**Базовые критерии успешной сдачи макета**

**Разметка**

* Б1. Выполнена HTML-разметка всех страниц и всех элементов на страницах.

Критерий требует, чтобы все страницы личного макета и все элементы на этих страницах были сверстаны.

***Указания к проверке***

При проверке этого критерия следует проверять наличие всех модальных окон и других скрытых элементов, если такие имеются в макете. Внимательно стоит проверять исходный код студента, так как скрытое модальное окно может находиться в неочевидном месте. Критерий не требует, чтобы все элементы имели эффекты наведения, а их размеры совпадали с макетом.

* Б2. К страницам подключён один стилевой файл (с учётом normalize.css или reset.css к каждой странице могут быть подключены два файла).

Критерий говорит о том, что все стили страниц должны находиться в едином файле. Исключением могут являться файлы со сбросом стилей normalize.css или reset.css. Такой файл может быть подключен отдельно, перед основным стилевым файлом.

Верно: к документу подключаются два файла - normalize.css и основной стилевой файл.

<head>

<link rel="stylesheet" href="css/normalize.css">

<link rel="stylesheet" href="css/style.css">

</head>

Неверно: к документу подключается несколько стилевых файлов. Один из них содержит общие стили, второй стили конкретной страницы.

<head>

<link rel="stylesheet" href="css/normalize.css">

<link rel="stylesheet" href="css/all.css">

<link rel="stylesheet" href="css/main-page.css">

</head>

* Б3. Стилевой файл подключён внутри <head>.

Подключение стилевых файлов должно производиться внутри <head>.

Верно: стилевой файл подключен в секции <head>.

<!DOCTYPE html>

<html>

<head>

...

<link rel="stylesheet" href="css/style.css">

</head>

<body>

...

</body>

</html>

Неверно: стилевой файл подключен в секции <body>

<!DOCTYPE html>

<html>

<head>

...

</head>

<body>

<link rel="stylesheet" href="css/style.css">

...

</body>

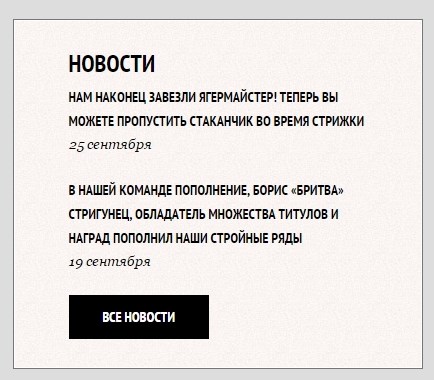
</html>

***Указания к проверке***

Данный критерий рассматривает только подключение стилевого файла внутри <head>, при этом порядок, в котором находятся остальные элементы внури <head>, не учитывается.

* Б4. Грубые ошибки в разметке отсутствуют. Например:
  + Грубые ошибки:
    - Ссылки сделаны не тегом <a>, а другими тегами;

Для создания определенных элементов на странице необходимо использовать соответствующие теги. Для примера рассмотрим блок с новостями, на главной странице макета Barbershop. В данном случае, ссылка на все новости хоть и выглядит как кнопка, но на самом деле является ссылкой.



***Указания к проверке***

При возникновении спорных ситуаций, в которых вам кажется, что данный элемент должен быть сделан другим тегом, следует обращаться в ТЗ.

* + - Использование строчных элементов для создания крупных (сеточных) блоков;

Использование строчных тегов для построения сетки недопустимо. Для построения сетки должны использоваться блочные элементы.

Верно: сетка построена с использование блочных элементов.

<div class="features">

<div class="features-item">

...

</div>

<div class="features-item">

...

</div>

<div class="features-item">

...

</div>

</div>

Неверно: сетка построена с использованием строчных элементов, которые впоследствии будут изменены на блочные средствами CSS.

<div class="features">

<span class="features-item">

...

</span>

<span class="features-item">

...

