



16 лучших CSS-фреймворков/библиотек для front-end разработчиков

Описание

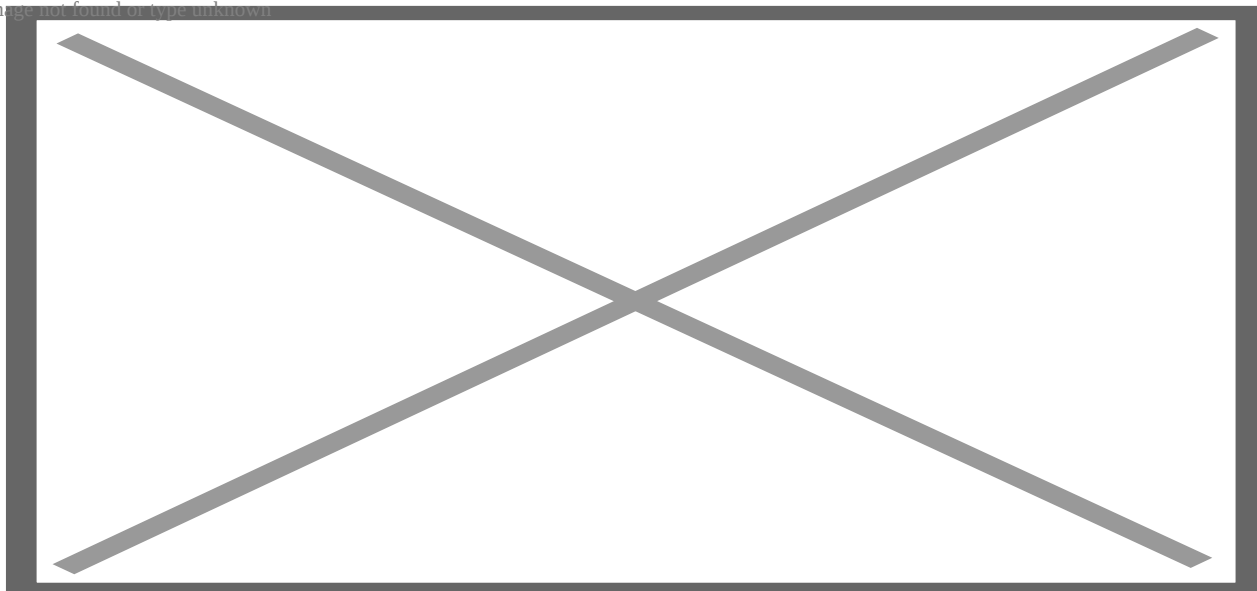
Однажды я услышал шутку одного front-end разработчика: “NASA высадило роботов на Марсе, а мы все еще пытаемся выровнять наши div по центру!”. И мне становится не по себе от того, что в этой шутке есть большая доля правды. Сделать что-то, что звучит так просто и продиктовано здравым смыслом, как центрирование коробки внутри коробки, невероятно сложно в CSS. Если только вы не сталкивались с этим раньше. И не сохранили где-то фрагмент кода. И даже если вам удастся это сделать, всегда есть таящийся страх, что это может ужасно сломаться в каком-нибудь дурацком браузере! CSS занимает первое место в списке “необходимых веб-злодеев” наряду с JavaScript. Это стандарт, который развивался бессистемно, по-разному интерпретировался разными производителями браузеров и сейчас настолько полон противоречий, что никто не осмеливается называть себя “экспертом по CSS”.

Неудивительно, что со временем появились CSS-фреймворки, которые избавили нас от большинства проблем. Сегодня мы не можем представить себе кодирование без нашего любимого CSS-фреймворка, поскольку ориентация на несколько размеров экрана стала необходимостью. Но как понять, что ваш фреймворк лучше всего подходит для конкретной задачи? Кроме того, если вы новичок в разработке front-end, какой фреймворк вам лучше выбрать? В этом посте мы окинем взглядом ландшафт фронтенд-разработки и сравним лидеров среди CSS-фреймворков. Так что если вы устали от ручного кодирования правил CSS, погрузитесь в эту статью, чтобы получить быстрое облегчение!

Bootstrap

Инициатива компании Twitter, Bootstrap – это заслуга внедрения отзывчивого дизайна в широком масштабе. Это был первый фреймворк, который продвигал философию “mobile-first”. Больше не нужно было разрабатывать дизайн для небольших размеров экрана как отдельный проект; достаточно было включить соответствующие классы Bootstrap, и дизайн автоматически подстраивался под различные размеры экрана (ну, почти).

Image not found or type unknown



Отзывчивый дизайн в Bootstrap (4.0 против 3.0)

Bootstrap добился отзывчивого дизайна, введя идею сетки. Сетка – это невидимое разделение экрана на колонки (вместе с шириной). Например, если у вас есть три

“коробки”, которые вы хотите расположить рядом на больших экранах, но вертикально на маленьких, вот что вы сделаете:

```
<div class="container">
  <div class="row">
    <div class="col-md">
      One of three columns
    </div>

    <div class="col-md">
      One of three columns
    </div>

    <div class="col-md">
      One of three columns
    </div>
  </div>
</div>
```

Текущей популярной версией Bootstrap является 4, которая была существенно переработана по сравнению с серией 3.3. Приведенный выше синтаксис – это то, как вы бы написали код в Bootstrap 4, который во многом обязан своей элегантностью сырой силе Flexbox и другим современным функциям верстки, поддерживаемым непосредственно браузерами. В младших версиях Bootstrap сетка определялась как 12 колонок, что приводило к такому коду, как

```
<div class="col-md-6 col-lg-4"></div>
```

чтобы div занимал треть ширины экрана на устройствах большого размера и половину ширины на устройствах среднего размера. Синтаксис теперь намного приятнее, хотя и требует знакомства с Flexbox.

Плюсы Bootstrap

В Bootstrap есть много того, что может понравиться, особенно разработчикам полного цикла:

- **Быстрое создание прототипов:** С Bootstrap почти не нужно тратить время на размышления о сложном позиционировании CSS и несовместимости с браузерами. Достаточно написать HTML, а затем применить соответствующие классы CSS, чтобы отзывчивость ожила.
- **Большая экосистема:** На сегодняшний день Bootstrap имеет самую большую экосистему среди front-end фреймворков. Количество макетов сайтов, тем,

панелей администратора, компонентов пользовательского интерфейса и т.д., созданных с использованием Bootstrap, просто умопомрачительно, и оно продолжает улучшаться. Для консультантов и производителей продуктов это означает, что готовые продукты и поддержка сообщества всегда будут в избытке.

- **При поддержке Twitter:** Наметившаяся тенденция в области открытого исходного кода – рост проектов, спонсируемых коммерческими организациями. Чаще всего эти организации строят прибыльный бизнес на основе своих предложений. Kotlin (JetBrains), WordPress (Automattic, Inc.), Angular (Google), React (Facebook) и т.д. – вот некоторые примеры. Когда проект поддерживается авторитетной организацией и не является шоу одного человека, это дает сообществу (особенно корпоративным клиентам) веру в то, что у проекта будет четкая дорожная карта и долгосрочное будущее.
- **Большая коллекция компонентов:** Bootstrap предлагает из коробки почти все компоненты пользовательского интерфейса, которые могут вам понадобиться. Навигация, формы, карточки, модалы, кнопки, значки, прогресс-бары, оповещения... Назовите это, и в Bootstrap это есть. Для многих компаний это практически сводит на нет необходимость иметь специальную команду front-end специалистов.
- **Поддержка LESS и SASS:** Среди множества популярных CSS-фреймворков Bootstrap – единственный, который поддерживает как LESS, так и SASS. Да, я знаю, вы не используете LESS (как и любой уважающий себя разработчик, не так ли?), но эй, есть огромные проекты, которые полагаются на LESS. Конечно, вы можете выбрать ни то, ни другое и писать свои обычные CSS-файлы.

Минусы Bootstrap

Ничто не имеет цены, не так ли? Что ж, Bootstrap не является исключением. Со временем Bootstrap попал под сильный огонь со стороны дизайнеров и экспертов по пользовательскому интерфейсу. Вот почему:

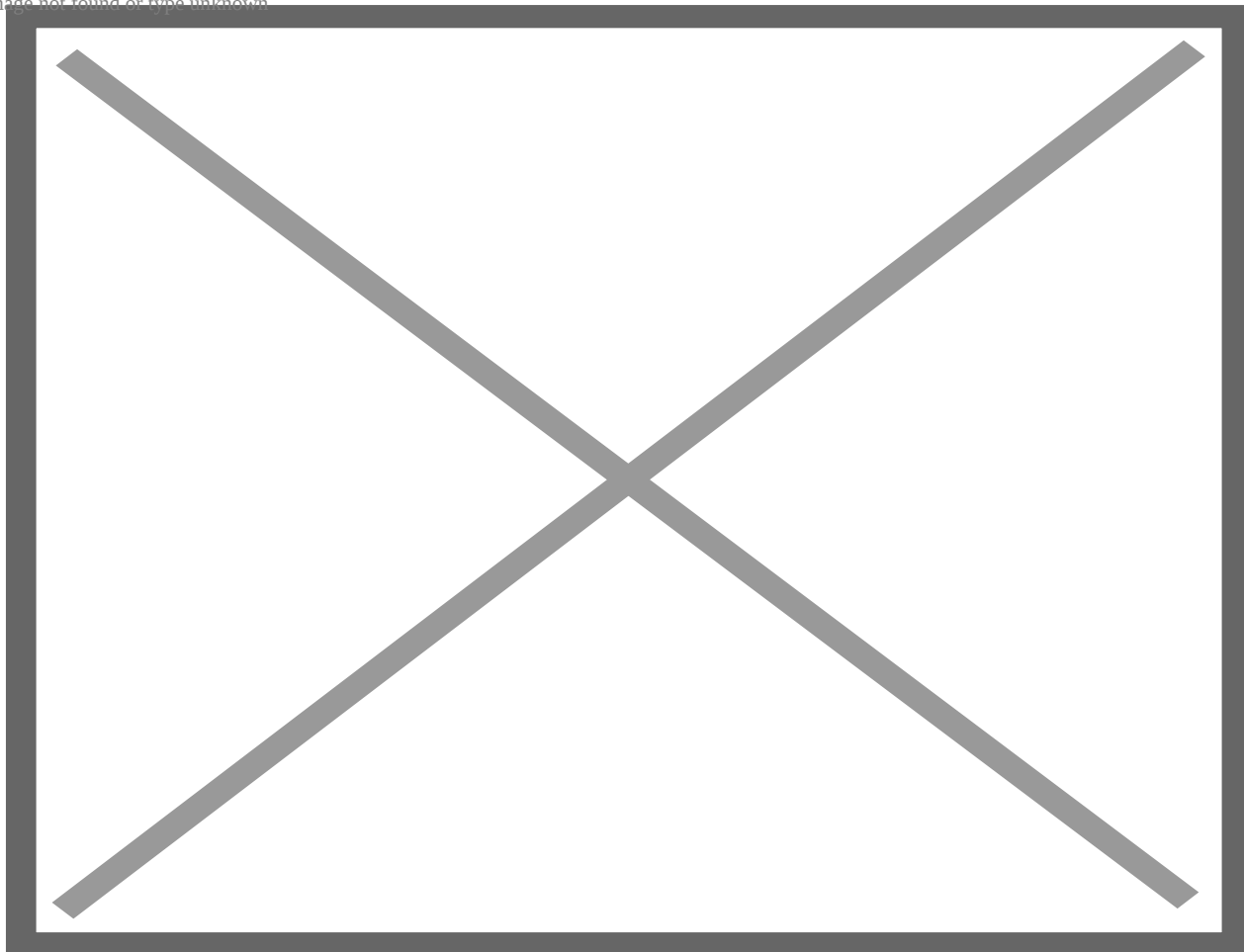
- **Однообразие UX:** Сам факт того, что Bootstrap имеет такую большую коллекцию встроенных элементов, приводит к тому, что сайты выглядят слишком знакомыми и, честно говоря, скучными. Достаточно заглянуть в официальные примеры, чтобы увидеть, насколько удручающе выглядят стандартные настройки. Просто наберите в поисковике “все сайты на bootstrap выглядят одинаково”, и вы поймете, что я имею в виду.

- **Проблемы со стилем:** Bootstrap – это то, что можно считать фреймворком, имеющим свое мнение. Другими словами, у него есть свои представления о макетах, и он заставляет вас прилагать дополнительные усилия, если вы хотите, чтобы он выглядел или вел себя по-другому. Рассмотрим стандартные точки разрыва CSS для ширины экрана: экран среднего размера для Bootstrap – это тот, который начинается с ширины устройства в 768px. А что, если вы хотите ориентироваться, скажем, на предел в 600px? Что ж, удачи вам! То же самое почти с каждым другим компонентом в Bootstrap: строки и контейнеры имеют стандартные отступы, кнопки имеют цвета и границы, которые очень сложно изменить без большой работы, и так далее.

Foundation

Если бы технологии были религиями, то парни из Foundation и Bootstrap жаждали бы крови друг друга. Ни одно обсуждение современных CSS-фреймворков не будет полным без упоминания Foundation, так что мы начинаем.

Image not found or type unknown



Зайдите на сайт Foundation, и вы не сможете не заметить примечание: “Самый передовой отзывчивый фронтенд-фреймворк в мире”. На первый взгляд, это выглядит как громкое заявление для маркетинговой кампании. Однако приверженцы фреймворка Foundation знают, что в этом есть хотя бы доля правды. Foundation был разработан, чтобы естественно сочетаться с фреймворком Rails, и несколько “дзен-подобных” руководящих принципов Rails можно увидеть в работе.

Например, если вам нужна строка, содержащая два элемента на маленьких экранах, три на средних и четыре на больших, то эквивалентный код в Foundation будет выглядеть следующим образом:

```
<ul class="small-block-grid-2 medium-block-grid-3 large-block-grid-4">
  <li><!-- Your content goes here --></li>
  <li><!-- Your content goes here --></li>
  <li><!-- Your content goes here --></li>
  <li><!-- Your content goes here --></li>
  <li><!-- Your content goes here --></li>
  <li><!-- Your content goes here --></li>
  <li><!-- Your content goes here --></li>
</ul>
```

По сравнению с предыдущими версиями Bootstrap, я нахожу это очень интуитивным и легким для запоминания. Больше никаких двенадцатиколоночных сеток и выяснения того, что должно быть 4/12! Хотя Foundation гораздо менее популярна, чем Bootstrap, она является коммерческим секретом для многих опытных разработчиков.

Плюсы фреймворка Foundation

Foundation обладает некоторыми необычными характеристиками из всех CSS-фреймворков, которые мы рассмотрим в этой статье:

- **Полный инструментарий:** Технически неправильно говорить, что Foundation – это CSS-фреймворк. То есть, это так, но он был создан как большая и модульная коллекция инструментов, направленная на решение практически всех видов проблем front-end. Существуют отдельные предложения фреймворка для веб-сайтов и электронной почты, в значительной степени оптимизированные для соответствующих доменов. Foundation также поставляется с интерфейсом командной строки (CLI), что покажется музыкой для разработчиков, привыкших работать с Webpack или другими модульными бандлерами.
- **Чрезвычайная гибкость:** В отличие от Bootstrap, Foundation был создан для того, чтобы дать разработчикам фронтенда полный контроль над их пользовательскими интерфейсами. В результате Foundation покажется новичку безвкусным и чрезвычайно сложным. Однако причина в том, что Foundation не навязывает вам какой-либо язык стилей, а стремится быть именно тем, чем он является: превосходным CSS-фреймворком.
- **Больше, чем просто компоненты пользовательского интерфейса:** Хотя Foundation имеет обычную коллекцию элементов пользовательского интерфейса, он выходит далеко за рамки своих возможностей. Разработчики включили в него продвинутую отзывчивую систему изображений, компонент

таблицы цен (да, тот самый, который используется для отображения различных тарифных планов), валидацию форм, поддержку ориентации справа налево, отзывчивые вставки и многое другое. Я хотел бы еще раз подчеркнуть, что это излишество для большинства простых сайтов, но для крупных сайтов это преимущество, которое признают опытные разработчики.

- **Обучение и консалтинг:** Хотя Bootstrap создан компанией Twitter, это побочный проект и очень маленькая часть общей картины. Однако компания, стоящая за Foundation (ZURB), стремится использовать, развивать и продвигать его. Для крупных клиентов предлагаются учебные курсы и профессиональные консультации, что отлично подходит для компаний, которые нацелены на масштабные проекты и готовы платить.

Минусы фреймворка Foundation Framework

Сильные стороны одного фреймворка становятся его слабыми сторонами при взгляде с противоположной точки зрения. Вот почему Foundation может быть не лучшим выбором для вашего проекта:

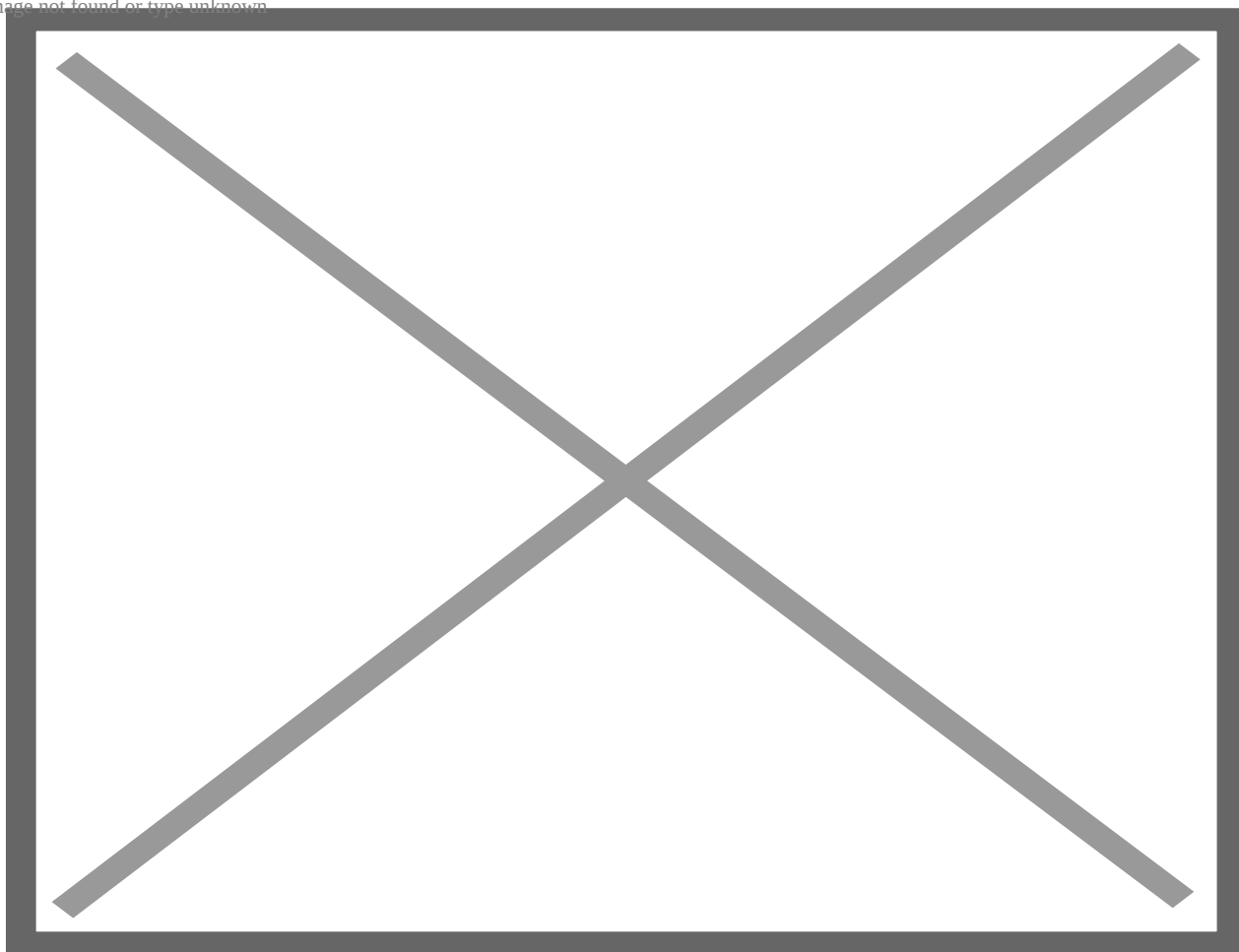
- **Маленькое(е) сообщество:** Сообщество Foundation намного меньше, чем сообщество Bootstrap, и если вы пробуете что-то экзотическое и застряли, шансы найти необходимую помощь ниже. Тем не менее, я бы добавил, что для всех практических целей сообщества достаточно. Просто оно на несколько порядков меньше, чем у Bootstrap, поэтому вы не сможете найти решение мгновенно.
- **Сложность:** Если вы привыкли к Bootstrap или чему-то простому, или, что еще хуже, к ванильному CSS, Foundation покажется вам бесконечным взрывом сложности. Слои в слоях, компоненты с компонентами, бесконечные возможности настройки. . . Очень скоро вы начнете сомневаться в целесообразности самой жизни! Но, опять же, у Foundation совсем другая цель, и его нельзя за это винить.
- **Слишком много опций:** Иногда вы просто хотите сделать что-то, а о совершенстве беспокоиться позже. В такие моменты расстраивает, когда вам предлагают слишком много вариантов с незначительными вариациями. Например, подумайте о необходимости заказать сэндвич из Subway, когда вы так голодны, что могли бы съесть грязь. Естественно, Foundation не для таких случаев.
- **Доступность талантов:** Поскольку Foundation (гораздо) менее популярен, чем

