</> Критерии вёрстки

1. Выполнена HTML-разметка всех страниц проекта и всех элементов на этих страницах. Критерий говорит о том, что все страницы проекта и все скрытые и дополнительные элементы должны быть размечены. Например, всплывающие и появляющиеся элементы, модальные окна, все слайды в слайдере.

   Важно: тег `<a>` используем только для ссылок — переходов на другие страницы. Если кнопка или текст должны открывать модальное окно или иная реакция на странице, то это уже не ссылка, а кнопка, для неё используем теги, например, `<span>`.

   Запрещено использование: `<a href="javascript:void(0)">Попап</a>`

2. Грубые ошибки в разметке отсутствуют.
   Грубые ошибки:
   - Ссылки сделаны не тегом `<a>`, а другими тегами;
   - Не ссылки сделаны тегом `<a>`;
   - Использование строчных элементов для создания крупных (сеточных) блоков;
   - Абзацы сделаны не тегами `<p>`, а `<br><br>`.

   Негрубые ошибки:
   - Отсутствие семантических тегов `<header>`, `<footer>`, `<section>` и других.

   Важно: тег `<a>` — элемент с «прозрачной» моделью содержимого. Поэтому если `<a>` вложен в элемент с фразовым содержимым, например в `<span>`, то внутри самого `<a>` может быть только фразовый контент. А если `<a>` вложен в элемент с потоковым содержимым, например в `<div>`, то и содержимое у `<a>` может быть потоковое, включая другой `<div>`.

3. Документ проходит проверку на валидность https://validator.w3.org/nu/.

4. В разметке отсутствует дублирование кода для одного и того же элемента, с помощью которого элемент отображается в разных местах страницы на разных версиях: мобильной, десктопной, планшетной. Этот критерий не касается элементов, которые скрываются или показываются в разных версиях.

5. Отсутствуют типовые ошибки в разметке по методологии.
   Отсутствуют следующие ошибки методологии БЭМ:
   - Создание элемента без родительского блока. Это означает, что не может быть элемента `block__element`, если выше по дереву нет DOM-элемента с именем block.
   - Создание элемента для элемента.
   - Создание модификатора для модификатора.
   - Использование модификатора без блока или элемента, который он модифицирует (при использовании модификатора у тега должно быть как минимум два класса: класс блока/элемента и класс модификатора).

   Допустимые стили именования БЭМ-сущностей:
   - Международный: `block__elem--mod`
   - Классический: `block__elem_mod` или `block__elem_mod_value`

6. Названия полей форм привязаны к своим полям с помощью `<label>`.

7. Раскладка блоков на странице сделана с помощью флексбоксов.
   Использование тега `<table>` и блоков с абсолютным позиционированием недопустимо. `<table>` является устаревшим методом построения сеток, а элементы с абсолютным позиционированием вырываются из общего потока. Допускается использование этих методов для создания декоративных элементов и модальных окон.
   Использовать табличную раскладку блоков с применением свойств `display: table, display: table-row, display: table-cell` и так далее не запрещено.

8. В CSS отсутствует `!important`.
   Допускается использование `!important` при обосновании его необходимости в комментарии.

9. Подключены правильные шрифты, их размеры, цвет и толщина равны соответствующим параметрам в макетах и техническом задании.

10. Нестандартные шрифты подключены локально. Формат шрифтов должен быть woff2 и woff.

11. Указаны альтернативные варианты шрифта и тип семейства в конце перечисления font-family.
    Альтернативный веб-безопасный шрифт и тип семейства необходимо указывать для того, чтобы в случае отсутствия основного шрифта изменения внешнего вида шрифтов на странице были минимальны.
    Порядок шрифтов следующий:
    1. Основной шрифт;
    2. Веб-безопасный;
    3. Тип шрифта.

    Список веб-безопасных шрифтов можно посмотреть здесь.

12. При наполнении контентом (как в макете) элементы каждой версии страницы (мобильной, планшетной и десктопной) соответствуют макету.

13. Код стилей должен быть разбит на несколько частей.
    Используя CSS-препроцессор, нужно разделить код таким образом, чтобы каждый блок был в своём файле. Для каждого БЭМ-блока нужно создать отдельный файл стилей.
    Исключение возможно для минимума глобальных стилей (стилизация по тегам, они должны быть описаны в одном файле), для CSS-нормализатора и для селекторов модульной сетки.
    Все файлы должны импортироваться в главный стилевой файл, который будет компилироваться в CSS-файл.

14. Выполнена вёрстка трёх состояний каждой страницы: мобильной, планшетной и десктопной.

15. В разметке есть правильный вьюпорт тег.

16. Для микросеток использованы флексбоксы.

