|  |  |
| --- | --- |
| Top Ten Front-End Design Rules For Developers  As front-end developers, our job is, essentially, to turn designs into reality via code. Understanding, and being competent in, design is an important component of that. Unfortunately, truly understanding front-end design is easier said than done. Coding and aesthetic design require some pretty different skill sets. Because of that, some front-end devs aren’t as proficient in the design aspect as they should be, and as a result, their work suffers.  My goal is to give you some easy-to-follow rules and concepts, from one [front-end dev](https://www.toptal.com/front-end) to another, that will help you go from start to finish of a project without messing up what your designers worked so hard on (or possibly even allowing you to design your own projects with decent results).  Of course, these rules won’t take you from bad to magnificent in the time it takes to read one article, but if you apply them to your work, they should make a big difference. Do Stuff In A Graphics Program It’s truly rare that you complete a project, and go from start to finish while maintaining every single aesthetic mutation in the design files. And, unfortunately, designers aren’t always around to run to for a quick fix.  Therefore, there always comes a point in any front-end job where you end up having to make some aesthetic-related tweaks. Whether it’s making the checkmark that shows when you check the checkbox, or making a page layout that the PSD missed, front-enders often end up handling these seemingly minor tasks. Naturally, in a perfect world this wouldn’t be the case, but I have yet to find a perfect world, hence we need to be flexible.  A good front-end developer has to use professional graphics tools. Accept no substitute.  A good front-end developer has to use professional graphics tools. Accept no substitute.  For these situations, you should always use a graphics program for mockups. I don’t care which tool you choose: Photoshop, Illustrator, Fireworks, [GIMP](https://www.gimp.org/tutorials/), whatever. Just don’t just attempt to design from your code. Spend a minute launching a real graphics program and figuring out how it should look, then go to the code and make it happen. You may not be an [expert designer](https://www.toptal.com/designers/photoshop/portfolios), but you’ll still end up with better results. Match the Design, Don’t Try To Beat It Your job is not to impress with how unique your checkmark is; your job is to match it to the rest of the design.  Those without a lot of design experience can easily be tempted to leave their mark on the project with seemingly minor details. Please leave that to the designers.  Developers have to match the original front-end design as closely as possible.  Developers have to match the original front-end design as closely as possible.  Instead of asking “Does my checkmark look amazing?” you should be asking, “How well does my checkmark match the design?”  Your focus should always be on working with the design, not on trying to outdo it. Typography Makes All the Difference You’d be surprised to know how much of the end look of a design is influenced by typography. You’d be just as surprised to learn how much time designers spend on it. This is not a “pick-it-and-go” endeavor, some serious time and effort goes into it.  If you end up in a situation where you actually have to choose typography, you should spend a decent amount of time doing so. Go online and research good [font pairings](https://femmebot.github.io/google-type/). Spend a few hours trying those pairings and making sure you end up with the best typography for the project.  Is this font right for your project? When in doubt, consult a designer.  Is this font right for your project? When in doubt, consult a designer.  If you’re working with a design, then make sure you follow the designer’s typography choices. This doesn’t just mean choosing the font, either. Pay attention to the line spacing, letter spacing, and so on. Don’t overlook how important it is to match the typography of the design.  Also, make sure you use the right fonts in the correct spot. If the designer uses Georgia for headers only and Open Sans for body, then you shouldn’t be using Georgia for body and Open Sans for headers. Typography can make or break aesthetics easily. Spend enough time making sure you are matching your designer’s typography. It will be time well spent. Front-end Design Doesn’t Tolerate Tunnel Vision You’ll probably be making small parts of the overall design.  Tunnel vision is a common pitfall for front-end developers. Don’t focus on a single detail, always look at the big picture.  Tunnel vision is a common pitfall for front-end developers. Don’t focus on a single detail, always look at the big picture.  