Bootstrap, доступных талантов гораздо меньше. Как правило, любой новый сотрудник, скорее всего, будет знать Bootstrap, но не будет иметь представления о Foundation. Обучение требует времени, а это роскошь, которую не все команды могут себе позволить.

Bulma

Bulma – это относительно новый участник битвы CSS-фреймворков, который за короткое время сделал себе имя. Ее привлекательность заключается в строгом подходе, основанном только на CSS (нет компонентов JavaScript), и элегантных настройках по умолчанию, что является тем, с чем многие разработчики с хорошим чувством дизайна сталкиваются при работе с Bootstrap.

Image not found or type unknown



Во многом импульс Bulma обусловлен высокими темпами внедрения в сообществе Laravel (PHP-фреймворк, если вы не знали), что, я уверен, в значительной степени помогло Vue.js подняться на вершину популярности среди JavaScript-фреймворков.

Почему стоит выбрать Bulma CSS Framework

Есть много причин полюбить Bulma и использовать его для вашего следующего проекта:

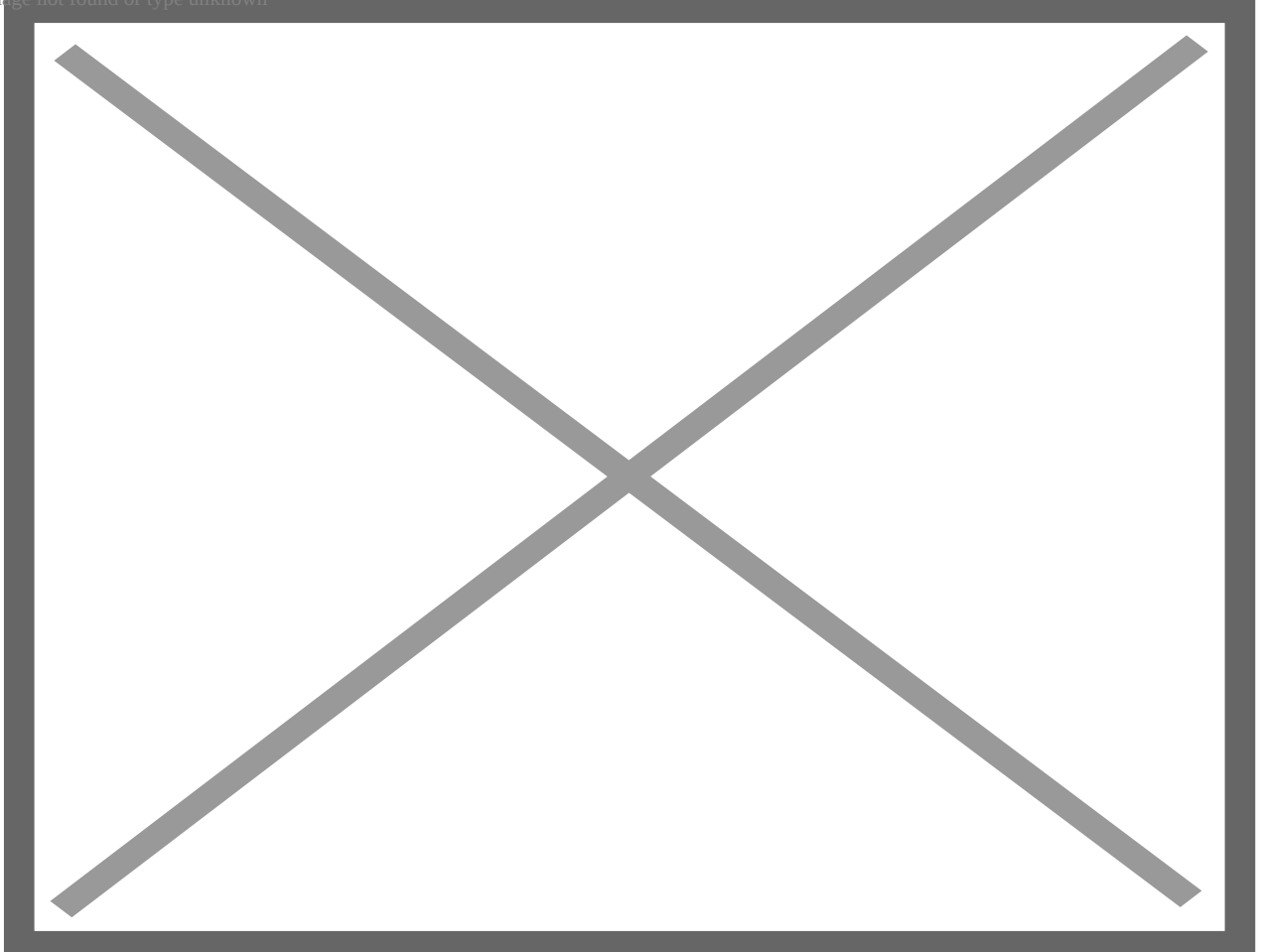
- **Довольно популярна:** Ладно, он не популярнее Bootstrap, но он популярнее Foundation. На момент написания статьи у Bulma 30k+ звезд на Github, что примерно на 3k+ больше, чем у Foundations. Конечно, количество звезд на Github не является показателем достоинств, но это говорит о том, что сообщество одобряет Bulma.
- **Очень удобные для чтения классы:** Для меня Bulma имеет самые читаемые CSS-классы из всех фреймворков, которые я пробовал. Кроме того, здесь есть до смешного мощная и простая система для создания сеток в стиле Metro, называемых плитками (просто посмотрите на код во второй половине скриншота и скажите мне, что вы не впечатлены!)
- **Плоская кривая обучения:** Bulma имеет высокую модульность и была создана для решения практических, повседневных проблем, с которыми сталкиваются небольшие команды и отдельные разработчики. Вы обнаружите, что Bulma очень проста в освоении, хотя я думаю, что приличный опыт в CSS всегда полезен, чтобы иметь представление о том, что может происходить под капотом. Это поможет вам, когда вы захотите отменить поведение по умолчанию.
- **Элегантность:** Взгляните на стандартную секцию Hero для Бульмы ниже. Достаточно сказано!

У Bulma небольшое, но очень увлеченное сообщество, поэтому если вы хотите избавиться от всех пустяков и при этом создавать элегантные пользовательские интерфейсы в рекордно короткие сроки, Bulma – это то, что вам нужно. Для разработчиков Bootstrap в Bulma есть отдельный раздел, чтобы убедить и помочь им перейти на него.

UIKit

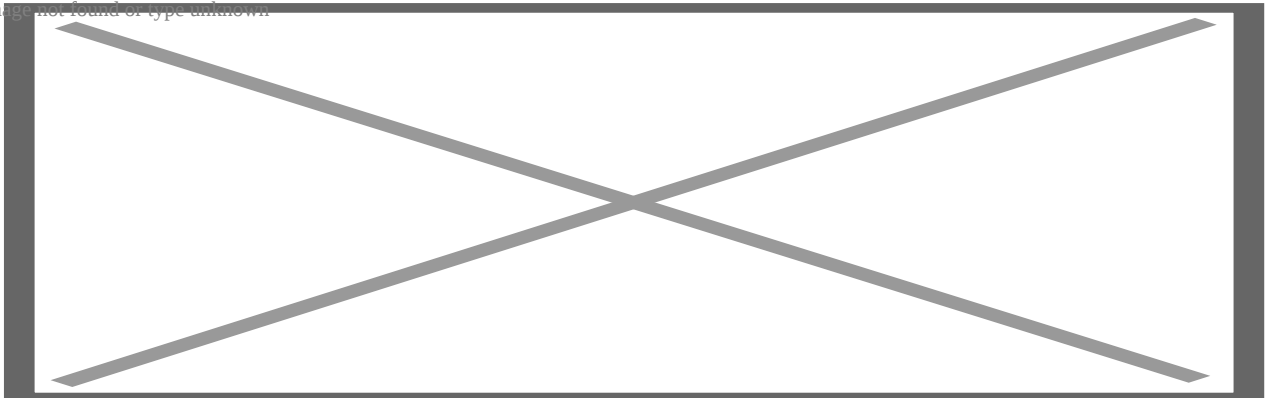
При мысли о UIKit на ум приходит минимализм. Минимализм не в функциях (на самом деле, он предлагает, пожалуй, больше всего функций среди всех фреймворков), а в дизайне. Если вам по душе суперчистый, элегантный, не стесняющий белых пятен дизайн, UIKit вам поможет.

