Чек-лист по тестированию HTML-верстки: что в себя включает проверка; проверяемые параметры.

Проверяемые параметры можно разделить на несколько групп:

1. Соответствие макету

2. Работа в разных окружениях

3. Проверка на разных разрешениях экрана (проверка десктопной и адаптивных версий)

4. HTML

5. Javascript

6. CSS

7. Фавиконки

8. Шрифты

9. Навигация

10. Заголовки

11. Контент

12. Изображения

13. Ссылки и кнопки

14. Футер

15. Формы

**1. Соответствие макету**

* Расхождение макета и верстки в пикселях.
* Корректный вывод элементов интерфейса в векторном формате.
* Поддержка retina-мониторов.
* Наличие элементов, выбивающиеся из цветовой гаммы сайта.
* Наличие элементов с малой контрастностью.

**2. Работа в разных окружениях**

* Кроссбраузерность.
* Корректная работа на разных устройствах.
* Корректная работа на разных операционных системах.
* Корректная работа сайта при различных настройках местоположения пользователя, часового пояса и времени.
* Корректная работа с разной скоростью интернета.
* Корректная работа при включенном расширением AdBlock в браузере.
* Наличие мета-тега <meta http-equiv="X-UA-Compatible" content="IE=edge">.
* Корректное отображение кнопок, полей ввода, выпадающих списков и радиокнопок на разных устройствах.
* Корректное отображение сайта с административной панелью CMS. (Только для сайтов созданных на конструкторе WordPress)
* Корректная настройка встраиваемых карт для тач-устройств.
* Корректное фиксирование хедера при прокрутке.

**3. Проверка на разных разрешениях экрана (проверка десктопной и адаптивных версий)**

* Корректное отображение на всех возможных размерах окна браузера.
* Область нажатия кнопок.
* Отображение страницы при масштабировании в десктопных браузерах.
* Корректное отображение страницы при масштабировании в мобильных браузерах.
* Наличие мета-тега viewport.
* Корректная прокрутка страницы при открытых попапах.
* Корректное отображение плавающих (закрепляемых за прокруткой) элементов.
* Использование чрезмерно большого количества различных брейкпоинтов для стилей.
* Смешивание различных брейкпоинтов для стилей.
* Слишком узкие блоки на маленьких разрешениях экрана.
* Наличие сломанной вёрстки при изменении размеров экрана.
* Некорректная верстка на мобильных устройствах при показе/скрытии адресной строки.

**4. HTML**

* Валидность HTML.
* Наличие корректного DOCTYPE.
* Наличие корректной кодировки.
* Наличие тега title и мета-тегов для SEO.
* Наличие атрибута lang у тега html.
* Повторяющиеся или некорректные атрибуты id.
* Наличие пустых и ненужных тегов.
* Наличие объёмных комментариев в коде.

**5. Javascript**

* Не должно быть ошибок javascript при работе с сайтом.
* Скрипты должны быть объединены в один файл и минифицированы.
* Файл скриптов должен подключаться внизу страницы.
* Использование кода из неподходящей версии javascript.
* Сторонние скрипты желательно должны иметь атрибуты async и defer.
* Наличие устаревших javascript-плагинов.
* Корректное подключение сторонних библиотек.
* Некорректная работа плагинов.
* Наличие непереведенного текста в плагинах.
* Адекватное отображение сайта при выключенном javascript.
* Наличие кода на “боевом” сайте, предназначенного для разработки на тестовом сервере.
* Корректная инициализация контентных слайдеров.

**6. CSS**

* Валидация CSS.
* Файлы стилей должны быть корректно подключены.
* Стили должны быть объединены в один файл и минифицированы.
* Наличие в файлах стилей лишних правил для не поддерживаемых браузеров.
* Использование контентных тегов для стилизации.
* Стилизация элементов по атрибутам name или id.
* Использование одинаковых цветов, скруглений, отступов, размеров шрифтов.
* Использование unicode-символов в файлах стилей и скриптов.
* Артефакты, возникающие при использовании стилей transform.
* Дергающиеся и некорректно работающие анимации.
* Разные стили плавности анимации.
* Корректное отображение сайта с включенным режимом редактирования в CMS. (для сайтов разработанных на Worpress)
* Адекватные отступы между блоками контента.
* Слишком резкая граница для overflow: hidden.
* Наличие горизонтальной прокрутки.
* Большой разброс z-index.

**7. Фавиконки**

* Наличие favicon.ico и фавиконки больших размеров.
* Корректный список подключаемых фавиконок.
* Наличие manifest.json или manifest.webmanifest.

