

РАЗРАБОТКА ВЕБ-САЙТОВ\*, HTML\*, CSS\*

## Что поправить в верстке перед выпуском в продакшн?

ИЗ ПЕСОЧНИЦЫ

anton\_gcor 16 января 2017 в 16:14 👁 57,5k

## Что поправить в верстке перед выпуском в продакшн?

Это статья — чек-лист того, что нужно не забыть поправить в любом проекте. Список полезных мелочей. Весь текст поделен на две части. Первая — про простые элементы страницы, такие как текст, кнопки, изображения, формы и другие. Вторая часть про производительность, масштабируемость, безопасность и доступность.

### Часть 1

#### Текст

#### Размер шрифта и интерлиньяж

Оптимальная высота строки (`line-height`) 1.4 — 1.8 от размера шрифта. Это значение лучше указывать множителями, а не конкретными значениями в пикселях. Так оно будет масштабироваться относительно любого размера шрифта.

## Длина строки

Ограничение длины строки помогает легче перепрыгивать глазу со строки на строку. Текст удобно читать, если в него уменьшается минимально 6 слов, а максимально 12. Достаточно использовать `min-width` и `max-width`.

Длина строки влияет на интерлиньяж. На телефонах `line-height` должен быть меньше, чем на большом мониторе. Существует интересный адаптивный подход ([CSS-шлюзы](#)), который плавно подстраивает размер шрифта и интерлиньяж под экран.

## Ограничение длинных слов

Текст в строке может оказаться непредвиденно большим, в этом случае его можно ограничить с помощью [text-overflow](#).

Эдин|

Эдинбург

Эдинбург семи мо...

## Выравнивание по ширине

Многие знают, как плохо выглядит выравнивание по ширине (`text-align: justify`). Текст становится трудно читать из-за больших отступов между словами. В книгах и газетах нет таких больших пробелов из-за переносов в словах. Чтобы включить такой перенос нужно использовать свойство `hyphens`, оно автоматически проставляет мягкие переносы в словах. К сожалению, это работает далеко не везде, но можно использовать [плагин](#) и [словарь к нему](#).

### Апериодический сушильный шкаф: методология и особенности

Вместе с тем, понятие тоталитаризма астати́чно. Маховик стабилизирует англо-американский тип политической культуры. Политическое лидерство, короче говоря, активно.

### Апериодический сушильный шкаф: методология и особенности

Вместе с тем, понятие тоталитаризма астати́чно. Маховик стабилизирует англо-американский тип политической культуры. Политическое лидерство, короче говоря, активно.

# Типограф

Из-за того что на клавиатуре нет нужных клавиш, в вебе используются знаки, которые не соответствуют правилам русского языка.

- Это правильные кавычки «», а это компьютерные ""
- Это тире —, а это дефис -
- В типографике также принято не разрывать предлог и слово к которому оно относится, даже если предлог оказывается в конце строки. Чтобы сделать правильный перенос, достаточно пробел между словом и предлогом заменить на неразрывный пробел `&nbsp;`;

Все эти проблемы можно решить с помощью [онлайн типографов](#), плагинов и предустановленных [расширений](#) на компьютер.

## Продвинутые методы работы с типографикой

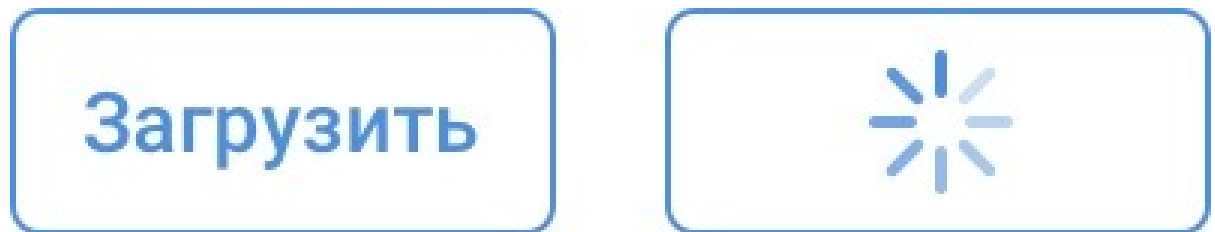
- [Интервалы](#)
- [Загрузка](#)
- [Сглаживание](#)

## Обратная связь (наведение, клик, скролл)

## Блоки

Удобный интерфейс откликается на каждое действие, будь то наведение мыши или клик. Нестандартному кликабельному элементу в некоторых случаях можно добавить:

- Курсор руки вместо стрелки `cursor:pointer` (если не получится сделать ссылкой)
- Добавить хOVER. Хотя бы просто `opacity: 0.8`.
- Добавить атрибут `role` со значениями `button` или `link`
- Убрать случайное выделение текста (`user-select: none`).  
Подходит для дропдаунов и других элементов на которые можно кликнуть несколько раз подряд.
- Добавить обработчик нажатия на `enter`
- Предусмотреть фокус
- Предусмотреть спиннер, если состояние элемента зависит от асинхронного события



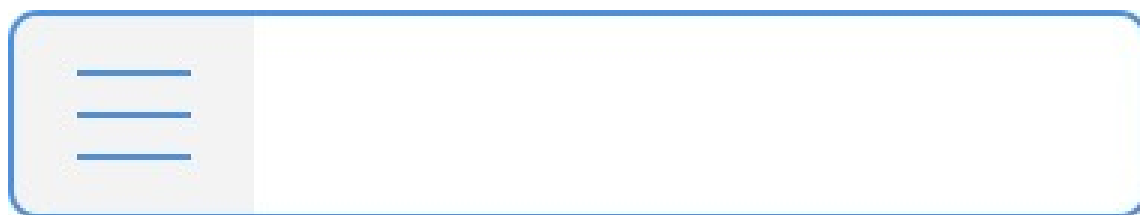
## Область клика

Чем больше и ближе объект, тем легче на него кликнуть.

Существует даже математическое [правило](#), которое это описывает. Например, в поисковой выдаче Google ссылкой является не только номер страницы, но и часть логотипа. Это спасает от случайных промахов.



Чтобы увеличить область клика, нужно использовать `padding` вместо `margin`. Серым показан отступ паддингом.



Не накладывайте поверх кликабельных элементов другие элементы, например, иконку лупы над инпутом. В крайнем случае добавьте иконке `pointer-events` или решите это более поддерживаемым способом (в случае с лупой, добавьте её через `background-image`).

## Правильный формат ссылки

Если ссылка является почтовым адресом, укажите в ссылке `mailto:..`. Тогда по клику на неё сразу откроется почтовый клиент. Тоже самое можно сделать для телефона префиксом `tel:..`

## Скролл на айфоне

По умолчанию, горизонтальный скролл в айфоне обрывистый и неуклюжий. Включить привычный скролл с инерцией можно

свойством `-webkit-overflow-scrolling: touch`.

## Инпуты

### Тег form

Чтобы форма работала в любых браузерах, даже без js, нужно использовать тег `form` по назначению. Форму можно улучшать очень долго, вплоть до [отправки данных даже в офлайне](#).

### Типы

Разные значения атрибута `type` (`email`, `number`, `password` и другие) включают правильную клавиатуру на телефонах и помогают валидировать данные

### Автофокус

Если на странице основной элемент поиск, ему можно добавить автофокус с помощью атрибута `autofocus`. В него сразу будет можно вводить текст, предварительно не фокусируясь.

## Изображение

### `picture` и `srcset`

В последнее время появляются новые возможности для [работы с изображениями](#). Они позволяют избежать лишних загрузок для отзывчивых изображений. На телефон больше не будет грузиться

тяжелая графика для компьютера. К сожалению, есть проблемы с поддержкой, но про это все равно стоит знать.

## Таблица

### Адаптивность

Таблица изначально не предусмотрена под ширину экрана телефона. Есть несколько вариантов, как изменять таблицу под небольшие экраны

- Уменьшить размер шрифта
- Обернуть таблицу в блок и сделать ему горизонтальный скролл
- Перевернуть. Сделать таблицу вместо короткой и широкой узкой и длинной
- Заменить таблицу на несколько списков

Сделать таблицу удобной не просто. Советую почитать статьи про отображение [больших таблиц](#) и [переверстку](#).

## Компоненты

### Теория близости

Объекты, расположенные близко друг к другу, воспринимаются связано. Если поставить заголовок посередине нескольких статей, будет сложно разобраться, к какому из текстов относится заголовок. Кстати, на Хабре именно такая ошибка. Заголовок параграфа должен быть ближе к своему параграфу, а не быть



равноудален от всех. Существует масса других примеров с [ошибкой в теории близости](#).