An example I’ve been going with is making the checkmark for a design that includes custom checkboxes, without showing them checked. It’s important to remember that the parts you are making are small parts of an overall design. Make your checks as important as a checkmark on a page should look, no more, no less. Don’t get tunnel vision about your one little part and make it something it shouldn’t be.  In fact, a good technique for doing this is to take a screenshot of the program so far, or of the design files, and design within it, in the context in which it will be used. That way, you really see how it affects other design elements on the page, and whether it fits its role properly. Relationships And Hierarchy Pay special attention to how the design works with [hierarchy](https://hackdesign.org/lessons/19). How close are the titles to the body of text? How far are they from the text above them? How does the designer seem to be indicating which elements/titles/text bodies are related and which aren’t? They’ll commonly do these things by boxing related content together, using varying white space to indicate relationships, using similar or contrasting colors to indicate related/unrelated content, and so on.  A good front-end developer will respect design relationships and hierarchy. A great developer will understand them.  A good front-end developer will respect design relationships and hierarchy. A great developer will understand them.  It’s your job to make sure that you recognize the ways in which the design accomplishes relationships and hierarchy and to make sure those concepts are reflected in the end product (including for content that was not specifically designed, and/or dynamic content). This is another area (like typography) where it pays to take extra time to make sure you’re doing a good job. Be Picky About Whitespace And Alignment This is a great tip for improving your designs and/or better implementing the designs of others: If the design seems to be using spacings of 20 units, 40 units, etc., then make sure every spacing is a multiple of 20 units.  This is a really drop-dead simple way for someone with no eye for aesthetics to make a significant improvement quickly. Make sure your elements are aligned down to the pixel, and that the spacing around every edge of every element is as uniform as possible. Where you can’t do that (such as places where you need extra space to indicate hierarchy), make them exact multiples of the spacing you’re using elsewhere, for example two times your default to create some separation, three times to create more, and so on.  Do your best to understand how the designer used whitespace and follow those concepts in your front-end build.  Do your best to understand how the designer used whitespace and follow those concepts in your front-end build.  A lot of devs achieve this for specific content in the design files, but when it comes to adding/editing content, or implementing dynamic content, the spacing can go all over the place because they didn’t truly understand what they were implementing.  Do your best to understand how the designer used whitespace and follow those concepts in your build. And yes, spend time on this. Once you think your work is done, go back and measure the spacing to ensure you have aligned and uniformly spaced everything as much as possible, then try out the code with lots of varying content to make sure it’s flexible.  Начало формы  Конец формы If You Don’t Know What You’re Doing, Do Less I’m not one of those people that thinks every project should use minimalist design, but if you’re not confident in your design chops and you need to add something, then less is more.  Less is more. If your designer did a good job to begin with, you should refrain from injecting your own design ideas.  Less is more. If your designer did a good job to begin with, you should refrain from injecting your own design ideas.  The designer took care of the main stuff; you only need to do minor fillers. If you’re not very good at design, then a good bet is to do as minimal amount as you can to make that element work. That way, you’re injecting less of your own design into the designer’s work, and affecting it as little as possible.  Let the designer’s work take center stage and let your work take the back seat. Time Makes Fools Of Us All I’ll tell you a secret about designers: 90 percent (or more) of what they actually put down on paper, or a Photoshop canvas, isn’t that great.  They discard far more than you ever see. It often takes many revisions and fiddling with a design to get it to the point where they’d even let the guy in the next cubicle see their work, never mind the actual client. You usually don’t go from a blank canvas to good design in one step; there’s a bunch iterations in between. People rarely make good work until they understand that and allow for it in their process.  If you think the design can be improved upon, consult your designer. It’s possible they already tried a similar approach and decided against it.  