</span>

<span class="features-item">

...

</span>

</div>

* + - Абзацы сделаны не тегами <p>, а <br><br>.

Если в макете текст разделен на несколько абзацев, то в разметке каждый такой абзац должен быть обёрнут в соответствующий тег. Использовать для разбиения на абзацы <br> недопустимо.

Неверно: для разбиения на абзацы используется <br>.

Наша парикмахерская занимается исключительно мужскими стрижками.

<br><br>

Стрижка каждого клиента для нас - это уникальная и продуманная до мелочей работа.

<br><br>

Наша мастерская расположена в центре города, поэтому сделать стильную стрижку можно в любое время.

Верно: Каждый абзац в своём теге <p>

<p>Наша парикмахерская занимается исключительно мужскими стрижками.</p>

<p>Стрижка каждого клиента для нас - это уникальная и продуманная до мелочей работа.</p>

<p>Наша мастерская расположена в центре города, поэтому сделать стильную стрижку можно в любое время.</p>

Не путайте эту ситуацию с использованием   
внутри абзацев для разбиения текста на строки, например, внутри адресов.

Верно: Весь адрес внутри <p> разбит на строки с помощью <br>.

<p>191186,<br>

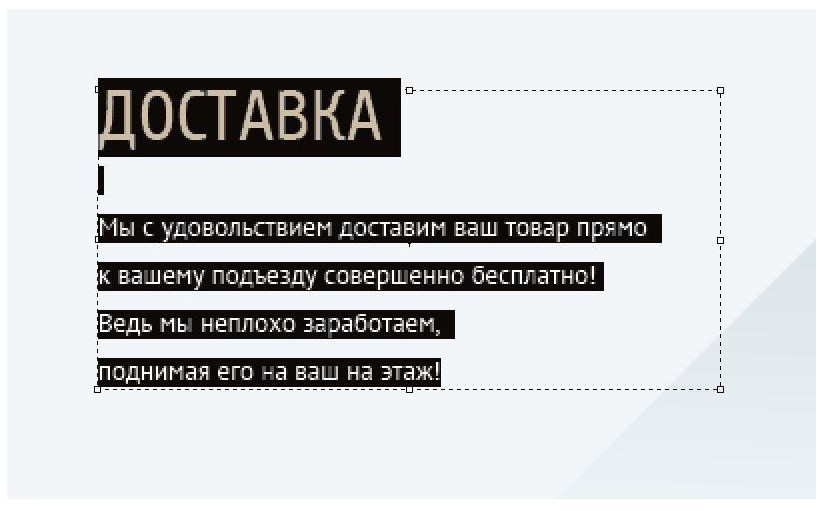
Санкт-Петербург,<br>

Большая Конюшенная, 19/8

</p>

Так же разрешено использовать <br> внутри абзацев, если нет возможности сделать переносы как в макете путем ограничения ширины контейнера с текстом.

Пример: текст под заголовком имеет явные переносы, которые нельзя сделать простым ограничением ширины.



***Указания к проверке***

При определении наличия в кода тега <br>, необходимо убедиться в том, что его использование необходимо. Например, в большинстве случаев, перенос текста внутри параграфа можно осуществить ограничив его размер. Если вместо этого используется <br>, то это тоже можно считать ошибкой.

**Например:**

<p>Стрижка каждого клиента для нас - это уникальная<br>

и продуманная до мелочей работа. Мы не работаем на<br>

количество, мы делаем качество.</p>

Если в определенном параграфе проекта нельзя осуществить перенос без использования тега <br>, либо перенос строки нужен для читабельности (адреса, стихи), то это не следует считать ошибкой.

* + Негрубые ошибки:
    - Отсутствие семантических тегов <header>, <footer>, <section> и других;

Критерий говорит о том, что отсутствие семантических тегов не является ошибкой. Оба приведенных ниже примера являются допустимыми.