- [EN](#) Двадцать лучших CSS анимаций на CodeCanyon
- [EN](#) Добавление Box Shadow (z-глубины) к компонентам Angular Material 2

## JavaScript

- [H](#) NativeScript, что за зверь и для чего он нужен?
- [EN](#) JavaScript везде [инфографика]

## Пустой блок

Бывает, что баннерная реклама не приходит в баннерную позицию, тогда остается большой непонятный пробел. Чтобы его убрать, достаточно использовать псевдокласс `:empty`:

```
.banner:empty {  
    display: none;  
}
```

## Почти, но не совсем

Верстка (в плане дизайна) приятного сайта не разваливается на ходу. Элементы ровно выровнены, размеры хорошо подобраны. Посмотрите статью [«Почти, но не совсем»](#), она поможет взглянуть на любой сайт свежим взглядом.

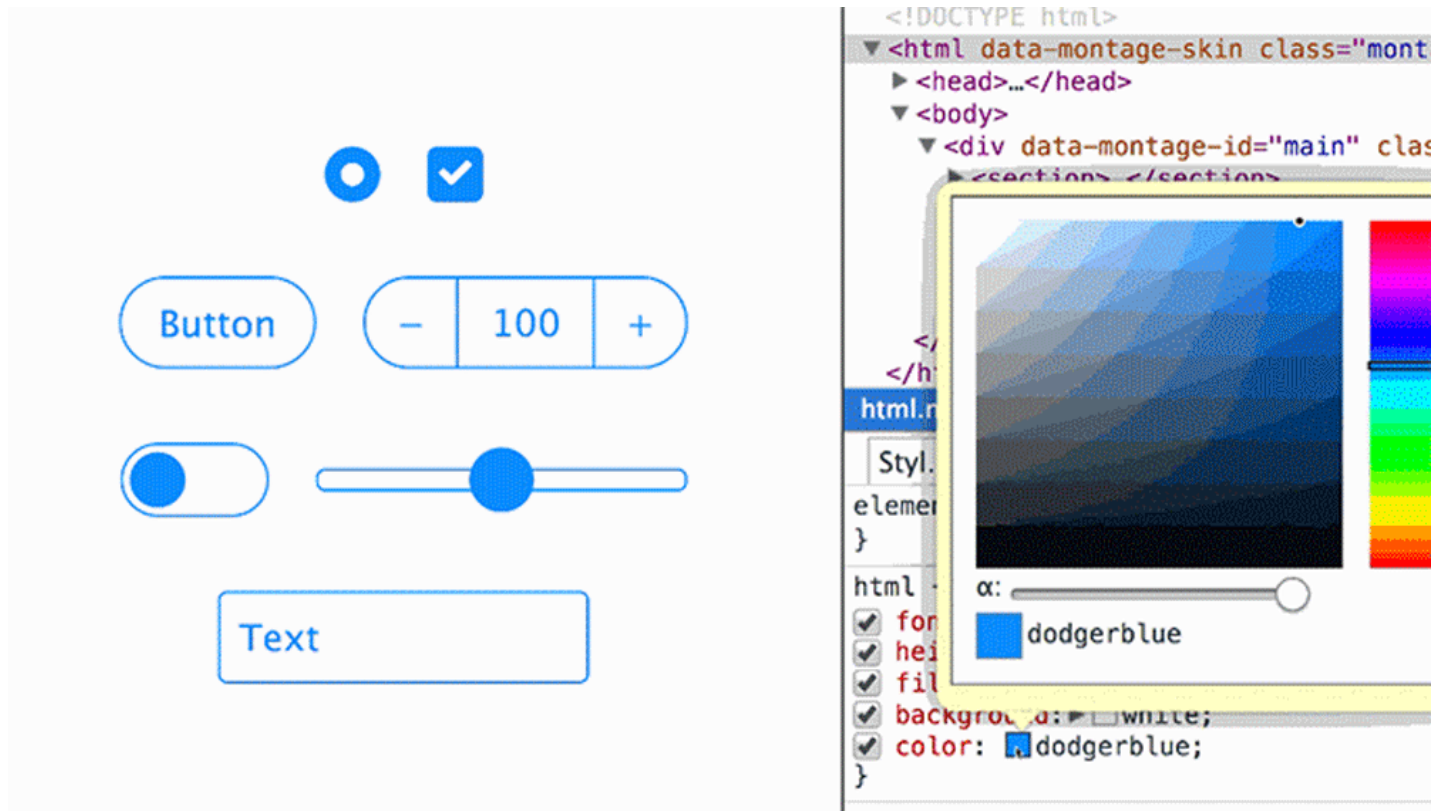
## Часть 2

## Масштабируемость

# Поддерживаемость

Несколько простых практик:

- Использовать меньше фиксированных размеров. Это даст возможность легко встроить одни компоненты в другие
- Использовать `inherit` и `currentColor`. Они поддерживаются в любом браузере и позволяют [наследовать стили](#)



- Не использовать [краткого написания свойств](#). Например, `background: red` изменяет не один, а сразу 10 стилей, включая `background-color`, `background-size`, `background-repeat` и другие
- Не использовать много сторонних библиотек и хитрых возможностей препроцессоров. Они мешают другим разработчикам быстро вникнуть в суть кода

## Архитектура

Важное условия для масштабирования кода — единообразие. Посторонний разработчик должен как можно быстрее вникнуть в код и продолжить его писать в том же стиле. Существуют множество методологий наименований селекторов, в том числе [БЭМ](#).

Хорошим примером единообразия является бутстрап. Даже незнакомый с проектом разработчик легко продолжит писать код. Ему не нужно будет вникать, он скорее всего уже все знает. Важно с самого начала определиться со стайл гайдом проекта и писать в одном стиле.

## Типовая страница

Если проект довольно крупный, попросите дизайнера собрать все основные компоненты в отдельный типовой лист. Страницы можно будет собирать из готовых элементов как конструктор. Такие типовые листы могут перерасти в целые гайды и библиотеки компонентов. Это отличный подход для крупных компаний.

Примеры:

- [Библиотека компонентов Яндекса](#)
- [Интерфейсные гайды Контура](#)

## Производительность

## Селекторы

Браузер обрабатывает свойства справа налево. Селектор `.section a div` вначале найдет все элементы `div` на странице, затем все ссылки `a` и только потом определит, какие именно `a` входят в элемент `.section`. Гораздо оптимальнее селектор по классу, например, `.section__link`. Он проще, менее специфичен и немного быстрее работает. Не стоит переживать, что имена классов могут получиться очень большими, это **ни на что не влияет**.

Вторая проблема селекторов — **неиспользуемые селекторы**. Селектор для поиска неиспользованного элемента обходит все дерево и ничего не находит. Время тратится впустую.

## Анимации

Тяжелые анимации сильно тормозят страницу. Самые лучшие свойства для анимирования — `transform` и `opacity`. Они меньше других влияют на производительность.

Существует способ оптимизировать сложные вычисления за счет переноса части сложных вычислений с центрального процессора на графический. Такое **аппаратное ускорение** можно включить свойством `transform: translate3d(0, 0, 0)`. Оно создаст отдельный слой компоновки с которым будет работать графический процессор. Существует еще один похожий подход — заранее предупредить браузер о будущих изменениях элемента с помощью **`will-change`**.

Не стоит злоупотреблять `translate3d` и свойством `will-change`, большинство оптимизации браузер сделает за вас сам. Почитайте также подробный разбор [анимаций на GPU](#).

## Тяжелые свойства

Не стоит использовать большие радиусы, лишний ресайзинг картинок (`background-size`) и сложные тени. Браузеру придется делать огромное количество перерасчетов. Например, Airbnb [увеличила производительность скролла](#) просто убрав размытие теней.

## Скорость загрузки

### Сжать стили

Конкатенируйте, минифицируйте и кешируйте файлы стилей.

### Критичные стили

[Критичный CSS](#) это минимальный набор блокирующего CSS, требуемого для рендеринга верхней части страницы. Оставшаяся часть CSS будет загружена асинхронно позже. Однако, у такой практики есть недостатки:

- Критичные стили не получится кэшировать в браузере
- Не получится кешировать HTML файлы, если страницы сайта генерируются динамически

## Префетчинг

Браузер можно предупредить об использовании на странице ресурсов с другого адреса. Тогда браузер заранее преобразует URL этих ресурсов в IP и не будет тратить время на поиск DNS.

```
<link rel="dns-prefetch" href="//example.com">
```

Кроме этого, браузер можно предупредить о файлах, которые обязательно понадобятся в будущем. Браузер к этому времени их скачает и сохранит в кэш.

```
<link rel="prefetch" href="image.png">
```

Существует еще [несколько других техник префетчига](#), однако эти возможности еще не очень хорошо поддерживаются браузерами и имеют множество ограничений.