If you think the design can be improved upon, consult your designer. It’s possible they already tried a similar approach and decided against it.  So how do you implement this? One important method is taking time between versions. Work until it looks like something you like then put it away. Give it a few hours (leaving it overnight is even better), then open it up again and take a look. You’ll be amazed at how different it looks with fresh eyes. You’ll quickly pick out areas for improvement. They’ll be so clear you’ll wonder how you possibly missed them in the first place.  In fact, one of the better designers I’ve known takes this idea a lot further. He would start by making three different designs. Then, he’d wait at least 24 hours, look at them again and throw them all out and start from scratch on a fourth. Next, he’d allow a day between each iteration as it got better and better. Only when he opened it up one morning, and was totally happy, or at least, as close as a designer ever gets to totally happy, would he send it to the client. This was the process he used for every design he made, and it served him very well.  I don’t expect you to take it that far, but it does highlight how helpful time without “eyes on the design” can be. It’s an integral part of the design process and can make improvements in leaps and bounds. Pixels Matter You should do everything in your power to match the original design in your finished program, down to the last pixel.  Front-end developers should try to match the original design down to the last pixel.  Front-end developers should try to match the original design down to the last pixel.  In some areas you can’t be perfect. For example, your control over letter-spacing might not be quite as precise as that of the designer’s, and a CSS shadow might not exactly match a Photoshop one, but you should still attempt to get as close as possible. For many aspects of the design, you really can get pixel-perfect precision. Doing so can make a big difference in the end result. A pixel off here and there doesn’t seem like much, but it adds up and affects the overall aesthetic much more than you’d think. So keep an eye on it.  There are a number of [tools] that help you compare original designs to end results, or you can just take screenshots and paste them into the design file to compare each element as closely as possible. Just lay the screenshot over the design and make it semi-transparent so that you can see the differences. Then you know how much adjustment you have to make to get it spot on. Get Feedback It’s hard to gain an “eye for design.” It’s even harder to do it on your own. You should seek the [input of others](http://graphicdesign.stackexchange.com/questions/19927/where-can-i-go-to-recieve-feedback-on-web-design-projects)to really see how you can make improvements.  I am not suggesting you grab your neighbor and ask for advice, I mean you should consult real designers and let them critique your work and offer suggestions.  Let designers critique your work. Put their criticism to good use and don’t antagonize them.  Let designers critique your work. Put their criticism to good use and don’t antagonize them.  It takes some bravery to do so, but in the end it is one of the most powerful things you can do to improve the project in the short-term, and to improve your skill level in the long run.  Even if all you have to fine tune is a simple checkmark, there are plenty of people willing to help you. Whether it’s a designer friend, or an online forum, seek out qualified people and get their feedback.  Build a long-lasting, productive relationship with your designers. It’s vital for useful feedback, quality, and execution.  Build a long-lasting, productive relationship with your designers. It’s vital for useful feedback, quality, and execution.  It may sound time consuming, and may cause friction between you and your designers, but in the big scheme of things, it’s worth it. Good front-end developers rely on valuable input from designers, even when it’s not something they like to hear.  Therefore, it’s vital to build and maintain a constructive relationship with your designers. You’re all in the same boat, so to get the [best possible results](https://www.toptal.com/designers/freelance/portfolios) you have to collaborate and communicate every step of the way. The investment in building bonds with your designers is well worth it, as it will help everyone do a better job and execute everything on time. Conclusion To summarize, here is a short list of design tips for front-end developers:   * Design in a graphics program. Don’t design from code, not even the small stuff. * Match the design. Be conscious of the original design and don’t try to improve it, just match it. * Typography is huge. The time you spend making sure it’s right should reflect its importance. * Avoid tunnel vision. Make sure your additions stand out only as much as they should. They’re not more important just because you designed them. * Relationships and hierarchy: Understand how they work in the design so that you can implement them properly. * Whitespace and alignment are important. Make them accurate to the pixel and make them evenly throughout anything you add. * If you’re not confident in your skills, then make your additions as minimally styled as you can. * Take time between revisions. Come back later to see your design work with fresh eyes. * Pixel-perfect implementation is important wherever possible. * Be brave. Seek out experienced designers to critique your work.   