Разметка выполнена с использованием семантических тегов

<header class="main-header">

<div class="container">

<nav class="main-navigation">

...

</nav>

</div>

</header>

Разметка выполнена без использования семантических тегов

<div class="main-header">

<div class="container">

<div class="main-navigation">

...

</div>

</div>

</div>

***Указания к проверке***

Если в разметке неверно использованы семантические теги, то это является ошибкой.

* + - Ошибки касающиеся SEO-оптимизации.

***Указания к проверке***

Если у студента нарушены базовые принципы SEO оптимизации, например, отсутствуют заголовки или нарушен их порядок, то это не следует считать грубой ошибкой. Если вы уверены, что студент совершил такую ошибку, вы можете указать ему на нее как рекомендацию к исправлению, но настаивать на исправлении не стоит.

* Б5. Документ проходит проверку на валидность [http://validator.w3.org/nu/](https://validator.w3.org/nu/" \t "_blank).

***Указания к проверке***

При проверке данного критерия следует опускать ошибки, которые возникают в подключаемых модулях. Например, шрифтов с Google Fonts или интерактивных карт. Также ошибками не стоит считать замечания, помеченные валидатором как Warning.

**Стилизация**

* Б6. Вся собственная стилизация выполнена в одном стилевом файле.

Все стили всех страниц должны находиться в едином файле. Нет необходимости разносить различающиеся файлы разных страниц в отдельные файлы и подключать их по необходимости.

Исключением могут являться файлы с подключением шрифтов, если они грузятся со стороннего сервиса, например, Google Fonts. В таком случае эти файлы могут идти отдельно и не являться частью основного файла стилей.

***Указания к проверке***

В случае если студент использует несколько файлов и собирает их в один с помощью import в css, это тоже является ошибкой.

Подключение шрифтов с помощью Google Fonts, что влечет за собой подгрузку еще одного стилевого файла, не является ошибкой.

* Б7. Раскладка блоков на странице сделана не с помощью таблиц или позиционирования.

Для построения основной сетки страницы необходимо использовать float или inline-block элементы. Допускается использование и продвинутых техник (CSS-таблицы, флексбоксы), если студент в них разбирается и хочет попрактиковаться в вёрстке с использованием этих свойств.

Использование тега <table> и блоков с абсолютным позиционированием недопустимо. <table> является устаревшим методом построения сеток, а элементы с абсолютным позиционирование 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.

Верно: Основная сетка сделана с использованием float элементов.

<div class="features clearfix">

<div class="features-item">

...

</div>

<div class="features-item">

...

</div>

<div class="features-item">

...

</div>

</div>

.features-item{

float: left;

width: 300px;

padding: 0 10px;

}

Неверно: Основная сетка сделана с использованием таблиц.

<table class="features">

<tr>

<td class="features-item">

...

</td>

<td class="features-item">

...

</td>

<td class="features-item">

...

</td>

</tr>

</table>

Неверно: Основная сетка сделана с использованием таблиц.

<div class="features">

<div class="features-item features-1">

...

</div>

<div class="features-item features-2">

...

</div>

<div class="features-item features-3">

...

</div>

</div>

.features{

position: relative;

}

.features-item{

width: 300px;

position: absolute;

top: 0;

}

.features-1{

left: 0;

}

.features-item{

left: 300px;

}

.features-item{

right: 0;

}

***Указания к проверке***

Использование <table> приветствуется для разметки табличных данных, но не сетки. Если у студента используются сетки с использованием display: table, то это не является ошибкой при проверке финального макета.

* Б8. В CSS отсутствует !important. Допускается использование !important при обосновании его необходимости в комментарии.

