

Ответы на вопросы к квалификационному заданию для разработчиков HTML/CSS.

Q1. Расскажите, чем, на ваш взгляд, отличается хорошая верстка от плохой с точки зрения пользователя, менеджера проекта, дизайнера, верстальщика, клиентского программиста и серверного программиста.

Хорошая верстка с точки зрения

1. Пользователя:

- Корректно отображается на любом устройстве.
- Загружается быстро, без видимого построения элементов.
- Нет зависаний/фризов при взаимодействии со страницей.
- Идентично или схоже отображается в различных браузерах.

2. Менеджера проекта:

- Соответствует заданному ТЗ и макету.
- Имеет возможно быстрого внесения правок.
- Имеет положительный отзыв у заказчика.
- Не создает сложностей другим участникам разработки.

3. Дизайнера:

- Близка или идентична созданному макету.
- Все интерактивные элементы которой совпадают с задумкой дизайнера.

4. Верстальщика:

- Имеет поясняющие комментарии для лучшей поддержки кода.
- Содержит понятный и легко читаемый код.
- Соответствует единому подходу к написанию разметки, стилей и скриптов.

5. Клиентского программиста:

- Удобна для манипуляции отдельными блоками и элементами (не имеет проблем с отображением при удалении отдельных частей, можно легко получить доступ к изменению стилей и скриптов)
- Адаптирована под контент различного размера и типа.
- Имеет понятное именование сущностей.

6. Серверного программиста:

- Не перегружает сервер "качественно" (часть вычислительных операций берут на себя скрипты на клиентской стороне).
- Не перегружает сервер "количественно" (присутствует компановка изображений и файлов для единой загрузки, имеется предзагрузка файлов).

Q2. Опишите основные особенности верстки крупных многостраничных сайтов, дизайн которых может меняться в процессе реализации и поддержки. Расскажите о

своем опыте верстки подобных сайтов: какие методологии, инструменты и технологии вы применяли на практике.

Основной особенностью верстки крупных многостраничных сайтов, дизайн которых может меняться в процессе реализации и поддержки, является применение повторно используемых компонентов/блоков. Эти блоки должны быть независимы от позиционирования на странице для облегчения поддержки.

Еще одной важной особенностью является использование переменных для определения стилей элементов страницы и их переиспользования, а так же миксинов для их генерации.

Помимо соблюдения принципа DRY немаловажным является использование системы контроля версий для быстрого перехода между нужными этапами разработки.

Для поддержки будет полезным использование единых стандартов разработки.

При разработке верстки я использую сборщик Gulp, препроцессор Sass(sass), плагины в IDE для быстрой замены частей кода во всем проекте, линты для js и css.

Q3. Опишите основные особенности верстки сайтов, которые должны одинаково хорошо отображаться как на любом современном компьютере, так и на смартфонах и планшетах под управлением iOS и Android. Расскажите о своем опыте верстки подобных сайтов: какие инструменты и технологии вы применяли, как проверяли результат на различных устройствах, какие именно устройства требовалось поддерживать.

Под различные разрешения устройств зачастую используют видоизменение дизайна страницы посредством адаптивной верстки. Для этого, в зависимости от задачи, я использую flexbox, grid css, bootstrap-grid и media-запросы.

Для правильного масштабирования страниц использую мета тег viewport с соответствующими свойствами.

Чтобы избежать исправления моих стилей браузером, использую css reset и заранее через плагин проставляю вендорные префиксы к свойствам.

Если есть изображения/иконки низкого разрешения, которые скейлятся с лесенками на мобильных устройствах, то по возможности перевожу их в формат svg. Если же такой возможности нет, то делаю подгрузку различных разрешений изображения под различные разрешения экрана.

Актуальные разрешения экранов для построения сетки я проверяю на сайте <http://gs.statcounter.com/screen-resolution-stats>.

Точность верстки я проверяю через pixel-perfect расширения для браузеров, а эмулирую устройства через Google DevTools Device Mode. При возникновении спорных моментов использую сервис <https://www.browserstack.com/responsive>.

