



Чек-лист по правильной верстке сайта

Как убедиться, что работа выполнена качественно? Как оценить качество вёрстки сайта? На самом деле, ответы на эти вопросы безумно просты и элементарны, ваша задача - всего лишь обратиться за помощью к нашему чек-листу по грамотной и эффективной верстке.

1. Соответствие макету

☐ 1.1 Соблюдение пропорций макета

Расположение блоков должно быть 1:1 по сравнению с макетом. Допускается расхождение до 5px для текста. Разрешены и даже приветствуются правки размеров и расположения криво нарисованных блоков (разница размерах в 1-2px на разных страницах).

2. Кроссбраузерность, кодировка и DOCTYPE

☐ 2.1 Учитывать кодировку UTF-8

Зачем нужно: UTF-8 это универсальность и совместимость. Это современный стандарт, за ним даже не будущее, а настоящее.

☐ 2.2 Посмотреть о DOCTYPE: HTML5

Наличие корректного doctype необходимо для отображения страниц в соответствии со стандартами. Новый doctype позволяет нам смело использовать современные тэги (canvas, header, article) и старые проверенные решения, ранее бывшие в опале (например embed). HTML5 - современный стандарт.

☐ 2.3 Проверить кроссбраузерность

Проверяется просмотром сайта в вышеперечисленных браузерах. Проверка в IE7 делается переключением IE8 в режим IE7 (F12→Режим обозревателя→Internet Explorer 7). В IE6 можно посмотреть на ipinfo.info/netrenderer или на виртуальной машине (удобно

использовать Windows XP Mode в Win7).

3. Валидность (включая CSSLint и JSHint), доступность, микроформаты

☐ 3.1 Проверить валидность титульника

Ошибки простительны в следующих случаях: -Секретарша "копирастит" тексты из Word'a в визиг; - Программеру ну очень нужны кастомные атрибуты (хотя для этого в HTML5 ввели специальные пользовательские атрибуты «data-»).*

☐ 3.2 Проверить наличие микроформатов

Как минимум – hCard. Желательно также hCalendar, XFN, hAtom.

☐ 3.3 Проверить качество кода

Обязательное условие - проверка статическими анализаторами качества кода: CSSLint и JSHint.

☐ 3.4 Определить наличие соответствия верстки стандарту доступности: WCAG

Он имеет три уровня сложности, если проходит хотя бы WCAG1 A (Priority 1) – уже хорошо. Идеальный вариант – WCAG2 Priority 3 (AAA). Самый простой способ проверить что скорее всего WCAG1 Priority A соблюден — www.cynthiasays.com (или [addon Web Developer](#) →Инструменты →Проверить WAI).

☐ 3.5 Уточнить независимость блоков в CSS: минимизация каскада

Проверяется в FF через плагин [addon Firebug](#). При наведении на любой блок, в его стилях не должно быть множество перечёркнутых правил. Для минимизации каскада и построения надёжной, современной, масштабируемой вёрстки сейчас применяют следующие техники: БЭМ, MCSS и SMACSS.

☐ 3.6 Оптимизировать визуализацию сайта

Сайт должен нормально смотреться во всех стандартных разрешениях от 1024 и выше, не иметь горизонтального скrolла и вписываться в экран мобильных устройств.

4. Корректная работа при вбивании реального текста,

надёжность вёрстки

☐ 4.1 Перепроверить порядок верстки

Вёрстка должна тянуться, не разваливаться и не терять дизайнерский вид при изменении контента на странице. Его может быть больше или меньше чем на макете, он может быть обёрнут во всякие

из визига и т.п. Обязательно нужно проверять удаление заголовков!

☐ 4.2 Проверить введение и удаление данных

Проверяется: на странице с контентом, пробуем добавлять и удалять содержимое – «что будет когда текста много?», «а когда мало?». Обязательно пробовать менять расположение элементов, чтоб после того как ты поменял блоки местами не развалилось оформление.