В большинстве случаев, необходимость использования !important в CSS возникает из-за неправильного обращения к элементам страницы через каскад. Однако, в редких случаях, без использования !important не обойтись. Если без использования !important не обойтись или использование осознанно, то это не ошибка. Подробнее в статье: <http://meiert.com/en/blog/20150310/important/>

Неверно: В коде CSS используется !important

.text-page p{

font-size: 14px!important;

line-height: 20px!important;

}

***Указания к проверке***

Необходимо не только искать !important, но и оценивать целесообразность его использования. В подавляющем большинстве случаев его использование не будет целесообразным, но проверять необходимо.

* Б9. Подключены правильные шрифты, их размеры и толщина соответствуют размерам в макетах и ТЗ.

Необходимо, чтобы в личном проекте использовались именно те шрифты, которые указаны в макете. При этом их размеры, высота строки и жирность должна соответствовать параметрам, указанным в макете.

***Указания к проверке***

При проверке этого критерия следует ориентироваться именно на параметры, указанные в макете. Из-за особенностей отображения шрифтов на разных платформах, определение этого критерия на глаз может привести к ошибке. Помимо этого, стоит допускать разное межбуквенное расстояние в макете и в финальной верстке. Считать разное межбуквенное расстояние ошибкой стоит только в том случае, если оно явно указано в макете.

* Б10. 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: <http://www.cssfontstack.com/>

**Важно:** альтернативный веб-безопасный шрифт должен быть такого же типа, что и основной, но подбирать максимально похожий шрифт не нужно.

Верно: Указан альтернативный веб-безопасный шрифт и его тип семейства.

body {

font-family: 'PT Sans Narrow', 'Arial', sans-serif;

}

/\* Кому-то нравится Arial, кому-то Tahoma \*/

body {

font-family: 'PT Sans Narrow', 'Tahoma', sans-serif;

}

Неверно: Указан только основной шрифт.

body {

font-family: 'PT Sans Narrow';

}

Неверно: Указан только основной шрифт и тип семейства, альтернативный веб-безопасный шрифт отсутствует.

body {

font-family: 'PT Sans Narrow', sans-serif;

}

Неверно: Times New Roman — шрифт с засечками, а основной шрифт — без засечек.

body {

font-family: 'PT Sans Narrow', 'Times New Roman', sans-serif;

}

***Указания к проверке***

Так как дополнительный шрифт не указывается в ТЗ, в случае, если студент укажет непохожий на ваш взгляд дополнительный шрифт, это не стоит считать ошибкой.

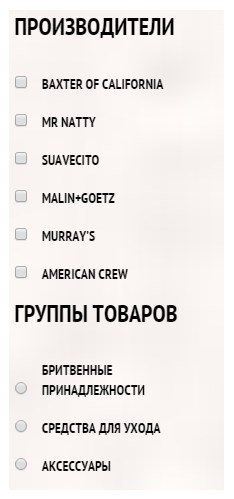
В случае если студент укажет неверный тип семейства - это следует считать ошибкой.

* Б11. При наполнении контентом (как в макете) элементы каждой страницы соответствуют макету, но допускаются:
  + Различия в 5 пикселей по высоте (при расстояниях более 30 пикселей) и 2 пикселя по ширине;
  + Отсутствие стилизации кастомных элементов форм;

Разрешается оставлять не стилизованными только элементы форм checkbox, radio и select.

Для стилизации select использовать JS-библиотеки запрещается. Подробнее:<http://habrahabr.ru/company/htmlacademy/blog/257743/>

Пример: Фильтры полностью сверстаны, использованы верные шрифты и отступы, но checkbox и radio оставлены без дополнительной кастомизации.



* + Различия в отображении шрифтов, связанные со сглаживанием на различных платформах.

**Тестирование**

* Б12. Вёрстка идентично отображается в последних версиях браузеров Chrome, Opera, Firefox, Safari, а также в Internet Explorer 10+.

При проверке этого критерия, необходимо смотреть на размеры и расположение блоков, внешнее сходство с макетом. Проверять работу анимации, если такая имеется. Допускаются небольшие отличия в отображениях шрифтов.