17. Вёрстка идентично отображается в последних версиях браузеров Chrome, Firefox, Safari, Edge.

18. Единообразное написание и форматирование кода в HTML, файлах CSS-препроцессора и JavaScript (включая файлы автоматизации).
    Критерий рассматривает единообразие в написании кода:

- используется один тип кавычек: в одном языке (HTML, CSS или JS) должны использоваться только кавычки определённого типа (двойные или одинарные). Однако между языками тип кавычек может отличаться;
- если размер отступа в два пробела, то таким он должен быть везде;
- если для отступов используются табы или пробелы, то для всех отступов должны быть либо табы, либо пробелы;
- названия классов должны быть оформлены единообразно;
- свойства, которые поддерживают несколько наборов значений (например, множественные фоны и множественные тени, нужно делать в одном формате: однострочном или многострочном).

Важно: этот критерий учитывает именно единообразие, а не стиль написания и форматирования кода.

19. Отсутствует транслит в названиях классов, атрибутах, переменных CSS-препроцессора, названиях примесей и так далее.

20. Проект соответствует техническому заданию.

21. У всех векторных изображений размер прописан в теге `<img>`, у встроенных SVG-изображений размер прописан в теге `<svg>`.

22. Использовано минимально возможное количество HTML-элементов (нет лишних элементов).

23. Для стилизации не использованы #id.

24. Для блока, у которого есть фоновое изображение, прописан фоновый цвет, который соответствует преобладающему цвету изображения (пока изображение не загружено, страница выглядит похоже на макет).

25. Все состояния элементов (смотрите styleguide) прописаны в стилевом файле.

26. Нет глобальных стилей тегов.
    Исключения:
    - normalize.css, который исправляет браузерные умолчания;
    - Уникальные теги документа: html, body;
    - Дополнительная нормализация: a и img;
    - Общее правило для box-sizing с помощью inherit.

27. Запрещено использовать цветовые функции для изменения цветовых значений в коде.
    Если в макете указаны конкретные цвета, нужно задавать конкретные цветовые значения в коде. Не нужно подбирать эти значения с помощью цветовых функций CSS-препроцессоров (изменение цвета, осветление, затемнение и так далее).
    Исключение составляют функции изменения альфа-канала (прозрачности цвета) — их использовать можно. Это функция rgba() в Less и функции opacify() и transparentize() в Sass.

28. Примеси не используются для генерации правил с вендорными префиксами.

29. Вложенность селекторов не больше двух уровней.

30. Родительский селектор & используется только для псевдоэлементов, псевдоклассов и модификаторов.
    Использование & для комбинации селекторов не допускается в именах блоков и элементов.

Комбинировать можно только псевдоэлементы, псевдоклассы и модификаторы блоков и элементов.

31. Не используются расширения (extend).
    Функции &:extend в Less и @extend в Sass запрещены.

32. Вёрстка проходит тест на переполнение контентом.

33. Все интерактивные элементы имеют текстовое описание.
    Интерактивные элементы, представленные на макете только графически, без текста, должны содержать текстовое описание. Тогда при чтении или прослушивании интерфейса пользователь будет понимать, что произойдёт при нажатии на элемент.
    К интерактивным элементам относятся те элементы страницы, которые реагируют на взаимодействие с пользователем: например, клик по иконке закрытия попапа закрывает попап, клик по иконкам социальных сетей вызывает переход на другую страницу и так далее.

34. Код соответствует правилам в Stylelint.

35. Произведена оптимизация загрузки шрифтов.
    Шрифты предзагружаются через link rel="preload".
    Используется подходящее значение font-display в описании @font-face.

36. При взаимодействии с элементами (наведение, нажатие) ни сам элемент, ни окружающие его блоки не меняют своего положения (если иное не прописано в техническом задании).

37. При выборке элементов из DOM-дерева и работе с ними, сперва нужно убедиться, что элемент в DOM есть, так как при интеграции в CMS бэкенд-разработчик может забыть вставить элемент, прописать не так класс или иные причины. Попытка работы с node-элементом, которого нет, приводит к ошибкам интерпретатора JavaScript.

38. Тоже самое относится и при работе с различными типами объектов и обращении к их свойствам, особенно это важно при работе с API. Перед обращением к свойству объекта нужно проверить его ниличие через hasownproperty или аналогичным образом.

39. Контекст поиска (при выборке в DOM) увеличивает производительность. Всегда при поиске эелментов по селекторам нужно стараться искать не каждый раз в document, а внутри какого-нибудь враппера или элемента над которым произошло событие.

40. Убирать отладку console.log

41. В финальной вёрстке в console не должно быть ошибок