*1. Расскажите, чем, на ваш взгляд, отличается хорошая верстка от плохой с точки зрения:*

* Пользователя – Хорошая: красивый необычный (и в тоже время простой) вид, понятный интерфейс и быстрая загрузка сайта. Пользовать хочет открывать одинаково быстро сайт на всех устройствах и понимать, что до него пытались донести. Плохая: сложная перегруженная информацией верстка, которая к тому же долго грузиться и некорректно отображается на всех устройствах.
* Менеджера проекта – Хорошая: верстка, созданная в точности по заданию утвержденному с заказчиком, которую сдали в срок. Верстка должна отвечать всем современным тенденциям – UI/UX (по дизайну), кроссбаузерность, адаптивность. Обязательно выдержана семантика кода. Также, верстка должна быть сео оптимизированной. Верстка должна работать на устройствах с отключенным js. Плохая верстка – противоположность вышеизложенным тезисам.
* Дизайнера:

Хорошая: реализованная по принципам UI/UX. Верстка должна быть гибкой, чтобы легко вносить изменения в дизайн. Плохая: например, чтобы добавить новый блок, стилизовать сайт под какой-либо праздник надо проделать колоссальный объем работы.

* Верстальщика – Хорошая: структурированный, чистый и валидный код, семантически верная, адаптивная и кроссбраузерная верстка, быстрая скорость загрузки. Использование БЭМ в задании имен. Пример хорошей верстки – мне дали чужой код и задание внести правки. За 10 минут я должен понять код и начать реализовывать поставленную задачу. Плохая – антипод хорошей вёрстки. Сложно вносить правки в код – легче переделать, чем копаться в коде.
* Клиентского программиста – Хорошая: простота и понятность кода. Минимальное использование сторонних скриптов, быстрая скорость загрузки страницы. Адаптивность и кроссбраузерность верстки. Пример хорошей верстки – делали новостной сайт, который в дальнейшем размещался на Битрикс. Было важно верстать блоки независимыми друг от друга, и компоновать стили функциональных блоков в отдельных css файлах. Плохая – «сырая верстка», с кучей костылей, неадаптивная и неправильно отображается в браузерах.
* Серверного программиста. Хорошая: понятный и чистый код. Возможность вносить изменения в код. Серверный программист должен понимать куда в верстке выводиться информация с сервера, чтобы блоки были готовы принимать контент (текст изображение) различный по размеру и верстка не «ехала». Пример хорошей верстки – делали новостной сайт, который в дальнейшем размещался на Битрикс. Было важно верстать блоки независимыми друг от друга, и компоновать стили функциональных блоков в отдельных css файлах.

*2. Опишите основные особенности верстки крупных многостраничных сайтов, дизайн которых может меняться в процессе реализации и поддержки.*

1. Функциональные блоки сайта должны быть независимыми. Тоесть, если создать блок (например, слайдер) на отдельной страницы, то он должен себя вести также, когда был интегрирован в верстку. Также, это позволить использовать идентичные блоки на разных страницах. При внесении правок в css код, изменения коснуться всего проекта.
2. Если удалить/добавить кусок html кода, верстка не должна разъехаться.
3. Использование методологию БЭМ (или другую).
4. Использование git репозитория для наличия резервных копий, чтобы иметь возможность проверять (или откатить) изменения

Опыт – информационный портал для города Сургут. (<https://github.com/1bodro/newsSite-Surgut>). Верстал под битрикс. На страницах сайта (главная, новости, пост, мнения экспертов и т.п.) повторяющиеся куски кода. Была задача их css код группировать.

*3. 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*.

Описанные требования сейчас актуальны для всех сайтов. Отличия могут заключаться на что больше ставка заказчика – на пользователей сайта с мобильных устройств (mobile first) или десктопной версии сайта. Сайт должен быть легким адаптивным и кроссбразуреным. Выполняю действия по уменьшению времени загрузки сайта, уменьшая размер файлов (CSS,js, imj, png для сайта сжимаю), количество запросов к серверу (для небольших изображений использую спрайты, объединяю стили css и js скрипты). Проверяю скорость загрузки на трех ресурсах и сравниваю результаты – [google pagespeed](https://developers.google.com/speed/pagespeed/insights/), [pingdom](https://tools.pingdom.com/), [webpagetest](https://www.webpagetest.org/). Для сжатия использую pug. На небольших проектах компрессию изображений делаю через [TINYpng](https://tinypng.com/).