***Указания к проверке***

Тестирование необходимо проводить именно в последних версиях браузеров, которые предоставляют поставщики, а не те, которые установлены в данный момент на компьютере проверяющего. Тестирование на мобильных устройствах проводить не надо.

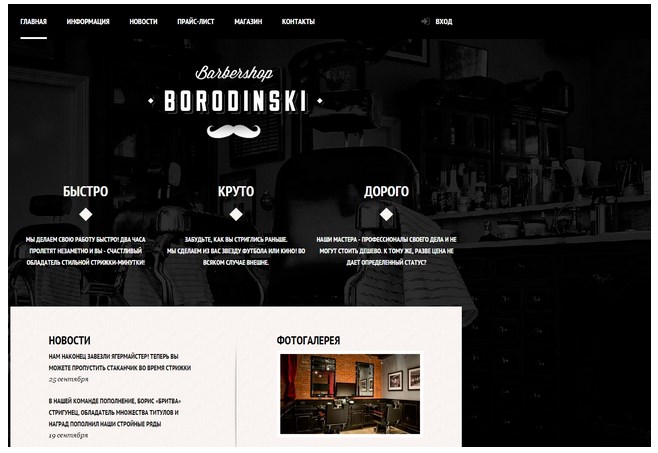
**Важно:** для пользователей Windows последняя версия браузера Safari - 5, реальная же версия этого браузера на данный момент 9, поэтому проводить тестирование в нём на Windows не надо.

* Б13. Сайт нормально смотрится на минимальном для макета разрешении:
  + Нёрдс — 880px (900px с учётом padding)
  + Техномарт — 940px (960px с учётом padding)
  + Седона — 768px (для фиксированной можно использовать либо версию 768px, либо 1200px)

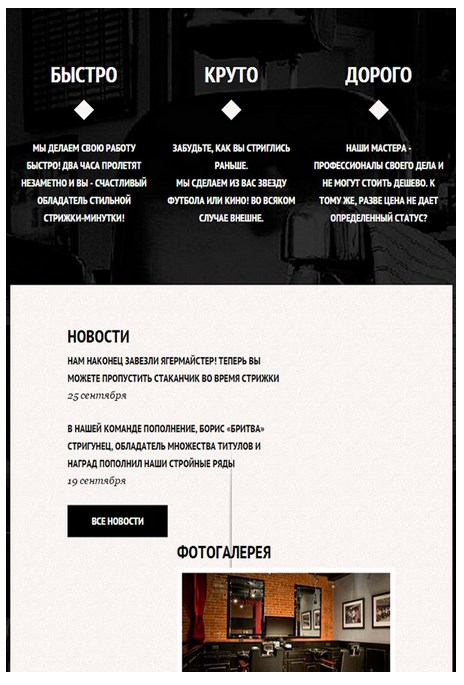
*padding по краям контейнера указывается для того, чтобы контент не прижимался к краям экрана на минимальном разрешении. При большем размере экрана макет должен оставаться по центру и не иметь горизонтального скролла. На разрешениях экрана меньше ширины контейнера вёрстка не должна менять свою структуру.*

Критерий рассматривает поведение страницы при размере экрана большим или меньшим, чем ширина основного контейнера страницы.

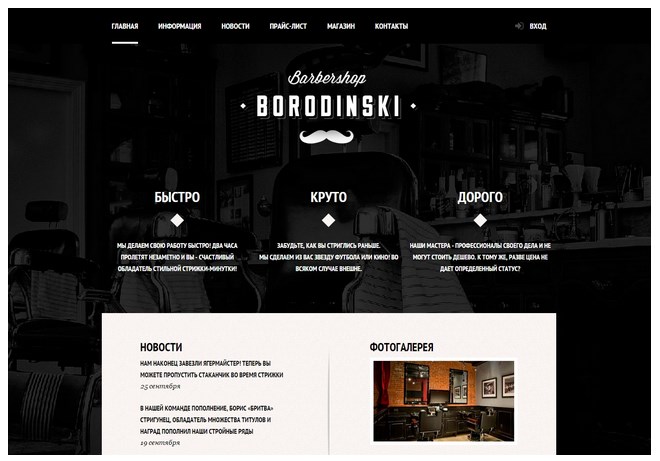
Неверно: При большем размере экрана страница остается слева, а не выравнивается по центру.