Image not found or type unknown

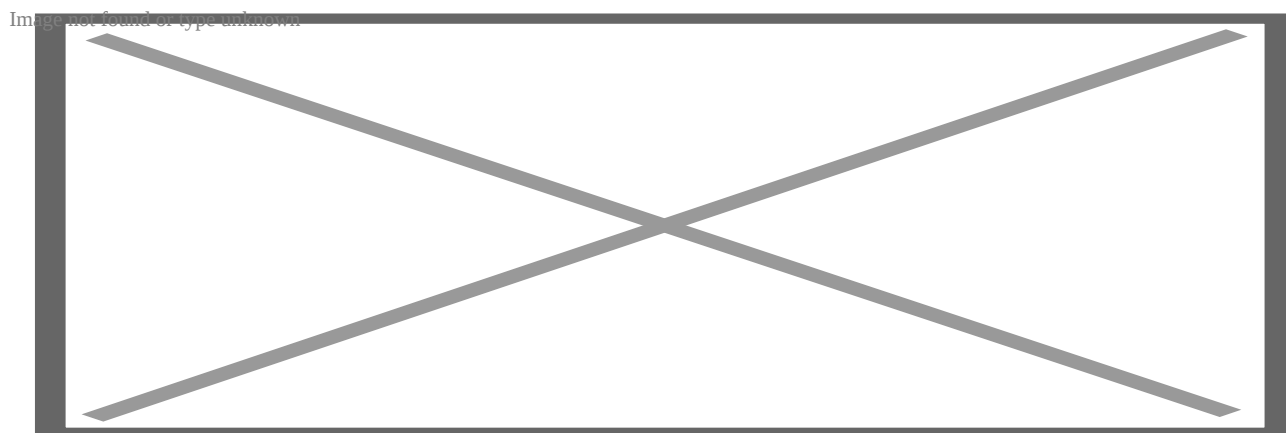


Например, посмотрите на компонент progress-bar:

Image not found or type unknown



Или компонент маркера изображения (интерактивный маркер для изображений на основе JS):



Если это не кричит об эlegantности во все горло, то я не знаю, что еще может кричать. Просто зайдите на сайт Ulkit и посмотрите все невероятные компоненты, которые он предлагает. Если только ваш руководитель проекта или заказчик не навязывает вам определенный язык стиля, я думаю, что Uikit берет корону за дизайн пользовательского интерфейса и на несколько миль опережает Material Design от Google. Но есть ли здесь какой-то подвох, спросите вы. Да, есть. Как и Bootstrap, Ulkit работает со своим JavaScript, и хотя вы можете использовать jQuery для манипуляций с DOM, использование фреймворка виртуального DOM, такого как React, невозможно. Кроме того, Uikit – это самодостаточная система, и вы не сможете изменить или расширить ее, не приложив значительных усилий.

Semantic UI

Еще один претендент на победу – Semantic UI, который пытается отличаться большим количеством тем и настроек. Существует более 3000 переменных тем, что приводит к огромной широте охвата. Так говорится в документации.

Image not found or type unknown



Bootstrap 4 как бы покрывает все это и также полностью настраивается, но одно из преимуществ Semantic UI заключается в том, что он по умолчанию приводит к красивым макетам. Тем не менее, он не является самым привлекательным из коробки, поэтому я поместил его ниже в моем списке. У него также одна из самых крутых кривых обучения, а конвенции кодирования намного строже. Попробуйте, я бы сказал, и посмотрите, не похоже ли это на то, что вы предпочитаете.

Susy

Susy – это малоизвестный на данный момент фреймворк, но это увлекательная и свежая идея. Еще один фреймворк с чистым макетом, Susy избавляет вас от всех предопределенных идей о float, grid, Flexbox, таблицах или чем-либо еще, и позволяет вам создавать макеты, которые вы хотите. “Составлять” – ключевое слово здесь, поскольку Susy предназначен для создания высокомодульных, ошеломляющих макетов с ультрасложными, необычными и точными потребностями.



В руках опытного разработчика Susy подобна огнемету, который сметает все на своем пути. Меньшие смертные, конечно, умудряются обжечь себе руки.

Чтобы получить представление о силе Susy, попробуйте использовать эту настройку по умолчанию (SASS):

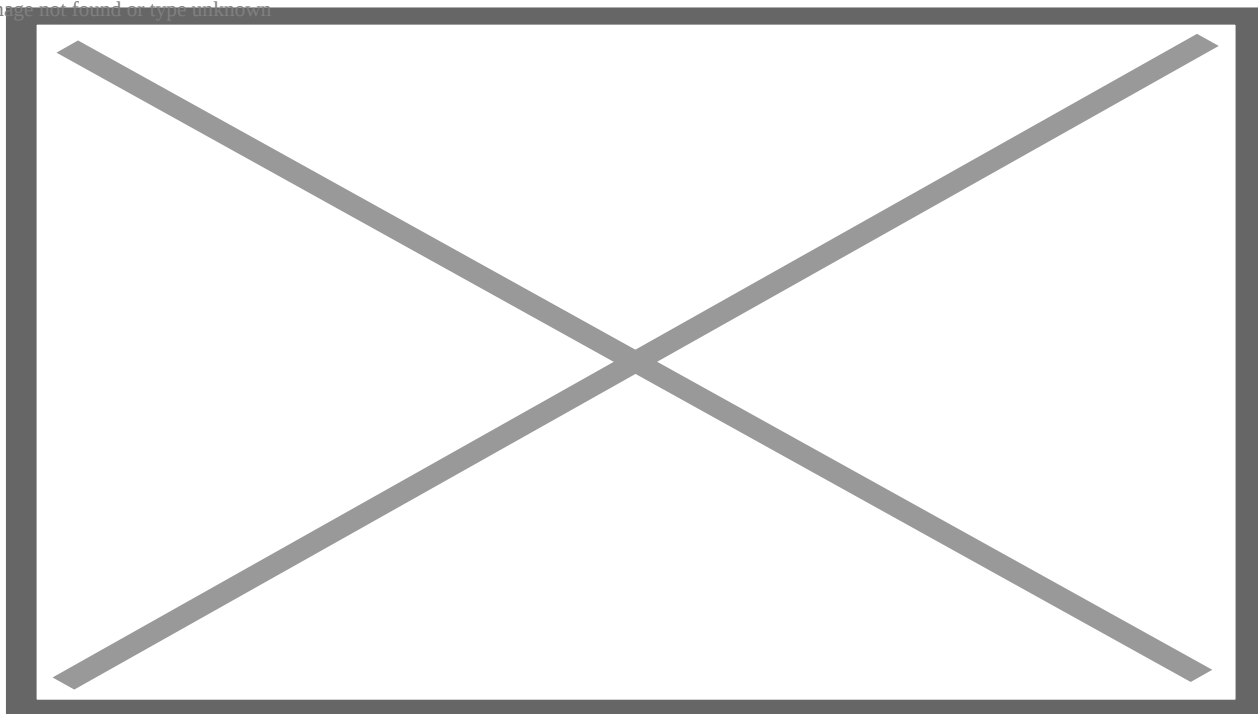
```
// 4 symmetrical, fluid columns
// gutters are 1/4 the size of a column
// elements span 1 less gutter than columns
// containers span 1 less gutter as well
$susy: (
  'columns': susy-repeat(4),
  'gutters': 0.25,
  'spread': 'narrow',
  'container-spread': 'narrow',
);
```

Я думаю, что код довольно понятен, хотя он и не для тех, кто торопится. Susy имеет смысл, если вы устали от всего раздутого, что навязывают вам современные фреймворки, и у вас есть потребности в верстке, которые, как вы знаете, не может удовлетворить ни один обычный фреймворк.

Materialize

Если вы любите Material Design от Google, Materialize – это фреймворк, который вам понравится. Самое приятное, что в нем всего несколько компонентов и классов, которые нужно изучить, и он ориентирован на то, чтобы вы как можно быстрее начали работать. Вариантов настройки немного, и Materialize следует популярному формату 12-колоночной сетки, установленному Bootstrap.

Image not found on type unknown



Однако, как по мне, дизайн Material становится настолько распространенным и настолько... плоским по умолчанию, что очень скоро мы будем жаловаться на него, как мы жалуемся на проблему Bootstrap “все сайты выглядят одинаково”. Тем не менее, это хорошая основа для начала.

Pure

Неужели Yahoo умерла? Нет, этот вопрос не отвлекает от темы, а подчеркивает

важное наблюдение: Yahoo создала фреймворк Pure и выпустила его под лицензией BSD. Беглый взгляд впечатляет меня, и я удивляюсь, почему об этом предложении не знает больше людей. В любом случае, что делает Pure, ну, чистым, так это то, что это чистый CSS фреймворк. На самом деле, разработчики пошли дальше и разбили его на различные CSS модули, которые вы можете импортировать по мере необходимости. Таким образом, если вам нужна только система сетки, нет необходимости импортировать весь CSS и увеличивать время загрузки сайта.



