаю его брекетами / и на самом деле это необязательно, но многие люди делают это просто, чтобы сделать код более понятным, что они ссылаются на переменную/ но давайте оставим это без брекетов пока/ хорошо/ подумайте о гибкости, которая у вас сейчас есть эта переменная name может быть определена где угодно/ это может быть даже определено в небольшом поле помещенном на странице, где вы, например перейдете на сайт и там написано - введите ваше имя/ так что вы, печатаете свое имя Иван Петров или Маша Петрова/ а затем, когда вы нажимаете кнопку, он говорит вам привет Иван или привет Маша/ так что значение, которое вы вводите может быть сохранено в переменной и это может соответствовать вашему имени/ так что если мы изменим это на Джейн / теперь это будет автоматически обновлено, чтобы отразить это/ хорошо, так что вот что мы делаем в этом уроке/ если Вы похожи на меня то на эти переменные потребуется немного времени, чтобы осознать и понять - почему я хотел бы использовать их/ так что если это то что вы чувствуете, вы абсолютно **нормальны** /никаких проблем, просто привыкните к этому/домашнее задание к этому уроку-это продолжать играться с этим/ просто создайте переменную и затем найдите способ, чтобы вывести его на экране и если вы хотите повеселиться с этим - создайте несколько переменных, таких как greeting/ и это может быть Hello, затем вы могли бы сказать echo greeting, а затем мы соединим строку, а затем соединим name/ сейчас мы строим полное предложение, используя две разные переменные /вот теперь, если мы обновимся, мы получим то же самое/ или, как я сказал, можно вложить это в строку - только если это двойные кавычки/ так что greeting name и будет то же самое/как я сказал, многие люди часто оборачивают в двойные ковычки, особенно в случае когда вы просто хотите выделить переменную чуть больше/ вы могли бы сделать нечто подобное/ обновимся и, конечно, вы все равно получите то же самое/ хорошо, так что продолжайте играть с этим/ и в очередной раз, когда вы закончите,

перейдите к шагу 4

Step 4: PHP and HTML

Добро пожаловать на шаг 4 / до сих пор мы просто выводили текст на экран/ но, конечно, вы знаете, что это сильно отличается от любого типичного веб-сайта/

