**Стефанова Дарья, 191-361**

**Лабораторная работа 3. Ответы на вопросы**

1. **Какие CSS фреймворки вам удалось изучить. В чем их особенности? Плюсы и**

**минусы использования CSS фреймворков.**

Был изучен Bootstrap. Он удобен, отлично подходит для начинающих разработчиков, можно быстро создать прототип сайта, хорошо работает на разных устройствах. Из минусов: сайты похожи друг на друга, отсутствие гибкости, сайты могут некорректно отображаться в старых браузерах. Так же была изучена библиотека

1. **Какие JS фреймворки вам удалось изучить. В чем их особенности? Плюсы и минусы**

**использования CSS фреймворков.**

Был частично изучен Vue.js, который идеально подходит для создания высокоадаптируемых пользовательских интерфейсов и сложных одностраничных приложений. Плюсы: подробная документация, адаптивность, масштабирование, небольшой вес. Минусы: достаточно молодой, есть более расширенные и функциональные фреймворки

1. **Изложите принципы адаптивного веб-дизайна (RWD, Responsive Web Design).**

* По мере уменьшения размера экрана контент начинает занимать большую высоту экрана, и нижележащие элементы при этом могут отодвигаться все дальше от нижней границы области просмотра. Это называется «*потоком*».
* Сайт должен одинаково хорошо выглядеть не только на больших десктопных мониторах и на крошечных телефонах, но и на всём многообразии устройств. Нужную гибкость дизайну может придать использование относительных единиц измерения.
* Точки разрыва помогут изменить расположение элементов дизайна при соответствии экрана заданным условиям.
* Использовать максимальные и минимальные размеры.
* поместить группу элементов в блок-контейнер, и в дальнейшем управлять позицией и размером контейнера. Этот приём также упрощает работу с элементами, имеющими абсолютные размеры.
* Если логотип сайта содержит множество деталей и графических эффектов, то используйте его растровую версию. В противном случае попробуйте сохранить его в **SVG**.

1. **Изложите основные принципы методологии БЭМ. Какие другие методологии верстки**

**существуют? Какие подходы к структуре кода, именованию стилей CSS вы будете**

**применять при разработке?**

Основные принципы: страница делится на блоки, в каждом блоке содержатся элементы, которые не используются вне блока. Состояния и поведение блока и элемента можно задавать с помощью [модификатора](https://ru.bem.info/method/key-concepts/#%D0%9C%D0%BE%D0%B4%D0%B8%D1%84%D0%B8%D0%BA%D0%B0%D1%82%D0%BE%D1%80).

Другие методологии: OOCSS, SMACSS, Atomic CSS, MCSS, AMCSS, FUN.

Подходы к структуре кода: элементы должны зависеть только от их родителей и иметь говорящие названия, располагать свойства в разных блоках в одном порядке.