Not every front-end developer is going to be a fantastic designer, but every front-end dev should at least becompetent in terms of design.  You need to understand enough about design concepts to identify what’s going on, and to properly apply the design to your end product. Sometimes, you can get away with blind copying if you’ve got a thorough designer (and if you’re detail oriented enough to truly copy it pixel for pixel).  However, in order to make large projects shine across many variations of content, you need some understanding of what’s going through the designer’s head. You don’t merely need to see what the design looks like, you need to know why it looks the way it does, and that way you can be mindful of technical and aesthetic limitations that will affect your job.  So, even as a front-end developer, part of your regular self-improvement should always include learning more about design. | 10 лучших правил по фронтенд-дизайну для разработчиков  По сути, работа фронтенщиков сводится к тому, чтобы «оживить» дизайны с помощью кода. Понимание и компетентность в вопросах дизайна – вот ключевые компоненты первоклассной работы. К сожалению, говорить о понимании фронтенд дизайна намного легче, чем делать. Написание кода и эстетичность дизайна требуют от исполнителей диаметрально противоположных навыков. Многие фронтенщики попадают впросак, из-за недостаточности знаний в области дизайна, что негативно сказывается на всей их работе.  Моя цель – донести до вас определенный свод правил и концепций (от одного фронтенщика другому, так сказать), которые помогут вам выполнять свой проект от начала и до конца, не нанося урон кропотливой работе дизайнеров. Кто знает, быть может, эта статья поможет вам самим придумать отличный дизайн для дизайн собственных проектов.  Все мы понимаем, что никто не сможет превратиться из новичка в матерого фронтенщика за одну статью. Но постарайтесь применять полученные знания на практике, и вы увидите существенную разницу. Работа в графической программе Мало когда весь проект ведется целиком в дизайн-файлах с сохранением всех эстетических вариаций. И, к сожалению, дизайнеры не всегда прибегают по первому зову для небольших доработок.  Поэтому любой фронтенщик рано или поздно оказывается в ситуации, когда ему самому приходится вносить какие-то эстетические правки. Это может быть что угодно, начиная с галочки-отметки о выборе чекбокса и заканчивая созданием макета страницы, отсутствующей в PSD. Суть в том, что фронтенщик остается один на один с этими, казалось бы, несущественными доработками. В идеальном мире такого, конечно же, никогда бы не произошло. Но давайте для начала попробуем отыскать этот идеальный мир! А посему, самое время учиться гибкости.  A good front-end developer has to use professional graphics tools. Accept no substitute.  Хороший фронтенд-разработчик должен уметь работать с профессиональными графическими инструментами. Не соглашайтесь на замену.  Для таких ситуаций вам необходимо использовать графические программы для разметки. Мне все равно, что это будет: Photoshop, Illustrator, Fireworks, [GIMP](https://www.gimp.org/tutorials/), да что угодно. Просто не пытайтесь менять дизайн только кодом. Потратьте лишнюю пару минут на запуск графической программы и набросайте эскиз дизайна в ней. А вот тогда уже можно будет вернуться к коду и продолжить процесс там. Возможно, вы и не [супер-дизайнер](https://www.toptal.com/designers/photoshop/portfolios), но результативность у вас явно повысится. Следуйте дизайну, не пытайтесь его превзойти Ваша работа заключается не в том, чтобы сразить наповал окружающих креативностью придуманной вами галочки, а сделать так, чтобы эта самая галочка не выбивалась из общей дизайн-концепции.  Обладатели скудного опыта в дизайне подсознательно будут стараться так или иначе оставить свой след в каких-то незначительных деталях проекта. Пожалуйста, оставьте дизайн для дизайнеров.  Developers have to match the original front-end design as closely as possible.  Разработчики должны неукоснительно придерживаться оригинального фронтенд-дизайна.  Вместо вопроса: «Ну разве не классную галочку я придумал?», лучше спросите себя: «Насколько хорошо моя галочка вписывается в общий концепт?».  