так кАк мы можем использовать PHP вместе с HTML?/ это столпы PHP/ и здесь он стал действительно супер популярным, поскольку РНР строился для web/ он подцепляет все очень просто и тут остается не так много работы для вас/ хорошо/прямо сейчас я просто хочу, показать немного HTML/ я надеюсь, у вас есть HTML опыт/, так что, если все это выглядит действительно подавляюще то вы могли бы нажать на паузу и освежить память об HTML/ а пока, я продолжу/ хорошо, так/ давайте поставим здесь тег эйч один и я напишу /hello world/ Итак, если я нажму command+tab то вернусь обратно в браузер /обновлюсь и мы все равно увидим/ привет мир/ но если я посмотрю источник страницы то мы увидим, что у нас есть хороший HTML-код, который можно стилизовать и настроить его так как мы хотим/так почему бы нам не написать еще немного HTML-кода/я собираюсь создать секцию для hello world и тогда я помещу немного CSS прямо здесь, в самом верху/ немного стиля для заголовка/Цвет фона серый и посмотрим, на что это будет похоже/обновимся/ добавим немного отступов вокруг, и затем добавим выравнивание по центру/вот результат/, **очень просто**/так что теперь давайте вернемся к PHP/ Сейчас это значение захардкожено. Верно?/ мы ссылаемся на него непосредственно/ но что, если мы хотим использовать PHP вместо этого/ вспомните, что вы узнали в первом или втором уроке/вы хотите, чтобы часть вашей страницы интерпретировалась как PHP и у нас есть это расширение/ затем вы открываете PHP тег/ да, вы также узнали, как вывести hello world, но Вы не узнали, как закрыть PHP/ так что то что вы увидите сейчас, это не правильно/ и если я обновлюсь мы получим сообщение о синтаксической ошибке /так вот правило/ **если у вас есть файл, который на 100% PHP то вам не нужен закрывающий тег**/ например PHP равно Джон, и если я не имею ничего ниже этого уровня, что не является PHP, я могу опустить закрывающий тег/ и закрывающий тег кстати это такая небольшая последовательность/ вопросительный знак знак больше/ так что открыли PHP и закрыли PHP/ это наш способ сказать, что все что между ними следует интерпретировать как PHP/ однако, если это целиком и полностью PHP-файл, это вообще хорошая практика, удалить это совсем/я собираюсь вернуть все как было/ и в этом случае у нас больше нет php кода/ Так что я должен поставить это/ если мы вернёмся обратно и обновим мы увидим сообщение Hello world снова но мы использовали php в этот раз/ так что, например, если мы хотим привести все в порядок и разместить все на своей линии/ можно форматировать так, как вы захотите/ может быть, мы зададим переменную здесь/ greeting равно Hello world/ и затем вы выводите greeting/ так что мы возвращаемся/ "обновить" /и я все еще получаю то же самое, теперь давайте сделаем что-то веселое/ Может быть значения приходят к нам из строки запроса или другими словами из части url/ Возможно вы это уже видели/ Когда вы переходите на веб-сайт Вы часто видите вещи что-то вроде - order равно asc И keys равно values/ вы видите подобное всё время / думайте об этом как о возможности передать данные ключ и значение на сервер/ в этом случае мы можем прочитать эти значения и обработать в коде как нам нужно/ например ,если вы видите вещи как asc /это может быть способ сказать /неважно какие значения мы напечатаем Я хочу отсортировать их по возрастанию/ может быть по дате/ или если здесь написано desc это наш способ сказать отсортируйте по убыванию /Так что возможно мы можем написать здесь что-то вроде - name равно Jeff/ Хорошо/ Мы собираемся захватить это имя и сказать - Hello и далее имя/ OK такчто вот кое-что новое/Мы собираемся сослаться на супер переменную/ и это звучит немного необычно / Но на самом деле это просто знак доллара нижнее подчеркивание GET /это вабще новая конструкция которую вы еще не знаете/ это массив/ мы будем изучать массивы в следующем уроке /Так что просто знакомитесь с этим/ OK/ Так что мы просто собираемся захватить ключ из url И в основном - То что идет до знака равно это ключ/ то что после знака равно это значение связанное с ключом/ Так что я хочу получить значение связанное с ключом name которое является url параметром/ и надеюсь это вернет значение Джефф/ пробуем/ Давайте выведем имя / если я вернусь и обновлю / и вот результат/ или Мэри/ или Джон **достаточно круто** /так Почему бы нам не поприветствовать их/ Hello запятая и затем name/ Так что если я вернусь и обновлю то мы увидим/ сейчас мы добавили немного динамики сюда /сейчас мы можем захватить имя из внешнего мира, Прикрепить его к переменной, И затем вывести его на экран/ например Джеффри/ и вот мы видим что значение обновилось / сейчас давайте Вернемся обратно и немного почистим/ забавная вещь в кодинге это то что кодинг **очень-очень креативный процесс** и Каждый имеет свое собственное мнение о том - что выглядит хорошо/ что подходит друг другу больше или выглядит более элегантно /Так что в нашем случае мы разбили это на несколько линий / но также мы можем сделать нечто подобное/ Давайте перепишем это немного/ эйч один / В этом случае также как мы и говорили в предыдущем эпизоде/ *мы действительно нуждаемся в этой переменной если мы ссылаемся на неё только в одном месте?