**8. Шрифты**

* Наличие малоиспользуемых или подключение неиспользуемых на сайте шрифтов.
* Правильное подключение шрифтов.
* Подключение шрифтов только из локальных источников.
* Предзагрузка шрифтов.
* Наличие правила font-display для подключаемых шрифтов.
* Наличие fallback-шрифтов.
* Указание наличия или отсутствия засечек при использовании шрифтов.

**9. Навигация**

* Битые ссылки.
* Ссылки, которые ведут на текущую страницу (на главной странице верхний логотип сайта не должен быть ссылкой).
* Корректная верстка меню.
* Корректная верстка “хлебных крошек”.
* Корректное отображение меню при различном количестве пунктов меню.
* Наличие стилей для индикации текущего элемента в навигации и неактивных элементов.

**10. Заголовки**

* Наличие одного тега h1 на странице.
* Семантичность заголовков (должны идти по порядку).
* Правильно настроенные стили и соотношение размеров заголовков.
* Заголовки должны использоваться только в контентной части, например, заголовки страницы.

**11. Контент**

* Стили для контентных элементов должны быть прописаны: параграфы, изображения, обычные и нумерованные списки, таблицы.
* Блоки сайта не должны расползаться при слишком больших размерах содержимого этих блоков.
* Необходимо тестировать сайт с реалистичными изображениями и текстами.
* Блоки должны корректно отображаться при любом количестве контента внутри них.
* Проверка орфографии, в том числе и в самом интерфейсе.
* Наличие clearfix у контейнеров с элементами со стилями float: left и float: right.
* Некорректная микроразметка.
* Корректное отображение валюты.
* Проверка вместимости длинных названий.
* Различная высота элементов в слайдерах.

**12. Изображения**

* Корректное распределение файлов изображений.
* Корректное подключение изображений.
* Наличие файла спрайтов для изображений интерфейса.
* Корректное подключение файла спрайтов.
* Корректное центрирование изображений в контейнерах.
* Некорректное содержимое svg-файлов.
* Корректные размеры изображений.
* Корректные стили для изображений.
* Адаптация изображений под мониторы высокого разрешения.
* Плохая оптимизация медиа-файлов.
* Отсутствие ленивой загрузки изображений.
* Изображения разных пропорций, загруженных в одной галерее.
* Наличие оптимизации изображений, загружаемых пользователями.
* Изображение капчи должно центрироваться относительно соседних блоков.

**13. Ссылки и кнопки**

* Наличие в ссылках на сторонние сайты атрибута rel="noopener".
* Выделение интерактивных элементов при наведении и фокусе.
* Телефоны и электронные почты должны быть прописаны как ссылки.
* Все нажимаемые области должны иметь cursor: pointer.
* Наличие атрибутов target=”\_blank”.
* Наличие корректных атрибутов type и role для кнопок и ссылок.
* Наличие специальных общих классов для кнопок и полей ввода.
* Стилизация кнопок, полей ввода и чекбоксов без помощи скриптов.
* Наличие пустых ссылок.
* Ссылки, ведущие на небезопасные ресурсы.
* Некорректное поведение кнопок на тач-устройствах.
* Незастиленные ссылки при наведении.
* Незастиленные ссылки при фокусе.
* Наличие неактивных кнопок, на которые можно нажать.

**14. Футер**

* Расположение футера, если контента меньше, чем на всю высоту экрана.
* Наличие фиксированной высоты у футера.
* Отступы перед футером должны быть одинаковые на всех страницах.
* Слишком маленький отступ внизу футера.

**15. Формы**

* Правильно прописанные заголовки полей (label).
* Проверка стилей кнопок и полей ввода.
* Наличие атрибутов для ограничения длины ввода.
* Наличие масок для полей ввода.
* Клиентская валидация полей ввода.
* Корректные атрибуты type у полей ввода.
* Корректная работа с русскими и английскими текстами при заполнении.
* Изменение размеров textarea не должно ломать верстку.
* Различные стили элементов форм в разных браузерах и устройствах.
* Наличие уведомления после отправления формы.
* Корректный вид уведомлений.
* Наличие клиентской валидации.
* Корректный сброс формы после отправки.
* Проверка отправки формы по нажатию Enter.
* Корректная работа форм при нажатию кнопки “Назад” в браузере.
* Некорректная повторная отправка форм.
* Слишком высокие формы в мобильных браузерах.

3. В каких браузерах и на каких устройствах необходимо тестировать сайт.

В разном окружении могут возникать абсолютно неожиданные ошибки, поэтому необходимо внимательно проверять все пункты чек-листа во всех возможных вариантах окружения.

Список браузеров и тестируемых устройств обязательно должен быть согласован между Исполнителем и Заказчиком перед началом разработки сайта.