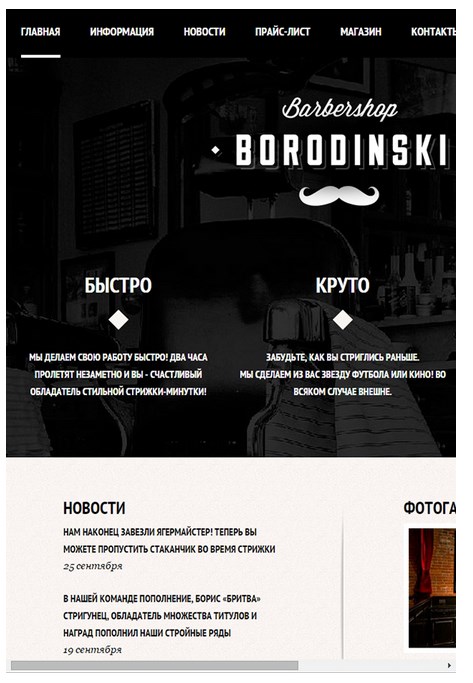
Неверно: При меньшей ширине экрана объекты страницы начинают менять свой размер и ломать сетку.



Верно: При большем размере экрана страница остается по центру экрана и имеет фиксированную ширину.



Верно: При меньшей ширине экрана страница не меняет свой размер, а внизу окна браузера появляется горизонтальный скролл для ее прокрутки.

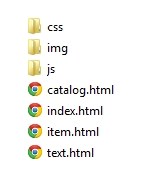


**Разное**

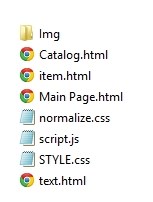
* Б14. В корне документа имеются папки css, img, js или аналогичные. Главная страница имеет название index.html. В названиях и расширениях файлов нет заглавных букв и пробелов, использованы только латинские символы.

Критерий рассматривает файловую организацию проекта. Все файлы должны лежать в соответствующих папках и иметь читаемые и единообразные написанные названия.

Верно: Все файлы находятся в своих папках. Названия этих папок отражает тип файлов, которые в ней содержатся. Главная страница имеет название index.html. Все названия написаны маленькими буквами и не имеют пробелов.



Неверно: Файлы смешаны в одну кучу. В именах используются заглавные буквы и пробелы. Главная страница имеет неверное название.



***Указания к проверке***

Допускаются альтернативные варианты названий для папок со стилями, скриптами и картинками.

* Б15. Единообразное написание и форматирование кода в HTML, CSS и JavaScript.

Критерий рассматривает единообразие в написании кода:

* + если используются двойные кавычки, то они используются везде;
  + 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.

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

Верно: Единообразное написание кода разметки. Везде используется одинаковое количество пробелов, одинаковые скобки, единообразное написание названий классов.

<header class="main-header">

<div class="container clearfix">

<nav class="main-navigation">

<ul>

<li class="active">

<a href="index.html">Главная</a>

</li>

<li>

<a href="#">Информация</a>

</li>

<li>

<a href="#">Новости</a>

</li>

<li>

<a href="text.html">Прайс-лист</a>

</li>

<li>

<a href="catalog.html">Магазин</a>

</li>

<li>

<a href="#">Контакты</a>

</li>

</ul>

</nav>

<div class="log-in">

<a href="#" class="log-in-button">Вход</a>

</div>

</div>

</header>

Верно: Единообразное написание кода стилей. Используется одинаковое количество пробелов, одинаковое положение скобок, единообразная последовательность написания свойств.