* Возможно нет /так что в этом случае я напишу/ php эко Hello запятая и затем/ я не могу оставить это в таком виде потому что мы находимся в строке/ это не Переменная /мы ссылаемся на элемент внутри массива /и снова - Вы научитесь этому в следующем уроке/ Так что мы собираемся использовать конкатенацию здесь/ период GET и затем name/ используем точку с запятой чтобы закрыть/ Так что мы можем взять всё это и удалить/ И множество людей скажут что это выглядит проще/ Так что если мы вернемся обратно и обновим мы получим тоже самое /далее ещё один маленький секрет /Вы часто будете делать что-то подобное/ php echo / но как оказывается есть сокращение для этого /Так что если я напишу знак меньше вопросительный знак знак равно /Так что если я вернусь обратно и обновлю / мы увидим тоже самое/ вы будете часто видеть подобное /OK/ **Всё выглядит отлично!/** Но со временем вы продвинетесь дальше и вы будете узнавать более сложные вещи такие как безопасность/ сейчас мы действительно не хотим ничего знать о безопасности потому что мы все еще пытаемся понять что такое Переменная или массив /Сейчас я просто хочу пролить свет на это/ представьте что пользователь вместо имени также добавил некоторый HTML код / И на самом деле если мы обновим/ вы увидите что мы обработали это / Или например пользователь оставил ссылку/ если мы запустим это /заметьте что всё что мы ожидали это имя пользователя/ Но на самом деле сейчас это ссылка и если мы нажмем мы попадем в Google/ **отлично** **но мы этого не хотим** /и вы можете подумать что пользователь не будет этого делать/ **но он будет** /основное правило при разработке веб-сайтов - думайте что ваш пользователь хочет навредить вам /Так что вам следует обзавестись специальной защитой/ Так что не разрешайте им вставлять подозрительный код в ваши скрипты/ OK/ просто немного забегая вперёд/ сейчас Вы не должны знать всё это просто помните что **безопасность очень важно**/ так что сейчас я собираюсь научить вас 1 функции /Думайте о функции как об обозначенной логике/ Представьте что вам нужно сделать А Б В/ так что если вы обернете это в функцию и вы можете просто ссылаться на имя функции / за кулисами она будет делать А Б В/ Вы можете использовать сколько угодно это функцию/ и если вам нужно это делать другом файле / да/ вы просто можете использовать имя этой функции и всё ещё получать всё тот же результат/ как оказалось Вы можете создавать свои собственные функции/ но здесь есть огромное количество встроенных функций Которые предоставляет PHP/ но сейчас вы заинтересованы только в одной html Special chars / Сейчас вы видите как я сослался на это/ это имеет следующий формат/ название функции и двойные скобки / для того чтобы передать параметр в функцию вы просто делаете Это здесь/ И я покажу вам всё это в следующих уроках/ сейчас вы видите как мы вызываем эту функцию и затем внутри Передаем значение из URL / Так что в нашем случае Мы хотим передать это прямо сюда/ если мы вернемся обратно/ сейчас мы видим что-то иное/ На текущий момент мы не сделали конвертацию /мы видим все эти специальные элементы/ и изза того что мы не сделали конвертацию мы видим всё в сыром виде / Однако используя функцию html Special chars Мы конвертируем все эти элементы которые мы называем HTML сущностями/ Так что это просто HTML представление для ключа/ это то что мы называем стерилизация входных параметров/ мы принимаем любые данные приходящие от пользователя которые поставляются пользователем на вашем веб-сайте/ мы считаем что он хочет навредить/ Так что нам нужно стерилизовать эти данные/ мы берём их/ преобразуем/ убираем всё что мы не хотим видеть и затем оставляем только то что мы собираемся использовать/ это выглядит немного странно Но не волнуйтесь/ как я ранее говорил Я не хочу вам рассказывать о безопасности слишком много/ вы до конца вашей карьеры будете знакомиться с этим/ но сейчас вы просто учитесь встраивать PHP в HTML код/ вы также выучили вашу первую функцию и вы также научились забирать любое количество параметров из URL / Так что я думаю этого достаточно для этого урока/ Вот ваша домашняя работа/ Попробуйте воссоздать предложение динамически в том же формате в котором мы делали ранее/ Запомните веб-сайт /знак вопроса ключ равно значение /Затем в вашем коде подтяните это в вашу функцию / Затем стерилизуйте данные используя HTML Special chars и свяжите это с любым забавным предложением /каким только хотите / и проследите что вы можете сделать это безболезненно/ И как только вы это сделаете - Вы готовы к следующему шагу