Вы должны работать с дизайном, а не пытаться его переделать. Типографика решает все Вы несказанно удивитесь тому, как же сильно конечный дизайн проекта зависит от типографики. Но куда больше вас удивит то, сколько времени тратят дизайнеры на изучение шрифтов. Подход «выбрал и пошел» тут явно не работает. Вам придется потратить много времени и нервов.  Если вдруг вы оказались в ситуации, когда именно вам приходится выбирать типографику, мой вам совет – не спешите! Поищите в интернете хорошие [комбинации шрифтов](https://femmebot.github.io/google-type/). Потратьте n-ное количество часов на тестирование этих комбинаций и лишний раз удостоверьтесь в том, что для своего проекта вы выбрали наилучшую типографику.  Is this font right for your project? When in doubt, consult a designer.  Подойдет ли этот шрифт для проекта? При любых сомнениях – обращайтесь к дизайнеру.  Если вы работаете с дизайном, проверьте ваши действия на соответствие дизайнерскому видению типографики. Одним выбором шрифта дело не ограничится. Обращайте внимание на интервалы между строками, буквами и проч. Не игнорируйте всю важность выбора типографики для проекта.  Тут важна еще одна вещь: правильные шрифты должны использоваться в правильном месте. Если дизайнер использует Georgia только в заголовках, а Open Sans – в остальном тексте, то не следует делать заголовки Open Sans, а основной текст –Georgia. Типографика может с легкостью создавать и разрушать эстетичность дизайна. Опять же, внимательно сверяйте получившийся результат с типографикой дизайнера. Ваши временные затраты с лихвой окупятся. Фронтенд-дизайн не терпит ограниченности взглядов Весьма вероятно, что вам поручат разработки каких-то мелких деталей совокупного дизайна.  Tunnel vision is a common pitfall for front-end developers. Don’t focus on a single detail, always look at the big picture.  Ограниченность взглядов – бич фронтенщиков. Не концентрируйтесь на чем-то одном; смотрите на картину в целом.  Вернемся к нашему примеру. Фронтенщика просят разработать отметку для дизайна кастомизированного чекбокса, но только чтобы не было видно, что этот чекбокс был выбран. Почаще напоминайте себе: то, что делаете вы, – лишь небольшой элемент совокупного дизайна. Уделяйте одинаковое количество времени созданию и чекбоксов, и галочек. Не зацикливайтесь только на том, что придумали вы; не раздувайте из мухи слона.  Между прочим, знаю я одно проверенное средство: делаете скриншоты окна программы или дизайн-файлов и для полноты картины вносите правки уже в них. Так вы сможете наглядно увидеть, как та или иная доработка влияет на все прочие элементы дизайна, и насколько правильно она выполняет свою роль. Связи и иерархия Внимательно изучите [иерархичность](https://hackdesign.org/lessons/19) дизайна. Как близко расположены заголовки от остального текста? А какое расстояние между ними и предыдущим абзацем? Как дизайнер показывает, какие именно элементы/заголовки/абзацы связаны между собой, а какие – нет? Как правило, связанный контент добавляется в виде боксов, разделенных белым пространством. Для этих же целей можно использовать и цветовую палитру, объединяя элементы схожими оттенками, а разграничивая контрастными цветами.  A good front-end developer will respect design relationships and hierarchy. A great developer will understand them.  Хороший фронтенщик уважает связи и иерархии в дизайне. Отличный фронтенщик не только уважает, но и разбирается в них.  Вы должны понять, как именно реализуются связи и иерархии в дизайне, и сделать все, чтобы ваш конечный продукт сохранил их в первозданном виде (включая содержимое, для которого не был задан собственный дизайн и/или динамическое содержимое). Это еще одна область (сродни типографике), которая требует от разработчика дополнительных временных затрат. Учитывайте белое пространство и выравнивание Вот вам отличный совет для улучшения собственного дизайна или более правильной реализации чужого: если в макете используется интервал в 20, 40, …, n единиц, то тогда каждый интервал должен увеличиваться на значение в 20 единиц.  Этот до боли простой метод как нельзя лучше подходит для профанов в области эстетики. Выравнивайте ваши элементы вплоть до пикселя, а интервал вокруг каждой границы каждого элемента сделайте максимально единообразным. В тех случаях, когда сделать это невозможно (например, когда вам требуется дополнительное пространство для визуального подчеркивания иерархии), увеличивайте интервал пропорционально его значению в остальном тексте. Например, для визуальной отбивки используйте удвоенное значение интервала по умолчанию, для большего разделения – больший интервал.  Do your best to understand how the designer used whitespace and follow those concepts in your front-end build.  Приложите максимум усилий, чтобы понять концепцию использования белого пространства дизайнером, и неукоснительно следуйте ей в программировании.  