## Безопасность

### Target

Страница, открытая через `target="_blank"`, получает частичный контроль над открывшей её страницей через js свойство `window.opener`. Эту уловку можно использовать, чтобы перевести пользователя на фишинговую страницу. Чтобы избежать

уязвимости, добавляйте к ссылкам с `target="_blank"` атрибут `rel="noopener noreferrer"`.

## Семантика

### Верстка

Нативные элементы HTML5 содержат правильные семантические атрибуты. Важно верстать таблицу с помощью `<table>`, формы с помощью `form`, кнопки с помощью `<button>` и т.д. Не забудьте поставить атрибут `alt` в каждой картинке.

### ARIA и роли

Роли и атрибуты ARIA помогают пользоваться сайтом со скринридера и других нестандартных технологий. На доступном сайте не должно быть недоступных с клавиатуры элементов управления. Каждый такой элемент должен иметь свое описание. Это описание может быть невидимым и указано через атрибут `aria-label`.

Попробуйте походить по интерфейсу с помощью кнопки таб, зайдите на сайт с электронной книги или кнопочного телефона (если есть). Проверьте, что все элементы управления фокусируются. Не отключайте просто так свойство `outline`. Например, Яндекс красит его в свой фирменный цвет.

[Карты](#)[Маркет](#)[Новости](#)[Переводчик](#)[Картинки](#)

Найдётся всё. Например, 100 долларов в рублях

Порядок фокусировки можно подредктировать атрибутом `tabindex`. Совсем отменить семантику элемента можно с помощью `role="presentation"` (например, чтобы используемая для вёрстки таблица не воспринималась как таблица с данными). Большое количество других рекомендаций можно почитать в этой [статье про доступность](#).

## Печать

Попробуйте [распечатать](#) ваш интерфейс.

## Микроразметка

Существуют различные [словари и синтаксисы](#) микроразметки. Они помогают различным роботам легче работать с вашей страницей. Сущности Schema.org могут быть полезны для поисковиков, а с помощью Open Graph можно настроить красивое отображение ссылки на ваш сайт в публикациях в соцсетях.

Проголосовать:



+82





Поделиться:



Сохранить:



Комментарии (47)

## Похожие публикации

### Верстка рассылок — что мы имеем?

dudeonthehorse • 6 ноября 2012 в 11:24

36

### Sciter — встраиваемый HTML/CSS/scripting engine

csmile • 14 октября 2012 в 08:39

30

### HTML/CSS-инъекция в почте Mail.Ru

shadowjack • 15 октября 2011 в 21:41

63

## Популярное за сутки

### Яндекс открывает Алису для всех разработчиков. Платформа Яндекс.Диалоги (бета)

BarakAdama • вчера в 10:52

69

## Почему следует игнорировать истории основателей успешных стартапов

ПЕРЕВОД

m1rko • вчера в 10:44

20

## Как получить телефон (почти) любой красоты в Москве, или интересная особенность MT\_FREE

ИЗ ПЕСОЧНИЦЫ

cab404 • вчера в 20:27

24

## Java и Project Reactor

zealot\_and\_frenzy • вчера в 10:56

10

## Пользовательские агрегатные и оконные функции в PostgreSQL и Oracle

erogov • вчера в 12:46

6

## Лучшее на Geektimes

### Как фермеры Дикого Запада организовали телефонную сеть на колючей проволоке

NAGru • вчера в 10:10

31

### Энтузиаст сделал новую материнскую плату для ThinkPad X200s

alizar • вчера в 15:32

49

## Кто-то посылает секс-игрушки с Amazon незнакомцам. Amazon не знает, как их остановить

85

Pochtoycom • вчера в 13:06

## Илон Маск продолжает убеждать в необходимости создания колонии людей на Марсе

139

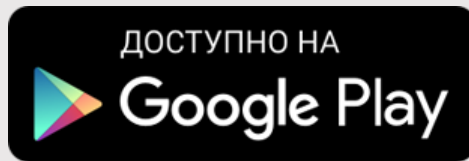
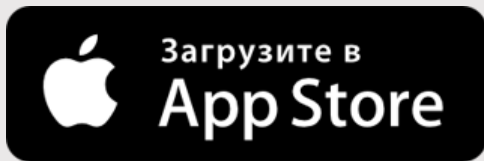
marks • вчера в 14:19

## Дела шпионские (часть 1)

16

TashaFridrih • вчера в 13:16

Мобильное приложение



Полная версия

2006 – 2018 © TM