В первую очередь мы тестируем сайт по всем пунктам чек-листа в браузере Chrome, так как этот браузер используется у разработчиков как основной. Затем тестируем сайт в остальный браузерах (последние стабильные версии):

**Десктопные браузеры:**

* Chrome
* Safari
* Firefox
* Edge
* Internet Explorer 11

Браузеры Yandex и Opera сделаны на движке браузера Chrome, поэтому на них отдельно мы не тестируем.

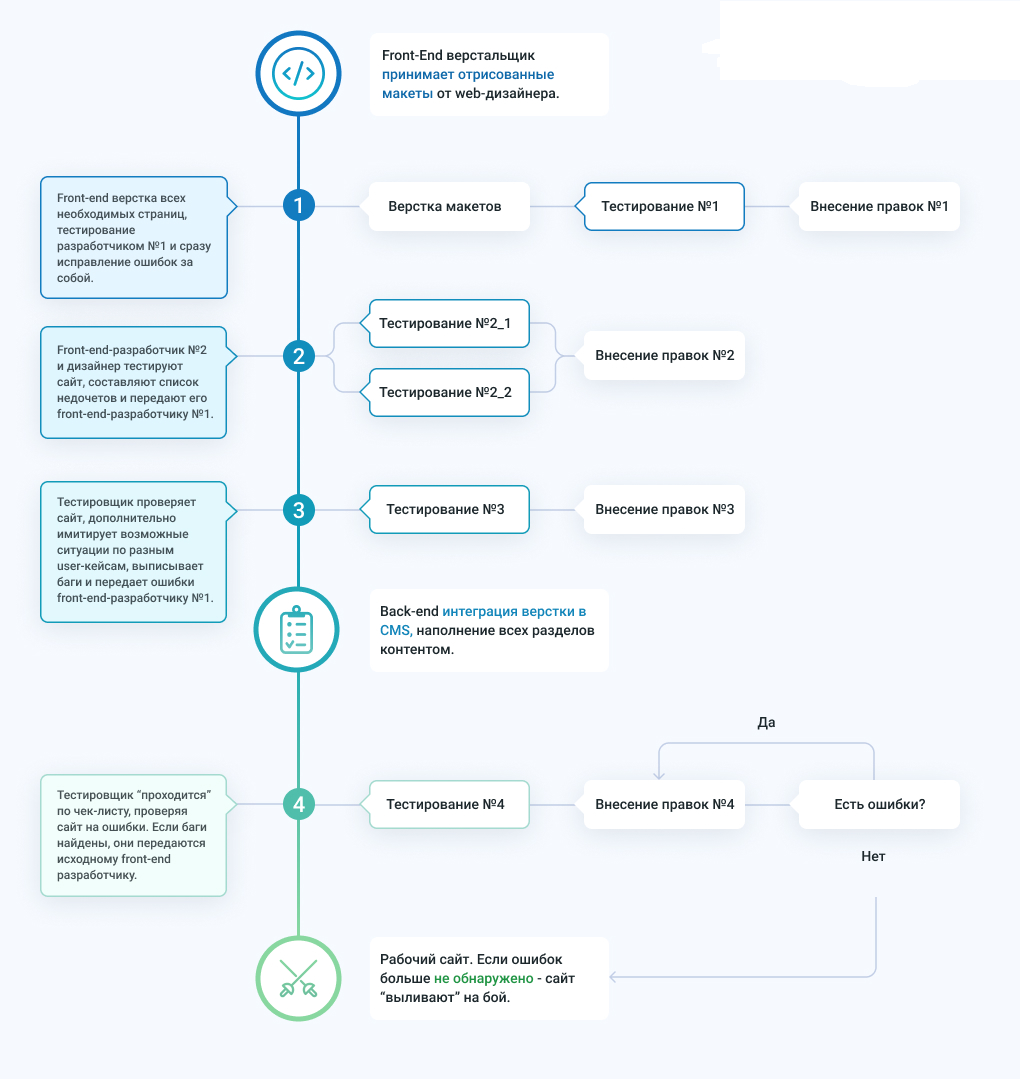
**Мобильные браузеры:**

* Safari
* Chrome

Все десктопные браузеры тестируем на операционной системе Windows, кроме Safari, который тестируем на MacOS.

Тестирование именно в этих браузерах связано с популярностью их использования. По статистике, самым популярным браузером является Chrome.

**Разработка сайта и тестирование (схема)**

****

**Пояснение к схеме:**

## Front-end

Дизайнер отрисовывает страницы сайта со всеми элементами (ui-kit), подготавливает исходники для отдела разработки и отправляет все макеты вместе с используемыми в проекте шрифтами, а также сопроводительную информацию с примерами эффектов, анимаций и тд.

**Итерация №1**

Проходим классический путь: html-верстка всех необходимых страниц, тестирование и сразу исправление ошибок за собой. Это все делает один и тот же frontend-разработчик.