Многие разработчики понимают эту схему на примере какого-то конкретного содержимого в дизайн-файлах. Но как только дело доходит до добавления/изменения контента или внедрения динамического содержимого, мысли программиста путаются, он перестает понимать, что именно от него хотят. И в результате все интервалы по тексту слетают.  Вы должны не только понять, как дизайнер использует белое пространство в макете, но и уметь повторять данный алгоритм при разработке. И да, потратить на это какое-то время. Как только вам кажется, что работа сделана, вернитесь к вашему макету и измерьте белое пространство. Убедитесь в том, что вы использовали *одинаковый интервал* и *выравнивание* везде, где только можно. А затем потестируйте свой код на предмет *гибкости*, добавляя в шаблон разнообразное содержимое.  Начало формы  Конец формы Не надо лезть в то, в чем вы не разбираетесь Я не из тех, кто считает, что в каждом дизайн-проекте следует придерживаться минимализма. Но если вы не уверены в своих дизайнерских способностях и хотите что-то добавить, то начните лучше с малого.  Less is more. If your designer did a good job to begin with, you should refrain from injecting your own design ideas.  Меньше – да лучше. Если ваш дизайнер хорошо справился со своей работой, то не стоит влезать в макет со своими дизайнерскими идеями.  Дизайнер позаботился обо всех важных аспектах; от вас требуется доработка лишь незначительных элементов. Если ваши познания в дизайне оставляют желать лучшего, то рекомендуется свести свое вмешательство к минимуму и просто сделать так, чтобы нужный элемент заработал. Так вы меньше «наследите» в работе дизайнера и оставите свое влияние на внешний вид макета минимальным.  Пусть работа дизайнера займет место на сцене, а ваши труды будут вестись из закулисья. Время дурачит каждого Открою вам секрет о дизайнерах: 90% (а, может, и больше!) того, что они отрисовали на бумаге или на канве Фотошопа, никуда не годится.  Процент выбраковки в их работах куда больше, чем это кажется нам с вами. Перед тем, как показать эскиз коллеге за соседним столом, дизайнер миллионы раз перепроверит и доработает свою работу. Что уж говорить о клиентах. Мало кому удается взять пустую канву и сразу же слепить из нее отличный макет. На пути к этому предстоит преодолеть кучу шагов и итераций. Хорошая работа не падает с неба. Вначале человек должен понять, что идет в нужном направлении, и лишь тогда процесс сдвигается с мертвой точки.  If you think the design can be improved upon, consult your designer. It’s possible they already tried a similar approach and decided against it.  Если вам кажется, что дизайн необходимо доработать, обсудите это с дизайнером. Вполне возможно, что ваши варианты уже были опробованы и забракованы.  А как же это сделать? Самое правильное в этой ситуации – брать перерывы между проверками различных версий. Работайте со своим эскизом до тех пор, пока он не станет нравиться вам самим. А затем забудьте о нем на какое-то время. Подождите пару часов (а еще лучше – целую ночь). Потом откройте и заново изучите его. Сами удивитесь тому, насколько разным будет ваше восприятие макета свежим взглядом. И вы сразу поймете, что именно и где требует доработки. Причем, все недочеты будут так сильно бросаться в глаза, что вы сами начнете гадать: и как же я мог не заметить этого раньше?  Один из лучших дизайнеров, которых я когда-либо встречал, пошел еще дальше. Сначала он придумывает три варианта оформления. На следующий день он окидывает макеты свежим взглядом, удаляет их и набрасывает четвертую версию дизайна. И каждое последующее изменение он претворяет суточным перерывом. Так его дизайн становится все лучше и лучше. И лишь в тот момент, когда проснувшись поутру, он открывает свой эскиз, и картинка ему нравится, дизайн отправляется клиенту. Такой подход в работе используется им постоянно и еще ни разу не подводил.  Я не жду от вас *столь радикальных* мер. Однако этот пример как нельзя лучше иллюстрирует проблему «замыленности глаза» и показывает, насколько полезным бывает на час-другой отвлечься от работы. Это важная часть люого дизайн-процесса, помогающая улучшать свой проект стремительными темпами. Пиксели имеют значение Вы должны сделать все возможное, чтобы готовый проект соответствовал первоначальному дизайну до самого последнего пикселя.  Front-end developers should try to match the original design down to the last pixel.  Фронтенщики должны соблюдать оригинальный дизайн до последнего пикселя.  Временами сделать это невозможно. Например, ваш буквенный интервал может несколько отличаться от задумки дизайнера, а тень в CSS – иметь немного другие начертания, чем в Photoshop. Но даже в этом случае старайтесь максимально придерживаться оригинальной концепции. Во многих аспектах дизайна вы можете работать с пиксельной точностью. А для конечного результата это важно. Пиксель здесь, пиксель там – казалось бы, что здесь такого? Но общая эстетичность макета меняется сильно. Так что следите за пикселями.  