Step 5

В этом уроке мы имеем типичную html страницу где мы можем разместить любое количество php логики/ так что если у вас есть переменная и вы получаете что либо

Из строки запроса/ вы изменяете это, а затем ,возможно, делаете Запрос к базе данных /и в итоге показываете это на экране/Есть кое что важное что нужно понимать/

Может быть это имеет более высокий уровень/Но это принято считать best practice/

Так что если вы думаете о разделении зон ответственности кода/Этот файл выглядит так как будто его задача это отобразить страницу/Но вы также имеете, как я сказал раньше, если ваш php/ который Соединен с базой данных, чтобы запросить данные/То это уже является другой задачей/Основное правило, которое является хорошей практикой Это разделить задачи так скоро как это возможно/Особенно, если вы привыкнете к этому в течении нескольких лет,Вы поймете ,что множество фреймворков, которые вы используете Следуют этому правилу/ вы имеете одну секцию где

Готовите данные и другую где вы отображаете данные/ Это именно то что я имел ввиду когда я говорил о разделении Зон ответственности/ Почему бы нам не применить это правило на самом простом уровне /Мы создадим один файл где мы создаем данные

И другой файл который будет содержать html и отображать страницу/ ОК

Я создам тут файл и назову его index точка view точка php Или index . tmpl . php который для шаблонов/ Или так как вы хотите, это не важно/

Окей/ Мы собираемся перейти сюда, захватить весь html И поместить его в наш view файл/ Далее, эта секция, где мы имеем php код/Мы преобразуем ее в простую переменную greetings/Я использовал это сокращение которое заменяет php echo И выглядит красивее/Html содержит определенную порцию вашего веб сайта/Мы собираемся использовать view в названии каждого файла/Это в некотором роде общая терминология/Ок/ прямо сейчас мы вернемся к index.php/Я могу удалить все это и затем вот что мы собираемся делать/Первый шаг наш view файл или template ожидают greeting переменную/Так что давайте ее создадим. Hello world / сейчас если я перейду обратно к chrome Конечно мы ничего не увидим пока/Это потому что мы загрузили эту страницу но нет ни одной ссылки на этот файл/Мы собираемся использовать ключевое слово require/И далее index.view.php/Думайте о require как будто говорите - вытащите все что расположено тут/Так что мы собираемся потребовать весь этот контент/Окей/Перезагрузим и вот результат/Сейчас если я посмотрю на источник мы имеем Все что мы видели в предыдущем уроке/Но сейчас мы разделили куски кода/У нас есть файл который получает и готовит данные/И также у нас есть ссылка на другой файл, задача которого Отобразить эти данные на экране/Это самая простая форма верхнеуровневого концепта разделения Зон ответственности/**Отлично!**/Немного сложно конечно/В следующем видео я расскажу вам о новых конструкциях/ Мы узнаем о массивах в php