A set of small, responsive CSS modules
that you can use in every web project.

```
<link rel="stylesheet" href="https://unpkg.com/purecss@1.0.0/build/pure-  
min.css" integrity="sha384-  
nn4HPE8lTHyVtfcBi5yW9d20FjT8BJwUXyWZT9InLYax14RDjBj46LmSztkmNP9w"  
crossorigin="anonymous">
```

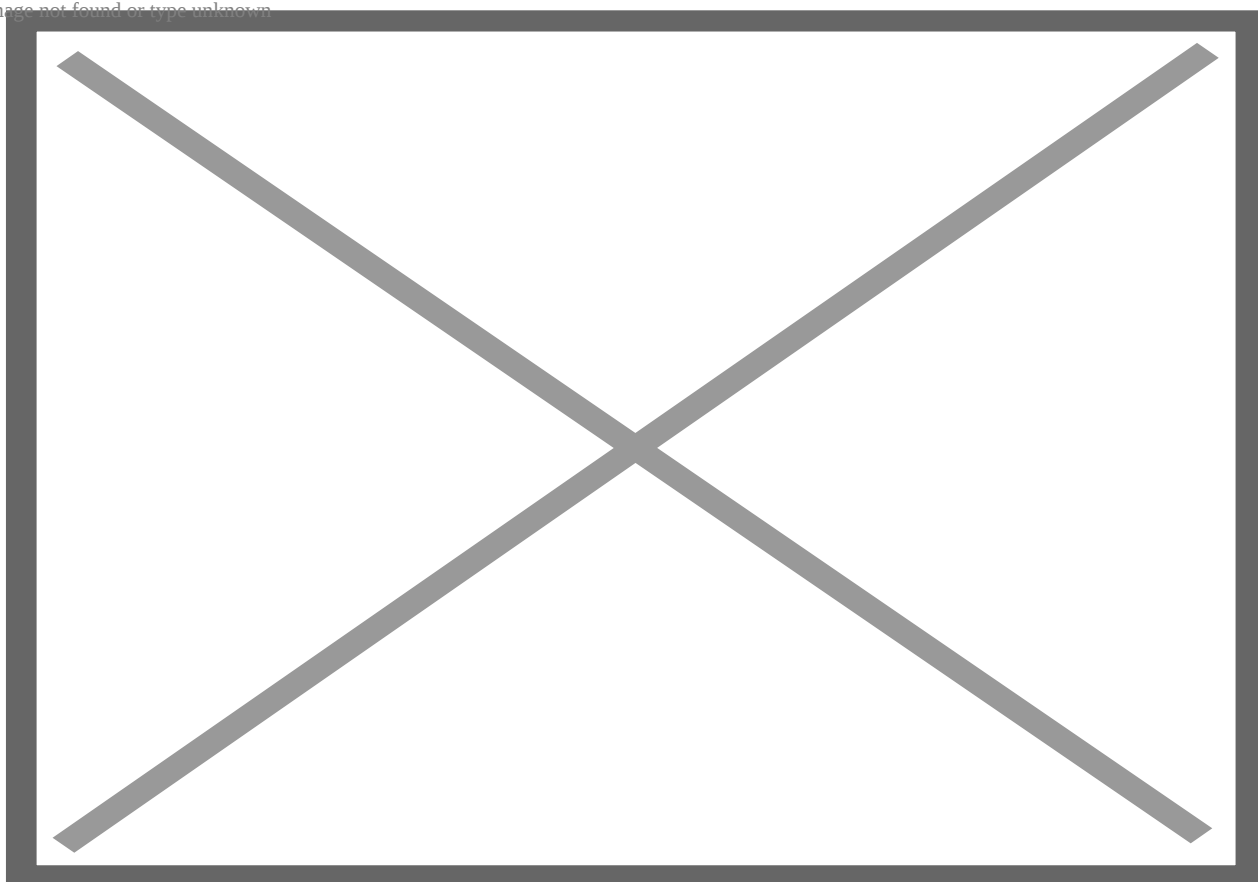
[Get Started](#)[View on GitHub](#)

Сетка Pure имеет несколько вариантов: 5-точечная, 2-точечная, 24-точечная и т.д., поэтому при создании колонок у вас гораздо больше гибкости. Pure не является самым красивым CSS-фреймворком по умолчанию, но я вижу, как он полезен для тех, кто хочет решить небольшую CSS-проблему в своем пользовательском интерфейсе и сокрушается по поводу “полезных” настроек по умолчанию, предлагаемых другими фреймворками.

Skeleton

Как видно на скриншоте, Skeleton настолько минимален, что даже не называет себя CSS-фреймворком, библиотекой или даже модулем. Это кодовый шаблон, который содержит всего 400 строк исходного кода! Невероятно? Я думаю, да, но если посмотреть на вещи в перспективе, Skeleton был разработан для крошечных или небольших проектов, которым нужно немного больше, чем макеты и позиционирование.

Image not found or type unknown



Стоит посмотреть; в конце концов, кто знает, может быть, Skeleton – это то, что вы искали все это время!

Milligram

Milligram – CSS-фреймворк, созданный для скорости и продуктивности. Разработчики сохранили его размер менее 2 КБ, что по сегодняшним меркам значит очень много. Milligram – это маленький забавный подход к CSS-фреймворкам, с

которым вам понравится работать. Расширить его легко, а с помощью нескольких строк пользовательского CSS вы сможете изменить его внешний вид так, как вам нужно.

Tailwind CSS

Tailwind пропагандировал такой стиль написания и использования CSS, который у многих вызвал крик отвращения. Посмотрите на использование классов CSS в следующем фрагменте кода (взято с сайта Tailwind):

```
<figure class="md:flex bg-gray-100 rounded-xl p-8 md:p-0">
  <img class="w-32 h-32 md:w-48 md:h-auto md:rounded-none rounded-full mx-auto" />
  <div class="pt-6 md:p-8 text-center md:text-left space-y-4">
    <blockquote>
      <p class="text-lg font-semibold">
        "Tailwind CSS is the only framework that I've seen scale
        on large teams. It's easy to customize, adapts to any design,
        and the build size is tiny."
      </p>
    </blockquote>
    <figcaption class="font-medium">
      <div class="text-cyan-600">
        Sarah Dayan
      </div>
      <div class="text-gray-500">
        Staff Engineer, Algolia
      </div>
    </figcaption>
  </div>
</figure>
```

Если вы работаете разработчиком больше пары лет, то, скорее всего, почувствуете тошноту. Так ... много ... классов ... ! И хотя здесь нет встроенного CSS, это кажется именно так, учитывая, насколько явными (и уродливыми?) являются имена классов. Итак, в течение некоторого времени Адам Вэтэн (создатель Tailwind) из уст в уста (то есть в подкастах, конференциях, сообщениях в блоге, твитах) рассказывал о том, почему он считает, что CSS, основанный на утилитах, превосходит семантический CSS.

Сегодня CSS, основанный на утилитах, добился значительных успехов. Взгляните на этот код Bootstrap 5 и посмотрите, насколько распространенными стали утилитарные классы:

```
<div class="d-grid gap-3">
  <div class="p-2 bg-light border">Grid item 1</div>
  <div class="p-2 bg-light border">Grid item 2</div>
```

```
<div class="p-2 bg-light border">Grid item 3</div>
```

```
</div>
```

Поговорив об “общей картине” Tailwind, давайте перейдем к конкретным преимуществам, которые он предлагает:

- **Повышение производительности:** О продуктивности можно спорить целый день и так ни к чему и не прийти; тем не менее, разработчики, которые перешли исключительно на Tailwind (особенно для крупных проектов), говорят, что они стали работать намного продуктивнее. Но вам придется приложить постоянные усилия, потому что Tailwind будет работать на вас только тогда, когда вы откажетесь от “лучших практик” прошлого.
- **Меньший размер пакета:** Tailwind более или менее JS манипулирует CSS, поэтому возможны некоторые элегантные вещи. Например, когда вы собираете проект, компилятор может удалить все ненужные CSS. Это довольно контрастирует с типичным веб-проектом, использующим модуль Bootstrap CSS; в этом случае весь Bootstrap CSS упаковывается в конечную вещь.
- **Конфигурируемость и настраиваемость:** Tailwind не только чрезвычайно модулен, но и построен таким образом, чтобы подчеркнуть простоту конфигурирования и настройки. Нет ничего лучше, чем “начать работу с Tailwind за 2 минуты”, потому что вам придется изучить, что, где, почему и т.д. Другой пример: если у вас нет готовой системы дизайна и вам не нравятся настройки по умолчанию, вы испытаете разочарование. Вам придется пройти через кривую первоначального обучения/настройки, прежде чем вы сможете сказать “Bay!”.
- **Отсутствие именования или переключения контекста:** Одна из самых больших обязанностей разработчиков – именование вещей. Должен ли конечный результат называться `totalAmountForUnits` или `expensesAcrossUnits`, например. Эта проблема еще более серьезна в CSS, поскольку в проекте могут быть сотни (или даже тысячи) классов. Еще одна проблема – переключение контекста: постоянное переключение между HTML и CSS, чтобы увидеть изменения в CSS. В Tailwind решения об именованиях уже приняты за вас, и поскольку вы всегда просто добавляете/удаляете классы из HTML, переключение контекста сводится к нулю.
- **Повторное использование кода:** Если вы столкнулись с проблемой “повторного использования” некоторых полезных классов путем их копирования в разных проектах, в Tailwind есть функция Components,

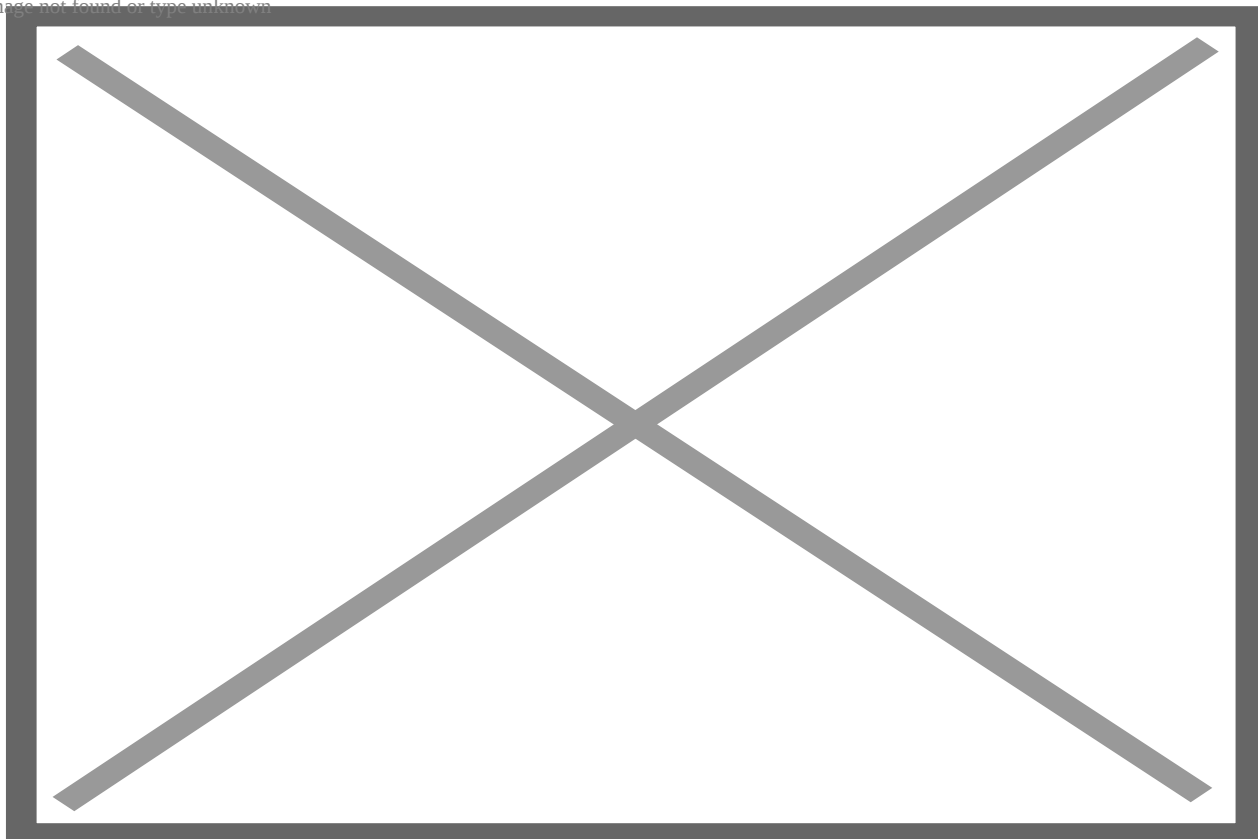
позволяющая элегантно решить эту проблему.