Сравнивать конечный результат с оригинальным дизайном можно по-разному. Например, с помощью специализированного инструментария. Но проще всего будет сделать скриншоты проекта, добавить их в дизайн-файлы и досконально пройтись по каждому элементу шаблона. Для этого вам потребуется наложить скриншот поверх слоя с дизайном-исходником и сделать верхний слой полупрозрачным – так вы сразу сможете понять, какие именно корректировки необходимо внести и где. Получайте обратную связь Крайне трудно развивать дизайнерское чутье. Еще труднее делать это в одиночестве. [Интересуйтесь мнением окружающих](http://graphicdesign.stackexchange.com/questions/19927/where-can-i-go-to-recieve-feedback-on-web-design-projects), и вы получите взгляд со стороны.  Я не предлагаю вам стучаться в дверь соседу и просить о помощи. Я говорю о том, чтобы консультироваться с настоящими дизайнерами, которые могли бы указать на недочеты вашего дизайна и подсказать способы их исправления.  Let designers critique your work. Put their criticism to good use and don’t antagonize them.  Разрешите дизайнерам критиковать вашу работу. Извлекайте пользу из их критики и не воспринимайте ее в штыки.  Никому не нравится, когда критикуют их труд. Но, в конце концов, это самый действенный способ улучшить проект в кратчайшие сроки и повысить свои профессиональные навыки на будущее.  И пускай вас попросили настроить одну простую галочку. Поверьте, многие люди готовы прийти вам на помощь. Вы можете обратиться за советом к другу дизайнеру, или задать вопрос пользователям онлайн-форума. Ищите профессионалов в сфере дизайне и интересуйтесь их мнением.  Build a long-lasting, productive relationship with your designers. It’s vital for useful feedback, quality, and execution.  Установите крепкие и продуктивные отношения с дизайнерами. Это важно для получения обратной связи, качества работы и мастерства исполнения.  Предложенные решения могут показаться вам времязатратными и приводящими к напряженности в отношениях с дизайнерами, но в итоге оно того стоит. Хороший фронтенщик прислушивается к мнению дизайнеров, пусть даже временами это самое мнение слышать не так уж и приятно.  Вы должны уметь выстраивать отношения и поддерживать конструктивный диалог с дизайнерами. Все вы находитесь в одной лодке. Для достижения [наилучших результатов](https://www.toptal.com/designers/freelance/portfolios) вам необходимо сотрудничать и обсуждать каждый шаг в реализации проекта. Выстраивание должных отношений с дизайнерами окупит себя с лихвой и положительно скажется на времени выполнения проекта и его качестве. Заключение Подводя итог, можно составить небольшой список советов для фронтенщиков на поприще дизайна:   * Создавайте дизайн в графической программе. Не используйте код даже для разработки самых незначительных деталей. * Следуйте дизайну. Уважительно относитесь к исходному дизайну, не пытайтесь изменить или как-то усовершенствовать его. Просто копируйте. * Типографика важна. Время, затраченное вами на выбор идеальной комбинации, во всей красе показывает всю важность проблематики. * Избегайте субъективности. Не считайте придуманные вами элементы важнее всех прочих; они должны занимать отведенное им место.. * Связи и иерархия: разберитесь, *как* они работают в дизайне, и реализуйте их правильно. * Неиспользуемое пространство и выравнивание важны. Повторяйте их с точностью до пикселя и выравнивайте содержимое одинаково. * Если не уверены в своих силах, – сведите свое вмешательство в дизайн к минимуму. * Перед тем, как выбрать какой-то вариант, сделайте небольшой перерыв. Возвращайтесь к дизайну спустя какое-то время и оценивайте его свежим взглядом. * Реализовывайте дизайн с точностью до последнего пикселя везде, где только можно. * Будьте смелыми. Ищите опытных дизайнеров, которые смогут указать вам на недочеты.   Не каждый фронтенщик сможет стать потрясающим дизайнером. Но каждый из нас должен быть *компетентен* в вопросах дизайна.  Для понимания происходящего, вам необходимо хорошо разбираться в концепциях дизайна и правильно применять их при разработке конечного продукта. Иногда можно ограничиться слепым копированием, но только том случае, если вы работаете со скрупулезным дизайнером, и сами внимательны к мелочам до такой степени, что сможете скопировать пиксель к пикселю.  Если вы хотите, чтобы ваши крупные проекты успешно «выстрелили» на фоне окружающего, то необходимо хотя бы в общих чертах понимать, что творится в голове у дизайнера. Вам не нужно мысленно воспроизводить картинку идеального дизайна, но вот понимать, почему он выглядит именно так, а не иначе, вы должны. Это поможет вам разобраться в технических и эстетических ограничениях, которые, так или иначе, влияют на вашу работу.  Пускай вы и фронтенщик, но вы всегда можете совершенствовать свои познания в дизайне на регулярной основе. |

Источник: <https://www.toptal.com/front-end/front-end-design-principles>