Step 6: Understanding Arrays

Сейчас Вы готовы узнать о массивах /так вот чтО мы оставили в предыдущих уроках/ я собираюсь избавиться от greeting переменной / и если я вернусь к нашему view файлу/ Я удалю эту ссылку/ *Так что такое массив ?*/думайте об этом как о Примитивной коллекции связаных вещей / не делайте это сложнее чем есть на самом деле/ так как насчёт имен/ хорошо / для начала Давайте создадим переменную также как мы создавали строку под названием names/ которая будет равна/ я собираюсь использовать этот синтекс - открытая скобка закрытая скобка/это простейшая форма для создания массива/ и в этом случае здесь ничего нет /Мы только что создали пустой массив /но Мы можем заполнить его/ так что давайте добавим несколько строк/ например, Как насчёт Джефф, Джон, Мэри/ то что мы здесь имеем это массив имен /Так что сейчас Вы можете думать - *Как мы можем это использовать и почему я должен создавать массив вместо* name один равной Джефф/ name два Джон / Ну я надеюсь если Вы посмотрите на это - вы поймете - что это достаточно болезненно делать также как и сама нумерация переменных/ поэтому массив здесь больше подходит / другая интересная вещь в массивах это то что мы можем фильтровать значения, отображать их или делать другие преобразования/ да/ так например - если я хочу отфильтровать имя я могу использовать это ключевое слово здесь/ foreach/ Так что для каждого элемента массива names/ вы собираетесь узнать много нового сейчас/ а затем echo name / хорошо, я собираюсь избавиться от этой секции на время / Я начинаю с создания массива и затем я говорю - для каждого name в массиве names мы проходимся по каждому элементу/ и одно важное замечание/ в первой итерации name будет равно Джефф/ Но затем -во втором проходе равно Джон/ и на третьем проходе ровно Мэри /и затем когда больше ничего нет цикл останавливается / и ещё раз ,Мы проходим через каждое имя и показываем это на экране / Так что если мы перейдем в chrome Джефф John Mary/ это именно то что мы хотим или если вам нужна запятая мы используем конкатенацию и добавляем запятую/ сейчас если я вернусь мы увидим нечто подобное /Сейчас давайте разберем этот код потому что он выглядит немного необычно /foreach - ключевое слово и мы говорим - для каждого , что может быть найдено здесь/ в нашем случае у нас есть массив/ каждое из этих имен я могу назвать любой переменной это может быть foobar, thing / но лучше что-то связанное/ массив имен поэтому я дал название name /Так что для каждого из них *что мы собираемся делать* / мы ставим брекеты /любой код который будет помещен внутри брекетов сработает для каждого из элементов в массиве /Так что мы хотим сделать что-то простое имея массив имен/ Я хочу отобразить их как элементы списка в нашем HTML/ О'кей давайте сделаем это/ так что в нашем view файле Мы устанавливаем неупорядоченный список для каждого элемента списка/ другими словами это будет выглядеть вот так/ Джефф Джон и Мэри / это в итоге то что я хочу/ но мы собираемся делать это динамически ссылаясь на нашу переменную/ сейчас мы будем использовать PHP / добавляем PHP код / Так что мы собираемся сказать - для каждого names as name - тоже самое что мы делали недавно/ и сейчас я собираюсь показать вам два разных способа/ 1 опция /это то что вы знаете /echo двойные скобки элемент списка И затем name / типа того /и это будет работать /сейчас давайте вернемся к index.php и require наш view файл/ Давайте посмотрим что у нас есть / Обратно в Chrome/ обновимся / и вот результат/ Так что это достаточно клёво/ мы подготовили данные ,затребовали файл view и внутри view мы запустили цикл /для каждого элемента списка мы вызвали функцию echo/ *просто не правда ли?* окей /*но какая Вторая опция* ? Так для этой команды У нас есть сокращение, что особенно хорошо когда вы работаете с html / Так что это может быть переписано/ опустим это вниз/ я покажу вам вторую форму / Так что вот что мы собираемся делать / Мы собираемся избавиться от этих брекетов и сейчас мы для того чтобы создать цикл мы используем такую комбинацию/ и вы можете не использовать это каждый раз/ но как я сказал - внутри HTML кода / так что Откройте php - установить логику - закройте PHP / Если вы закроете PHP раньше - это значит остальной код является HTML /Нам нужен элемент списка/ и в итоге Мы открываем pHP и пишем endforeach / Так что Так выглядит вторая форма /первую форму Вы будете использовать в большинстве случаев / но альтернативный синтаксис - foreach endforeach / и вы используете такую комбинацию и теперь вы хотите вывести name / и теперь мы имеем два способа сделать это /перейдем обратно к Chrome/ я ожидаю увидеть оба набора/ обновимся/ Первый набор и второй набор/ так что эти двое идентичны/ далее вы сами решите что вам больше подходит/ но вы найдете однако, что в этом случае нет никаких проблем/ но по мере того как ваш HTML становится сложнее вы можете начать использовать что-то вроде/ strong item и затем добавить ссылку/ и затем когда вы добавите двойные ковычки вы увидите что это создает беспорядок /И В таких случаях вы можете использовать альтернативный способ/ так что вот что мы собираемся сделать в этом уроке/ давайте пройдемся еще раз/ вы создаёте массив через определение массива в качестве переменной и затем добавляйте квадратные скобки /внутри массива вы можете иметь любое количество элементов/ каждый из них является элементом внутри массива/ также как и сейчас мы имеем массив имён вы также можете иметь массив Чего угодно/ может быть это массив животных и это будет вашим домашним заданием/ создайте массив животных и заполните его пятью любыми животными/ а Затем в вашем view файле используйте цикл для этого массива и отобразите элементы - как элементы списка /Так что Сделайте это и затем Когда вы почувствуйте себя супер комфортно - переходите к следующей серии