**Итерация №2**

В этой итерации участвует уже другой frontend-разработчик и дизайнер, который рисовал проект. В процессе тестирования каждый составляет свой список недочетов, по которому впоследствие проходит frontend-разработчик и вносит нужные корректировки.

**Итерация №3**

Тестировщик №3 проверяет сайт по чек-листу, а также дополнительно имитирует возможные ситуации по разным user-кейсам, выписывает ошибки, после чего передает на доработку исходному frontend-разработчику. Разработчик вносит корректировки.

## Backend

Backend-разработчик интегрируют html-верстку в CMS. Сайт наполняется реальным контентом.

**Итерация №4**

Тестировщик “проходится” по чек-листу и проверяет сайт на ошибки. Если ошибки найдены, они передаются исходному frontend-разработчику. Frontend-разработчик вносит изменения в код. Затем тестировщик проверяет выполненную работу разработчика, снова “прогоняя” сайт по чек-листу. Если ошибок больше не обнаружено — сайт “выливают” на бой.

**Если ошибки обнаружены, они передаются frontend-разработчику и снова вносятся правки. Действие повторяется, пока ошибок больше не будет обнаружено.**

**Сервисы, которые используются нами при тестировании html-верстки**

Сервисы для проверки сайта на разных разрешениях экрана и разных устройствах:

* [iloveadaptive](http://iloveadaptive.com/) — уникальный сервис по разработке и тестированию адаптивной HTML-верcтки под все устройства.
* [BrowserStack](https://www.browserstack.com/) и [lambdatest](https://www.lambdatest.com/" \t "_blank) — SAAS-платформа для тестирования frontend на эмуляторах реальных устройств, запущенных в виртуальной машине сервиса.

**Сервисы для валидации:**

* [W3C validator](https://validator.w3.org/) — проверка соответствия HTML-кода страниц сайта и CSS-кода на корректность.
* [Яндекс.Вебмастер “Валидатор микроразметки”](https://webmaster.yandex.ru/tools/microtest/) — позволяет проверить семантическую разметку, внедренную на вашем сайте, и убедиться в том, что роботы поисковых систем смогут извлечь структурированные данные.
* [PerfectPixel](https://chrome.google.com/webstore/detail/perfectpixel-by-welldonec/dkaagdgjmgdmbnecmcefdhjekcoceebi?hl=en) — расширение для Chrome, позволяющий накладывать изображение макета поверх сайта, таким образом можно сравнить приближенность верстки к макету.

**Сервисы, помогающие понять, как оптимизировать сайт:**

* [Google Speed](https://developers.google.com/speed/pagespeed/insights/) — комплексный инструмент для определения производительности, удобства и выбора эффективных путей оптимизации сайта.
* [Think with google](https://thinkwithgoogle.com/feature/testmysite/) — проверяет удобство просмотра на мобильных устройствах, а также скорость работы страницы на мобильных и настольных устройствах.
* [Яндекс.Вебвизор](https://metrika.yandex.ru/promo/webvisor) — анализ поведения пользователей на сайте.

**Сервисы для написания автоматизированных тестов:**

* [Jest](https://jestjs.io/) — фреймворк для тестирования JavaScript, основным преимуществом которого является скорость и простота, а также широкий круг возможностей.
* [Selenium](https://www.selenium.dev/) — инструмент для автоматизации действий браузера и тестирования frontend.

**Сайты, показывающие работоспособность различных технологий в браузерах:**

* [CanIUse](https://caniuse.com/) — инструмент, благодаря которому можно узнать в каких десктопных и мобильных браузерах поддерживаются различные веб-стандарты.
* [Kangax.github.io](https://kangax.github.io/compat-table/es6/) — таблица работоспособности различных функций javascript на разных движках и в разных браузерах.

**SЕО-инструменты, проверка некорректных ссылок:**

* [Screaming Frog](https://www.screamingfrog.co.uk/seo-spider/) — SEO-инструмент, который позволяет находить некорректные и битые ссылки на сайте.
* [Xenu 2](http://home.snafu.de/tilman/xenulink.html) — бесплатная альтернатива предыдущему сервису, он проще устроен, но менее функционален.
* [Check My Links](https://chrome.google.com/webstore/detail/check-my-links/ojkcdipcgfaekbeaelaapakgnjflfglf?hl=en) — расширение для браузера Chrome, позволяет проверить текущую страницу на наличие “битых” или неправильных ссылок.

**Инструменты для минификации и оптимизации ресурсов:**

* [Terser](https://terser.org/) — библиотека для минификации скриптов.
* [Compressor.io](https://compressor.io/), [Image Compressor](https://imagecompressor.com/), [TinyPNG](https://tinypng.com/" \t "_blank) — сервисы для сжатия изображений с возможностью детальной настройки степени сжатия.