Для многих Tailwind стал глотком свежего воздуха, в котором они так нуждались. Если вы устали от своей текущей библиотеки CSS и хотите попробовать что-то радикальное и новое, Tailwind – это то, что вы ищете!

Tacit

Tacit – это чей-то побочный проект, на который я наткнулся, когда искал интересные вещи для этой статьи. Да, просто побочный проект; не более чем работа одного человека, основанная на его вкусе. Почему я так подчеркиваю это? Потому что это обычно означает, что риск быть заброшенным очень высок. Но есть и пара положительных моментов: 1) Проект находится на GitHub, и если вы разработчик и хотите поддержать/расширить Tacit, вы знаете, что делать. 2) CSS не похож на JavaScript в плане изменений или эволюции; какой бы стиль вы ни использовали сегодня, он будет поддерживаться браузерами еще очень долго. Итак, что же это за штука такая – Tacit?

Image not found or type unknown



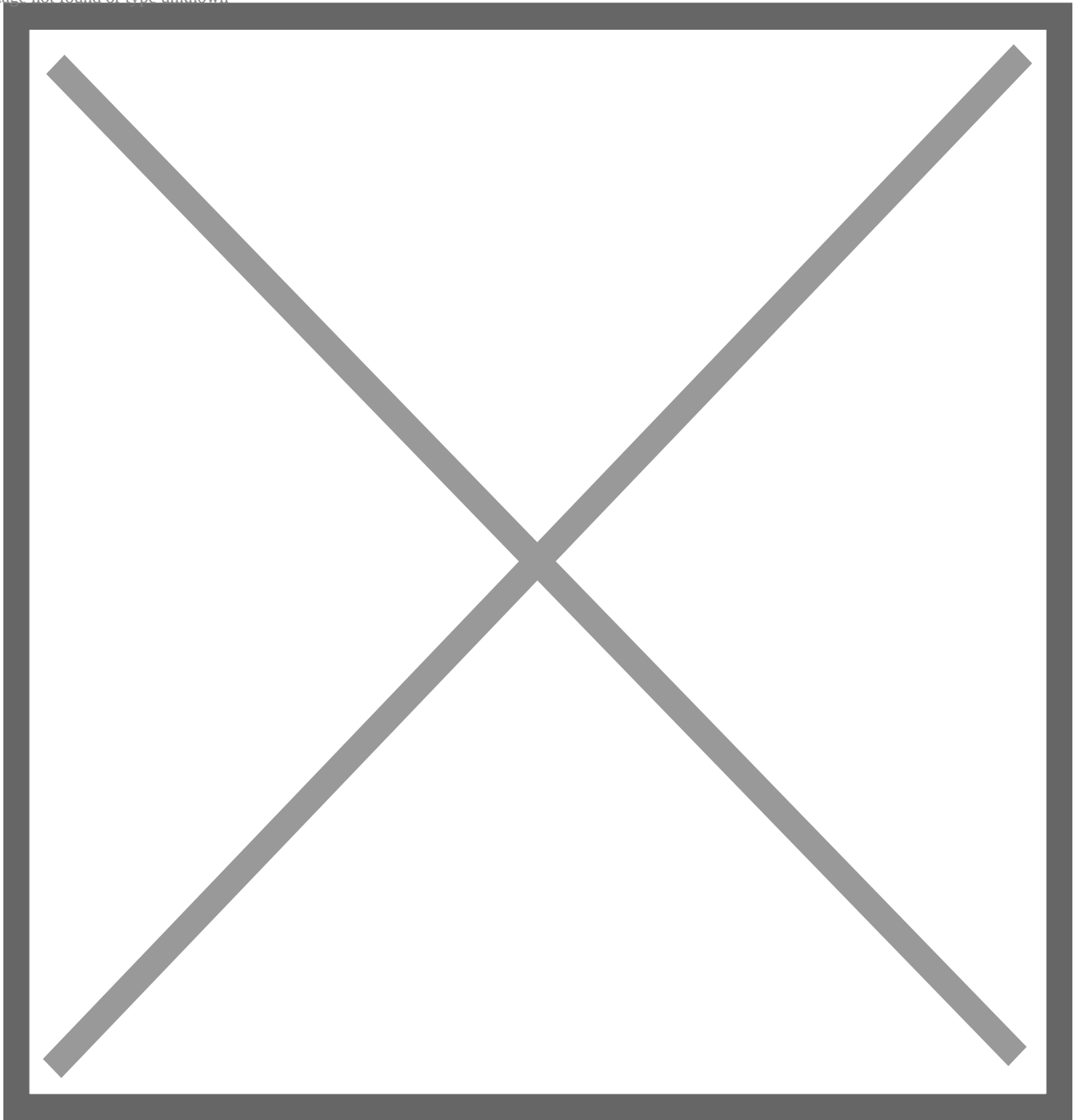
Как сказано на скриншоте, Tacit предназначен для тех, кто не знает CSS или имеет

плохой вкус в дизайне. Идея новаторская: добавьте ссылку CSS в свой проект, и вы мгновенно получите великолепно выглядящий сайт! Но убедитесь, что вы сдержите свое обещание: никогда не добавляйте другие CSS и не смешивайте материалы из других фреймворков, так как это приведет к серьезным поломкам. А хорошо ли выглядит сайт Tacit?

Если вы хотите поверить мне на слово, я бы сказал, что выбранный стиль дизайна “очень хорош”. На самом деле, я нахожу его даже более приятным, чем Semantic UI, один из самых популярных CSS-фреймворков/библиотек! Не вступая во флеймовую войну и не делая боковых сравнений скриншотов, я оставляю эту тему, сказав, что дизайн субъективен. Но вы можете сравнить и решить для себя. В любом случае, позвольте мне добавить пару скриншотов из Tacit’s look-and-feel.

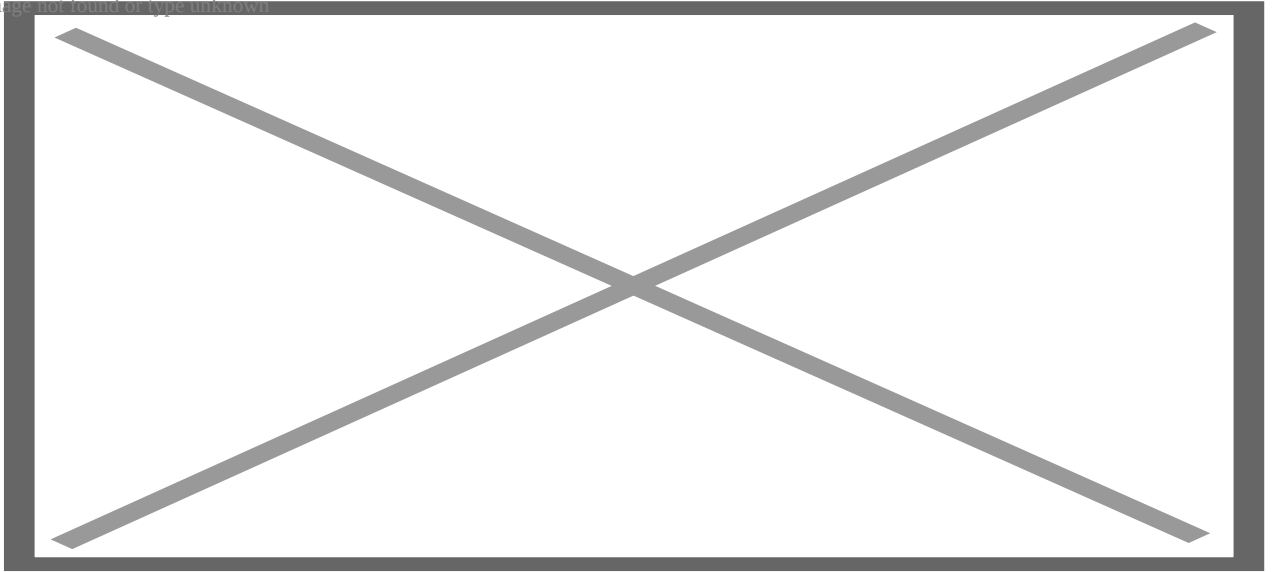
Во-первых, вот как выглядят элементы формы:

Image not found or type unknown



Вот как выглядят таблицы в их самой простой форме:

Image not found or type unknown

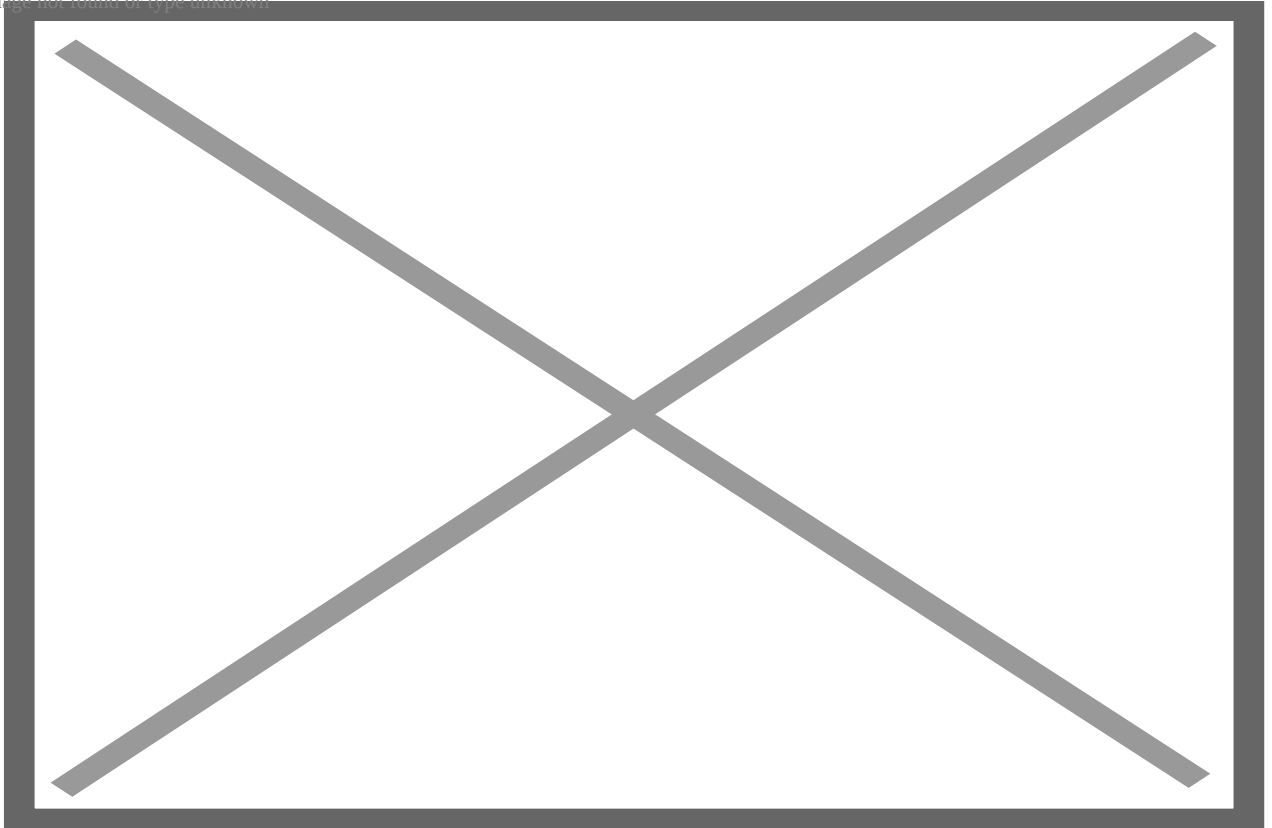


В заключение я повторю очевидное: если вы считаете, что такие цвета, шрифты, стили и т.д. отлично подходят для использования по умолчанию, Tacit станет для вас приятным облегчением.

Spectre

Современный, гибкий, легкий – если вы ищете что-то с такими характеристиками, Spectre заслуживает внимания.

Image not found or type unknown



Позвольте мне объяснить мой выбор прилагательных для Spectre:

- **Современный:** Система верстки Spectre основана на Flexbox, что является одним из лучших достижений современного CSS. Одновременно с этим, множество средств и инструментов, найденных в современных рабочих процессах CSS или других фреймворках, также включены.
- **Гибкость:** Вы можете написать самый модульный и продвинутый CSS-фреймворк, но толку от него будет мало, если он не будет легко настраиваться. Мне нравится, что о настройке упоминается в самом верху документации; следуйте за кроличьей норой, и вы найдете исчерпывающие объяснения.
- **Легкий вес:** Мы живем в эпоху, когда обеспокоенность производительностью веб-сайтов находится на подъеме; каждый передаваемый килобайт тщательно изучается и критикуется. Поэтому Spectre упаковывает все в милый пакет размером 10 КБ, который не должен быть “слишком большим” для всех нас.

Легкость Spectre не означает, что он жертвует функциональностью; здесь есть все, к чему вы привыкли как поклонник Bootstrap: компоненты (аккордеоны, хлебные крошки, карточки и многое другое), элементы (формы, таблицы, кнопки и т.д.),

компоновка (сетки, секция героя, навигационная панель и т.д.), утилиты (загрузчики, спиннеры и т.д.) и многое другое. Что я думаю о Spectre? Я более или менее продан, но есть пара вещей, которые я хотел бы упомянуть. Во-первых, мне не очень нравится основной синий цвет по умолчанию, поэтому его изменение – это первое, что я бы сделал в своих проектах. Во-вторых, в 2020 году я считаю, что CSS Grid – лучший инструмент для макетов; я нахожу Flexbox немного странным для всех макетов, поэтому я хотел бы, чтобы Spectre использовал Grid вместо Flexbox. Но это вряд ли является решающим фактором, если я могу разрабатывать быстро и получать хорошие результаты. Так что, я скажу, что смело пробуйте Spectre!

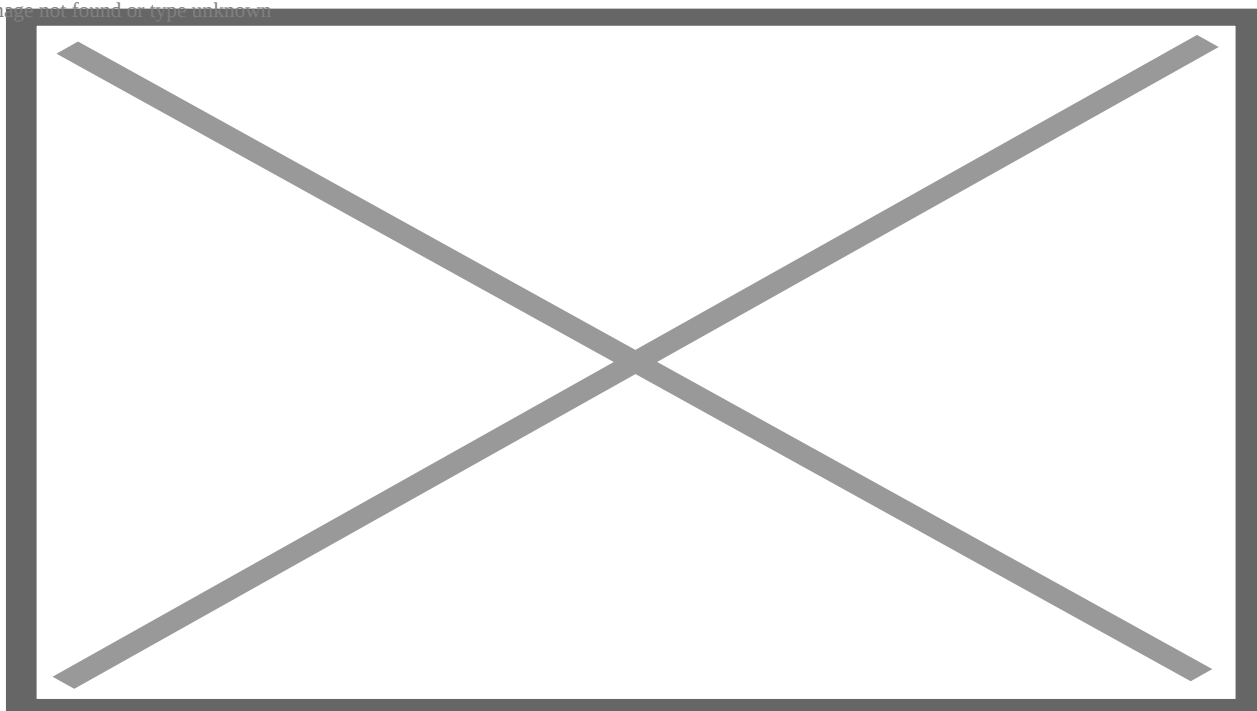
Primer

Primer – это не фреймворк или библиотека CSS. Это даже не коллекция CSS-классов или другой странный поворот в терминологии.

Тогда почему он попал в этот список? По двум причинам:

- У него есть CSS-фреймворк.
- Это одна из самых всеобъемлющих и умопомрачительных вещей, которые вы когда-либо видели.

Image not found or type unknown



Да, как сказано на скриншоте, Primer – это система проектирования, разработанная

GitHub для собственного использования. В конце концов, они поняли, что эта штука мощная и чрезвычайно полезная для фронтенда и визуальной работы; поэтому они сделали ее с открытым исходным кодом. Я долго пытался найти правильные слова, чтобы описать Primer. Их собственная команда Design System Team называет его ... ну, да, системой проектирования, но я думаю, что сфера применения гораздо шире, чем то, что мы подразумеваем, говоря “система проектирования”.