Step 7: Associative Arrays

Хорошо добро пожаловать в episode 7/ в этом уроке мы собираемся поговорить об ассоциативных массивах/ как вы можете видеть наша отправная точка это файл index.php который загружает наше представление/ если мы посмотрим внутри/ да /здесь ничего нет кроме html/ отлично/ мы имеем отличную начальную точку/ сейчас вы знаете о массивах/ и мы имеем массив Animals, который содержит значения- собака и кошка/ однако мы также можем создать ассоциативный массив/ единственная разница в том, что мы можем назначить ключ для каждого из этих значений/ представьте что мы описываем себя самого или кого-то другого и мы хотим указать возраст – 31, цвет волос – коричневый/ но заметьте здесь только значение/ *что есть коричневый*? *это цвет кожи или цвет волос или это наш любимый цвет*/ У меня нет этой информации/ Ну в этом случае использование ассоциативных массивов разумно поэтому давайте изменим его/ но обращаю Ваше внимание, что сейчас вы впервые знакомитесь с числами/ до этого мы имели дело только со строками/ если мы используем число то мы не храним его внутри кавычек/ это будет текстовая версия числа 31 /но нам нужно число, так что мы просто оставляем 31/ отсутствие кавычек означает, что это не строка - это число/ в любом случае если мы перепишем это/ сейчас единственная разница в том что мы указываем ключ/ смотрите - возраст и затем syntax стрелка знак равно, знак больше, равно больше сейчас значение 31 и на этом всё мы создали ассоциативный массив который может содержать любое количество значений Так что давайте добавим еще одно цвет волос будет коричневый и позиция или карьера веб-разработчик отлично это выглядит хорошо это может содержать вашу информацию и сейчас это ассоциативный массив Теперь мы собираемся загрузить view файл который сразу же будет иметь доступ к вашему ассоциативному массиву здесь мы можем написать внутри неупорядоченного списка так что мы пишем foreach person как и что каждый из этих элементов представляет может быть особенность так что сейчас вы знаете как сделать нечто подобное для каждого элемента в массиве так что если мы попробуем это мы можем использовать сокращения как вы знаете и сейчас внутри этого мы собираемся вывести feature так что давайте посмотрим что у нас есть давайте запустим сервер как вы ранее узнали если вы используете iterm вы способны нажать command click по ссылке если у вас мак в ином случае просто скопируйте это в ваш браузер Так что мы получаем значения здесь но мы также хотим и ключи так что перейдем обратно в Sublimetext эти названия могут иметь жирное выделение а эта секция без изменений так что как нам выкинуть ключи прямо сейчас это значение мы можем сделать нечто подобное foreach person как и здесь мы вставляем ключ это приобретает Такую форму так что foreach person в массиве как ключ и затем значение заметьте это форма выглядит также как и здесь ключ и затем значение так что сейчас в каждой итерации ключ будет равен возраст цвет волос карьера и затем значение будет равно 31 коричневый веб-разработчик но в основном использование упрощенных названий не очень хорошая практика на текущий момент это OK ну вообще это сложно понять что значит это val так что правила хорошего тона указывайте специфичные имена И в нашем случае мы можем сказать feature и затем в этом случае val подойдет потому что у нас есть особенность и затем значение вообще это особенность быть веб-разработчиком чтобы определять как я собираюсь назвать переменную некоторые находят это особенно сложным каждый раз думать о внешнем виде кода и те кто тратит больше времени на это в итоге получают более качественный продукт но позже вы узнаете об этом больше в любом случае feature и val это подойдёт или вы можете использовать key feature если вы думаете что val не очень хорошее название так или иначе я собираюсь поместить key внутри тега strong и затем feature снаружи сейчас если я вернусь обратно в Chrome обновлюсь и вот результат прохождения цикла через каждый элемент массива oK ну давайте сделаем ещё кое-что до того как мы закончим на текущий момент вы создали массив с которым вы ничего не делаете но в реальной жизни вам Иногда нужно удалить значение из массива или прикрепить значения в массив что же вы конечно всегда можете обновить значение в массиве но иногда значение приходят из внешнего мира и вы не можете управлять ими напрямую вместо этого вы просто будете вставлять их так что мы можем сделать следующие добавим новый ключ в этот массив как насчёт имени и затем определим значение так что мы имеем массив person и мы собираемся добавить новое значение в этот массив ключ это name и значение джеффри так что это значит что если мы вернемся обратно и обновим мы можем увидеть новые значения тоже время вы можете подумать что если у меня нет ассоциативного массива если мы вернемся к массиву животное где у нас есть собака и кот так что мы будем делать мы не можем поместить это здесь так что просто укажите новые значения