Для старых версий сайта с помощью gulp или [автопрефикс](https://autoprefixer.github.io/ru/) проставляю префиксы на сайте. В некоторых случаях использую определение устройтсва через css @media или js и заданию стили под необходимый браузер (конкретной версии) или под конкретное устройство.

После верстки тестирую сайт различных реальных устройствах – телефоны, ноутбуку и пк. Для проверки сайта использую встроенных инструменты.

Отдельные сложности – верстка под apple устройства. Сталкивался в некорректным отображением сайта на старых айфонах. В той ситуации выручил ресурс – [appetizo](https://appetize.io/).

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

До закрытия проекта adobe assets помогал вытягивать стили из макета. Теперь в поисках новых инструментов и использую чистый photoshop. В скорости работы помогают макеты в figma.

На скорость работы влияют плагины редакторов кода vs code (или brackets), встроенные инструменты разработчиков в браузеры, когда отрыл для себя gulp – понял, что он также может экономить львиную долю времени. Указанные в 3 вопросы ресурсы + w3c (для валидации кода) – также экономят мне время при работе.

*5. 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?*

На фирме давали задание – предыдущий верстальщик используя бутстрап, сдал заказчику проект. После принятия работы, посыпались правки. Их переадресовали мне. Исходников нет, но исправлять надо. Общался с заказчиком напрямую. Он мне писал, что не нравиться, а я вносил правки в html и css. Из-за бутстрапа верстка некорректно отображалась на планшетах и телефонах, видео блоки были не адаптивны, и т.п. Для IOS и safari использовал – [appetizo](https://appetize.io/).

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

Если есть контакт с дизайнером, то уточню наличие макетов. Если их нет, то могу предложить свои варианты, которые надо будет утвердить с заказчиком.

*7. 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?*

Информацию черпаю из книг и интернет ресурсов

* Книг авторов: Илья Кантор, Морган, Дэвид Макфарланд, Этан Браун, Ник Морган и т.п.
* Интернет ресурсов: coursehunters, codelearn, stepik, learn.javascript, code.mu, habr, dou.ua, Сайт веб-документации MDN, geekbrains, proglib.io и другие.

Интересно в общем данное направление. Сейчас точного разделения между версткой и фронтенд частью нет. Привлекает фронтенд разработка/верстка – это созидание продукта, большой объем информации, которую можно учить и применять на практике. Наличие простой и понятной формулы данной профессии: больше знаешь – больше зарабатываешь, больше знаешь – интересней работать.

У меня существует много интересующих меня тем. Выделю две: горный туризм и электрика(автоматика). Занимаюсь пешим горным туризмом, много раз был в крымских и кавказких горах, Карпатах. Электрика – «понимать» электричество, чинить технику настраивать автоматику. Как по мне много схожего с программированием, поскольку эти понятия для многих неизвестные материи.

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

Мои работы находятся по ссылке на мой репозиторий 🡪 <https://github.com/1bodro?tab=repositories>

Последние   
<http://hostest.kl.com.ua/alliexpress/>

<http://surgut.kl.com.ua/index.html>

<http://site-testing-2.kl.com.ua>

О себе:

Моя страница в вк <https://vk.com/id4402925>

Закончил университет. Стал специалистом автоматическое управление процессами. Работал в разных сферах, от электрослесаря на шахте, до инженера пищевой промышленности. Из-за ситуаций в регионе, желанию развиваться, изучать новую информацию решил стать frondend engineer.

Почему именно данное направление? Посоветовался с друзьями, сказали сможешь реализовать свой креатив в данной профессии, а если после изучения станет скучно, то можешь постепенно изучать phyton. И началось. С азами программирования я знаком давно и не по «слухам». В коллеже, в университете была заложена основательная база алгоритмизации, логики работы языков. На том этапе разобрал принципы программирования «на винтики», поэтому проблем при изучении верстки (html, css) и сопутствующего языка js не возникло.

По состоянию на сегодняшний день, за плечами прочитанные книги, прохождение курсов, просмотры обучающих видео, вебинаров и интеративов, работа верстальщиком на компанию Hotel Technology Company. За время работа в данном направлении понял, что сделал правильный выбор сменив курс своего развития, совершенствую свои навыки и знания для покорения новых горизонтов.

Имеется опыт работы контент менеджером и дизайнером, который помогает при работе верстальщиком.