☐ 4.3 Удостовериться в корректности работы стилей

Проверяется: на страницы с контентом вбиваем текст с абзацами и без абзацев (важно! бывает горе-верстальщики прописывают стили только для абзацев), со списками и картинками, таблицами и заголовками разных уровней.

☐ 4.4 Использовать препроцессоры и системы сборки

CSS должен быть написан с использованием препроцессоров (LESS/Sass/Stylus).

Проверяется поиском файлов с расширениями вида: .less, .sass, .scss, .styl — какое-то одно должно быть.

5. Оптимизация скорости загрузки

☐ 5.1 Ускорить работу

Скорость загрузки оказывает ключевое влияние на доступность сайта (больше психологическую, чем фактическую), активность пользователей на сайте (медленными сайтами люди предпочитают не пользоваться) и его конверсию (медленным сайтам не доверяют).

☐ 5.2 Сохранить доступность при выключенных(загружающихся) картинках

Надписи (особенно логотип и главное меню сайта) должны оставаться читабельными, у всех информационных картинок должны быть подписи аккуратным небольшим серым шрифтом (да, для `img` можно задавать `font` – это внешний вид `alt`-текста, что выводится

вместо картинки).

6. HTML5 формы, линковка, валидация

☐ 6.1 Слинковать Label и input/select

Это нужно для удобства юзеров. Также это очень облегчает жизнь пользователям с ограниченными физическими возможностями. Проверяется кликом по label – должен активироваться соответствующий ему элемент ввода.

☐ 6.2 HTML5 валидация- правильность заполнения формы

Проверяется в Opera: выключаем javascript, не заполняем форму, жмём Submit – должны появиться уведомления о необходимости заполнить поля.

☐ 6.3 Проверить правильность JS-валидации формы

Это ожидаемое поведение. Пользователи привыкли что если они неправильно заполняют форму, им не дадут её отправить, а укажут на ошибки. Проверяется в Opera/Safari/Chrome: включаем javascript, не заполняем форму, жмём Submit – должны появиться уведомления о необходимости заполнить поля.

☐ 6.4 Точность в input type="email/url/tel".

Пока-что практическая ценность для пользователя лишь в том, что на iPhone будет показываться клавиатура соответствующая формату поля ввода. Проверяется на iPhone — в зависимости от типа поля ввода он должен показывать различную клавиатуру: стандартную/цифровую/для набора web/email-адресов.

☐ 6.5 Убедиться в работоспособности при выключенном JavaScript

JS может быть выключен согласно корпоративных требований безопасности. А в Opera Mini он работает только методом перезагрузки страницы. Но самое главное — сайт должен сохранять нормальный вид, пока он грузится на медленном 3G и js-скрипты ещё не выполнились!

☐ 6.6 Проверить работоспособность при выключенном Flash

Flash не работает на iOS-девайсах. Flash плохо индексируется поисковиками.

Проверяется в FF отключением flash-плагина: Tools→Add-ons→Plugins→Shockwave Flash→disable.

☐ 6.7 Отсутствие багов при увеличенном шрифте

Проверяется в FF: Шрифты – включением View→Zoom→Zoom Text Only и последовательным нажатие клавиш Ctrl + — (или View→Zoom→Zoom In). -120 dpi – настраивается отдельным апплетом в Control Panel (Vista/Seven) или в свойствах

драйвера видеокарты в XP.

☐ 6.8 Учесть нюансы

Для проектов, где это оговорено, проверяется: -Версия для печати (она должна быть реализована через *css*) -Мобильная версия (таки тоже должна быть через *css*)

☐ 6.9 Обратить внимание на важные мелочи

-Лого на внутренках должно вести на титулку. На титулке *logo* = *h1*, на внутренках *H1*=заголовок контента, а *Logo*=*div*; -Проверять что все интерактивные элементы страницы что должны работать - работают; -В вёрстке не должны оставаться закомментированные «на всякий случай» куски кода, лишние файлы и т.д.

Создано с помощью онлайн сервиса Чек-лист | Эксперт: <https://checklists.expert>

как это убрать?