сейчас animals равно собака кот и слон так что постепенно вы работая с кодом время от времени вам захочется узнать текущее значение переменной есть отличный способ сделать это я собираюсь показать вам 2 способа Так что вы знаете echo Если мы хотим узнать значение ключа age давайте за комментируем это небольшое отступление в pHP комментирование это значит что мы не хотим выполнять этот код это хорошо использовать когда вы думаете я не хочу загружать этот шаблон я я просто хочу вывести значения так что я собираюсь закомментировать это используя два слэша и в основном дела Это в редакторе кода это будет помечено серым цветом что означает этот код не будет выполнен так что сейчас мы собираемся вывести возраст персона вернуться обратно и перезагрузить и мы получим 31 если вы хотите вывести весь массив person так что это не будет работать : преобразование массива в строку так что когда вы используете echo работает со строками но мы уже выучили что возраст это строка но массив это не строка так вот то что здесь случилось echo ожидает строку но вместо этого получает массив и показывает нам ошибку Эй я действительно не знаю что делать так что в ситуациях когда вы не работаете с чем-то примитивным вроде строки или Целого числа вы должны использовать функцию под названием var нижнее подчеркивание dump так что Запомните мы знаем немного функциях на текущий момент ну как правило хорошего тона когда мы вызываем ее мы указываем параметр внутри или если функция не принимает входные параметры это будет выглядеть вот так но в нашем случае мы Передаем Так что мы указываем person переменную oK так что если мы вернемся обратно вы увидите что-то подобное всё ещё трудночитаемо здесь есть опция чтобы сделать это более читаемым но в любом случае но всё ещё видим возраст 31 цвет волос коричневый и карьера которая содержит строку из 13 символов веб-разработчик и имя джеффри Такчто заметьте у нас есть несколько строк а также число обозначенная int Так что давайте сделаем это читабельным конечно по мере вашего обучения вы узнаете пакеты которые вы можете подтянуть который стилизуют это красиво вы также можете/ помните у вас всё ещё есть HTML и вы можете использовать что-то подобное обернуть это в pre теги/ чтож Если вы знакомы с pre тегами/ которые отвечают за форматирование/ вы увидите что сейчас это выглядит лучше/ у вас есть множество опций вы даже можете использовать расширение браузера которая будет форматировать Это для вас автоматически вы можете также поместить эту функцию мы поговорим о функциях позже но на текущий момент просто примите что у вас есть var dump функция которая проще говоря выбросит всё что вы хотите так что давайте вернем обратно require Я собираюсь вам показать кое-что ещё так что в этом случае мы выбросили переменную person а затем отобразили свойства в представлении но иногда вам просто хочется выбросить значение и прекратить выполнение кода/ как вроде я хочу увидеть это и я закончил я больше ничего не хочу/ pHP имеет специальную функцию для этого которая называется die так что вы могли видеть это иногда так чтоже происходит здесь так подумайте об этом у нас есть Переменная person мы используем функцию мы выбрасываем это переменую dump это функция но die это также функция так что например мы укажем Привет мир и затем мы увидим здесь Привет мир и заметьте мы не увидели продолжение Так что это неважно что идет после это неважно потому что мы умерли мы не достигли этой точки Так что мы больше ничего не увидим это иногда очень полезно так что в нашем случае мы собираемся умереть после того как выбросим нашу переменую так Что вы пишите var dump person и затем вызываете функцию die сразу же после этого это будет опция или что делают многие люди die и затем var dump person прямо внутри и мы получим тоже самое это так для справки и в следующем уроке мы создадим функцию которую мы назовем например dd которая за кулисами будет делать это для нас это будет полезно/ в итоге вы знаете как создать ассоциативный массив /вы знаете как добавить значение в массив/но как насчет удаления что если мы хотим удалить возраст /окей/ для массива вы можете использовать unset чтобы сделать это unset person age/и сейчас если мы вернямся обратно и обновим заметтье здесь возраст больше не отображается/ Окей вот ваша домашняя работа/представьте что вы готовите задачу/вроде как todo листа/ окей вы можете назвать переменную – таск/и я хочу это сделать равным ассоциативному массиву/который содержит детали о задаче/вклюячая заголовок задачи/например – сходить в торговый центр/дата завершения – сегодня или завтра/кто будет ее выполнять это можете быть вы или ваша жена кто угодно/а также статус выполнения да или нет/это подойдет вы можете использовать строку да или нет для этого/однако в следующем уроке мы собираемся поговорить о логическом типе данных, который больше подойдет для этого/после этого мы поговорим о функциях и после этого мы поговорим о чем то другом/так что увидимся