Primer включает в себя множество вещей, важность которых может быть не сразу очевидна. По крайней мере, для backend/full-stack разработчиков. Поэтому давайте рассмотрим, что это за особые вещи, почему они особые:

- **Руководство по интерфейсу:** В команде, состоящей из одного человека, разработка и поддержка пользовательского интерфейса является тривиальной задачей. Но добавьте в команду еще трех человек – и при этом опытных, успешных людей – и вы увидите, как все начинает ухудшаться. Причина – отсутствие четко определенных и строго соблюдаемых правил, чтобы каждый знал, что делать в любой ситуации. Primer поставляется с хорошим набором руководящих принципов, которые вы можете изменить или адаптировать под свои нужды.
- **Компоненты Figma:** Несмотря на то, что я разработчик, я достаточно разбираюсь в дизайне, чтобы понять, что отсутствие четкой привязки/использования компонентов может выйти из-под контроля по мере продвижения проекта. Например, как должна выглядеть кнопка при нажатии в темном режиме? Если никто не задокументировал это, а на проект оказывается давление, то у вас будет несколько вариантов. Поэтому сотрудники Primer предоставили свои собственные компоненты Figma, чтобы мы могли учиться и делать лучше.
- **CSS Framework:** Наконец, то, благодаря чему Primer занял место в этом списке! Primer имеет модульную, BEM-стиль CSS-фреймворк, который обеспечивает отдельные стили для страниц продукта и маркетинга. Это то, что GitHub сделал для своего удобства, и вы вольны не трогать это или принять общую философию такой, какая она есть. Даже если вас интересует только CSS, я рекомендую вам заглянуть сюда.
- **Иконки (Octicons):** У каждой целостной системы дизайна есть свой взгляд на иконки. Для Primer ответ – Octicons. На самом деле, иконок не так уж и много. Тем не менее, я ставлю их выше в списке, потому что иконки являются частью абсолютных основ: если все совпадает, но несколько иконок не совпадают, результатом будет пользовательский интерфейс, который вызовет у вас

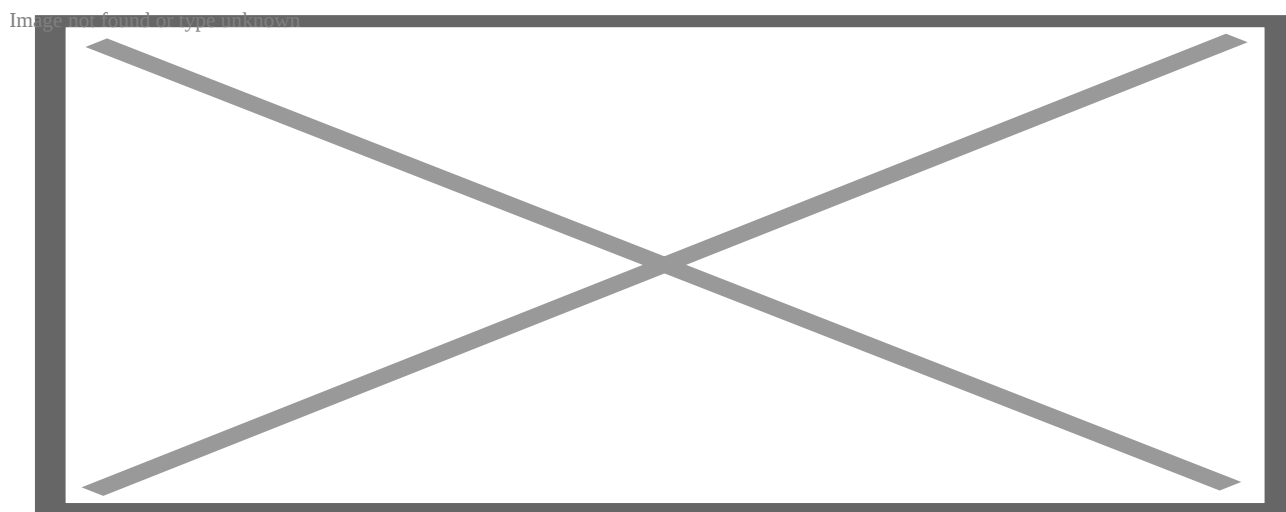
КОНЪЮНКТИВИТ.

- **Компоненты React:** И, конечно, поскольку React – самая популярная библиотека для фронтенда, ребята с GitHub закодировали CSS-элементы как компоненты React, которые теперь доступны всем нам!
- **Презентации:** Эта часть Primer посвящена тому, какой язык дизайна использовать при презентации аудитории (как это делают сотрудники GitHub внутри компании). Шрифты, цвета, границы, контрасты – все решения уже приняты. Вы можете свободно изучать их рекомендации и разрабатывать свои.

Наибольшую пользу от Primer можно получить, приняв его целиком. Это возможно не для всех, и некоторые могут даже обнаружить, что их язык дизайна конфликтует с языком GitHub. И потом, некоторые люди находят одну конкретную часть (или даже часть части!) Primer полезной и бегут с ней. Независимо от того, где вы стоите на этом спектре, Primer есть что вам предложить: либо многократно используемые, высококачественные компоненты системы проектирования, либо пищу для размышлений для ваших собственных процессов и рекомендаций!

Fictoan

Fictoan – это интуитивно понятный фреймворк для веб-дизайнеров, которые хотят создавать пользовательские интерфейсы. Фреймворк был создан, чтобы размыть границы между разработчиками и дизайнерами.



Создатели считают, что дизайнеры должны полностью контролировать пользовательский опыт и пользовательский интерфейс. С другой стороны, разработчики должны сосредоточиться на производительности и таких вещах, как

развертывание.

Зачем использовать Fictoan?

- **Разнообразие компонентов:** Художественная литература состоит из сотен компонентов, которые классифицируются по разным категориям. Например, если вы ищете форму, существует более десяти дизайнов.
- **Открытый исходный код:** Fiction доступен для бесплатного использования, и вы можете найти код на GitHub.
- **Доступность:** Платформа разработана с особым акцентом на доступность. Дизайнеры также могут настроить доступный код для улучшения доступности.
- **Легко начать:** Fiction был создан для проектов React.

Вы можете создать простое приложение React, чтобы продемонстрировать, как оно работает. Используйте эту команду:

```
npx create-react-app fictoan-app
```

```
cd fictoan-app
```

Теперь вы можете добавить fictoan в свое приложение React следующим образом:

```
npm install fictoan-react
```

Мы можем открыть проект в редакторе кода и создать простую кнопку.

```
<Button kind="primary">
```

```
  Button
```

```
</Button>
```

- **Легкая настройка:** Вы можете настроить приведенный выше код под свои нужды.

Недостатки Fictoan

- **Все еще новый:** 21 июня 2020 года первый релиз был выложен на GitHub. У платформы менее 10 форков, и ей еще предстоит пройти долгий путь, чтобы привлечь массы.
- **Ограниченные возможности:** Несмотря на то, что Fictoan имеет несколько удивительных современных дизайнов, ему еще предстоит пройти долгий путь,

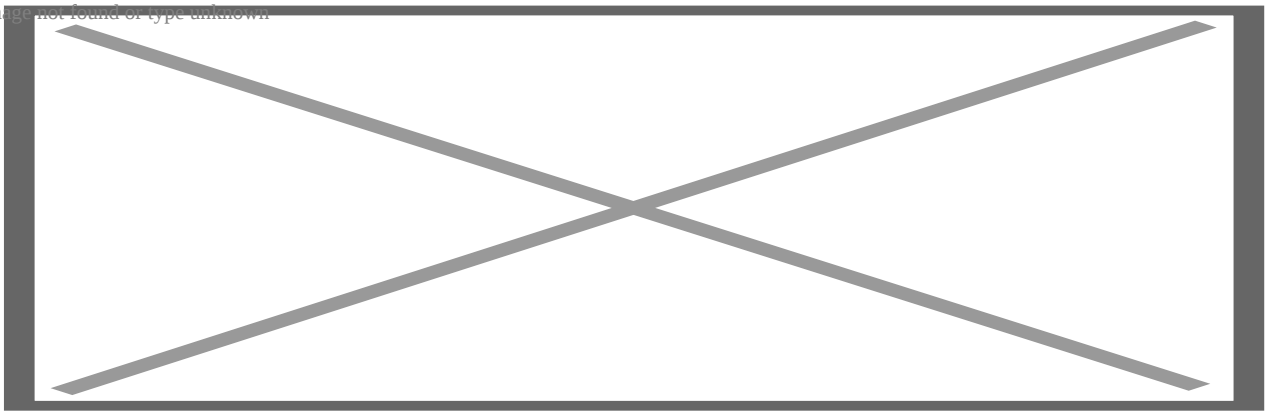
чтобы конкурировать с другими, такими как Bootstrap, в плане возможностей.

- **Зависимость от зависимостей:** Для своей работы Fictoan зависит от некоторых библиотек, таких как styled-components. Наличие большого количества зависимостей может привести к большому размеру приложения.

Vanilla

Vanilla – это простой, расширяемый CSS-фреймворк, написанный на Syntactically Awesome Style Sheets (Sass), надстрочном элементе CSS. Vanilla используется в большинстве продуктов Canonical, таких как Ubuntu и JAAS.

Image not found or type unknown



Чтобы начать использовать Vanilla, создайте файл проекта и перейдите в него. Затем вы можете установить Vanilla с помощью yarn или npm.

```
yarn add vanilla-framework
```

или

```
npm install vanilla-framework
```

Также вы можете добавить ссылку на CDN в раздел **<head>** вашего html-файла, чтобы начать использовать Vanilla Framework.

```
<link rel="stylesheet" href="https://assets.ubuntu.com/v1/vanilla-framework-ve
```

Почему стоит использовать Vanilla?

- **Она отзывчива:** Приложения, созданные с использованием Vanilla Framework, реагируют на различные размеры экранов, например, смартфонов и персональных компьютеров.
- **Написан на языке Sass:** Sass позволяет легко определять переменные и

миксины в коде CSS.

- **Поддержка нескольких браузеров:** Вы можете использовать приложения Vanilla в различных браузерах, таких как Firefox, Chrome и Safari.
- **Компонуемость:** Вы можете включить определенный код, который вам нужен, или установить весь фреймворк в ваш проект.
- **Открытый исходный код:** Исходный код является открытым, он создан и поддерживается веб-командой Canonical.

Недостатки использования Vanilla

- **Ограниченный уровень кастомизации:** Vanilla не может предложить тот уровень кастомизации, который может понадобиться некоторым разработчикам.
- **Ограниченная документация:** По сравнению с другими CSS-фреймворками, такими как Foundation и Bootstrap, документация по Vanilla не такая подробная.
- **Отсутствие активного сообщества:** Vanilla не имеет большого сообщества по сравнению с другими фреймворками.

Итак, какой CSS-фреймворк является лучшим?

Признайтесь, вы уже задавали подобные вопросы и получали неутешительный ответ: ни один. Выбор фреймворка (или инструмента, или даже человека в вашей жизни, если на то пошло) зависит от множества факторов. Если вы хотите услышать мой совет, то вот он: Отсекайте шум. Если люди сходят с ума по чему-то новому и блестящему, это не значит, что вы должны изучать это, иначе вы останетесь позади. Пробовать новое – это здорово, но бегать по кругу в поисках идеального инструмента – пустая трата времени.

Дата Создания

03.05.2023