Q4. Расскажите, какие инструменты помогают вам экономить время в процессе написания, проверки и отладки кода.

* Таск-раннер - Gulp

- * Сборщик модулей - Webpack
- * Система контроля версий - Git
- * Пакетный менеджер - npm
- * DevTools браузеров
- * Препроцессор css - Sass
- * Шаблонизатор и препроцессор html - Pug
- * Линтеры: JSLint и Stylelint
- * Проверка совместимости свойств и параметров - сервис <https://www.caniuse.com>.
- * Проверка актуальных разрешений экранов - <http://gs.statcounter.com/screen-resolution-stats>.
- * Тестирование через эмуляцию - <https://www.browserstack.com/responsive>.
- * Плагины для текстового редактора. Например, emmet, prettier, eslint, git history, htmltagwrap, TODO highlight, autofilename и прочие
- * Горячие клавиши в текстовом редакторе.
- * Тестирование JavaScript - Jest (+ Enzyme)

Q5. Вам нужно понять, почему страница отображается некорректно в Safari на iOS и в IE на Windows. Код писали не вы, доступа к исходникам у вас нет. Ваши действия? Сталкивались ли вы с подобными проблемами на практике?

Если есть доступ к элементам страницы и стилям через инструменты разработчика, то буду пробовать исправлять ошибки там. Некорректное отображение в Safari на IOS скорее всего будет связано с высоким PPI устройства. IE - самый проблемный браузер, ошибки до 9 версии возможно придется исправлять через условные комментарии или хаки. С подобными проблемами сталкивался, чаще всего несоответствия ожиданию проявляются в IE и Firefox.

Q6. Дизайнер отдал вам макет, в котором не показано, как должны выглядеть интерактивные элементы при наведении мыши. Ваши действия?

Отмечу у себя, что это спорный момент, отправлю сообщение с вопросов дизайнеру и продолжу работу над версткой. Если моя работа над макетом данного дизайнера будет походить к концу, а ответа от дизайнера не будет, то спрошу у проектного менеджера или тимлида.

Если в команде есть соглашение, что все пробелы подобного рода заполняются через соблюдение правил style guide или ui kit, то сделаю в соответствии с ними.

Q7. Какие ресурсы вы используете для развития в профессиональной сфере? Приведите несколько конкретных примеров (сайты, блоги и так далее). Какое направление развития вам более близко: JS-программирование, HTML/CSS-верстка

или и то, и другое? Какие ещё области знаний, кроме тех, что непосредственно относятся к работе, вам интересны?

Примеры ресурсов:

Справочники: [developer.mozilla](#), [learn.javascript](#), ...

Поиск решений: [stackoverflow](#), [toster](#), ...

Статьи: [habr](#), [medium](#), ...

Курсы: ~~изучаю сливы платных курсов~~

Youtube каналы: [freeCodeCamp](#), [ITVDN](#), [loftblog](#), [skillbox](#), [codedojo](#), Хауди Хо, Гоша Дударь, ...

Мне ближе JS-программирование, чем верстка.

Помимо javascript мне интересен UX-дизайн, английский язык, back-end разработка, алгоритмы и математика.

Q8. Расскажите нам о себе и предоставьте несколько ссылок на последние работы, выполненные вами.

С ранних лет увлекаюсь информатикой, математикой и программированием. Начинал свой путь с изучения Basic и Fortran. Позже перешел на Pascal, C и затем на C++. В старших классах увлекся web разработкой и понял, что это мое.

С тех пор пробовал писать на десятке языков, от python и php до prolog и языков ассемблера.

Учебу в университете рассматривал для себя только на кафедре, связанной с IT, поэтому поступил в МГТУ им Н.Э. Баумана на "Компьютерные системы и сети".

Почти все свободное время уделяю изучению новых технологий и практике. За последний год изучил более 10 платных курсов, связанных с JavaScript (React, Redux, Node.js и т. д.).

Крупных работ в портфолио нет, часть проектов залиты на github.