Step 21

Добро пожаловать

Я собираюсь перейти в файл ядра bootstrap

Вы видите как мы вручную запрашиваем все файлы

Которые нам нужны

Это все еще относительно говоря очень простой проект

Но мы понимаем чно чтостепенно мы запрашиваем больше и больше

Файлов и это действительно болезненно

Каждый раз как я хочу добавить класс

Я нехочу вручную запрашивать файл

Я хочу чтобы это было автоматически

Php подерживает такую функцию

и в прошлом каждый делал это по-своему

и использовать свои собственные способы

, но затем несколько лет назад был опубликован

менеджер зависимостей называемый composer

Это дает вам очень быстрый и эффективный способ тянуть пакеты, построенные остальной частью сообщества,

Например, вам нужно изменить изображение или создать миниатюру

Вы могли бы использовать composer для этих целей

И так далее

мы не собираемся рассматривать каждый из этих вещей в этих сериях

Но есть одна отличная вещь которую composer делает

Это как я говорил ранее - автозагрузчик, включенный в коробочное решение

Давайте вернемся к Chrome и посетим composer.org

сейчас, кстати, если вы думаете про себ

я не хочу беспокоиться об этом Мне это не нужно.

Поверьте мне, это отраслевой стандарт

в данный момент мы просто знакомимся с composer и это не очень тяжело

мы уже несколько раз его освещали в других сериях

так что мы схватим это очень быстро

Перейдем к скачиванию и здесь вы получите список ссылок

Для запуска в терминале

Как только вы закончите, вы должны быть в состоянии запустить composer, и вы получите что-то вроде этого

отлично

окей

так что давайте вернемся к sublime и я собираюсь создать новый composer json-файл

Я собираюсь указать то, как я хочу, чтобы композитор автоматически загружал мои файлы

Теперь конкретно я собираюсь предоставить карту классов, что означает - загружай каждый класс в пределах пути, который я указал

так, например, если мы хотим автоматически загрузить все контроллеры, мы могли бы сделать что-то подобное

Теперь он найдет любые классы в этом каталоге и автоматически загрузит их

или на данный момент в реальной жизни вы можете в конечном итоге сделать это с помощью немного другой Конвенции под названием PSR-4

но для наших нужд мы просто хотим, чтобы был автоматически загружен каждый класс

поэтому я буду использовать это для представления текущего каталога

- автоматическая загрузка любых классов отсюда

окей

Итак, мы имеем начальную установку нашего composer

Я собираюсь сказать, composer install

Теперь, как я сказал, если бы у нас были какие-либо указанные зависимости, composer подтянет их для вас

однако

в нашем случае у нас нет никаких объявленний

так что он просто сделает первоначальную установку и в рамках этого уведомления мы получим автозагрузчик

и если мы углубимся в карту класса – заметьте он имеет ссылку на имя класса,

а также путь, где находится файл

так что теперь composer понимает, как автоматически загружать эти файлы и где они находятся

так что, если мы вернемся к нашему bootstrap файлу, и если мы закомментируем эти строки

Давайте посмотрим

Я вернусь к chrome

Мы хотим чтобы это по прежнему работало

но

нет, построитель запросов не найден

так что это очень важная вещь, которую нужно понять о composer

для того, чтобы убедиться, что он работает

вы должны требовать этот единственный файл

так что вы должны требовать autoloader.php,

а затем в основном это единственный файл,

о котором вам нужно будет беспокоиться

окей

так что, как правило, вы будете делать это начальной

точкой входа в ваше приложение

вы можете установить это здесь

require vendor autoloader.php

просто помните, что при использовании composer

в точках входа убедитесь, что вы требуете autoloader

окей

так что если мы переключимся обратно

эти строки закоментированны и мы не должны требовать их вообще

но если я вернусь к chrome и обновлю страницу

все работает

и это потому, что у нас есть автозагрузчик, работающий на нас

очень крутой

чтобы мы могли избавиться от всего этого хлама вот

и мы в хорошей форме

так

да

На самом деле есть много всего что можно сказать про composer

но я не хочу, чтобы вы беспокоились об этом пока

у нас есть другие серии, которые покажут вам его использование

так что пока

даже если вы только использовали его для его автозагрузки,

это действительно хорошая привычка, чтобы для начала

окей

так что теперь, когда мы очистили наш bootstrap файл

Я хочу обратить наше внимание на это

где мы как бы храним эти зависимости, такие как конфигурационный

файл и построитель запросов в массиве

но вместо этого давайте создадим специальный класс для этого

так что мы можем связать зависимости в него, а затем, когда нам

нужно разрешить зависимости из него, мы называем его простой

контейнер инъекции зависимостей