.btn{

font-size: 14px;

line-height: 24px;

width: 100%;

padding: 10px 15px;

cursor: pointer;

text-align: center;

color: #fff;

}

.main-navigation a{

font-size: 16px;

line-height: 20px;

display: block;

padding: 30px 0 23px;

text-decoration: none;

color: #fff;

border-bottom: 4px solid transparent;

}

Неверно: Используются разные кавычки и разное количество пробелов в табуляции. В именах классов нет единообразия.

<header class="main-header">

<div class="container clearfix">

<nav class='mainNavigation'>

<ul><li class="active">

<a href="index.html">Главная</a>

</li>

<li>

<a href="#">Информация</a>

</li>

<li><a href=''>Новости</a></li>

<li>

<a href="text.html">Прайс-лист</a>

</li>

<li>

<a href="catalog.html">Магазин</a>

</li>

<li>

<a href="#">Контакты</a>

</li>

</ul>

</nav>

<div class='logIn'>

<a href="#" class="log-in-button">Вход</a>

</div>

</div>

</header>

Неверно: Используется разное количество пробелов в табуляции, скобки в разных местах. Написание свойств перемешано.

.btn{

width: 100%;

font-size: 14px;

line-height: 24px;

padding: 10px 15px;

cursor: pointer;

text-align: center;

color: #fff;

border: none;

outline: none;

background: #0000000;

text-decoration: none;

text-transform: uppercase}

.main-navigation a

{

display: block;

padding: 30px 0 23px;

font-size: 16px;

line-height: 20px;

text-decoration: none;

color: #ffffff;

border-bottom: 4px solid transparent;

}

***Указания к проверке***

Если стиль кодирования вам не нравится, но он применяется единообразно, то критерий нужно засчитывать. Например, ниже представлен редкий стиль оформления CSS, но он единообразен и поэтому допустим.

.btn { width: 100%; padding: 10px 15px; font-size: 14px; line-height: 24px; }

.main-navigation a { display: block; padding: 30px 0 23px; font-size: 16px; line-height: 20px; }

* Б16. Выбран подходящий формат изображений, например:
  + JPEG для фотографий;
  + PNG или SVG для всех прочих.

**Дополнительные критерии**

**Разметка**

* Д1. У всех изображений в теге <img> прописан размер.

Должны быть указаны действительные размеры картинок без указания размерности.

Верно: Размеры картинке заданы.

<div class="logo-area">

<img src="img/logo-big.png" alt="" width="300" height="150">

</div>

Неверно: Размеры картинке заданы с размерностью в px

<div class="logo-area">

<img src="img/logo-big.png" alt="" width="300px" height="150px">

</div>

Неверно: Размеры картинке не заданы.

<div class="logo-area">

<img src="img/logo-big.png" alt="">

</div>

***Указания к проверке***

Должны быть указаны именно действительные размеры картинок.

* Д2. Использовано минимально возможное количество HTML-элементов (нет лишних элементов).

В разметке должно быть использовано минимально возможное количество элементов. Не должно быть лишних оберток и блоков, которые используются для оформления и могут быть заменены на псевдоэлементы.

Верно: Не используется лишняя обертка, а для декоративного элемента нет дополнительного блока в разметке.

<div class="features clearfix">

<div class="features-item">

<h2>Быстро</h2>

<p>Мы делаем свою работу быстро!</p>

</div>

...

</div>

Неверно: Используется лишняя обертка, которую можно сократить, а для декоративного элемента использован тег <i>, который можно заменить на псевдоэлемент.

<div class="features">

<div class="clearfix">

<div class="features-item">

<h2>Быстро</h2>

<i class="triangle"></i>

<p>Мы делаем свою работу быстро!</p>

</div>

</div>

...

</div>

***Указания к проверке***

В случае, если в верстке осознанно и единообразно разделяются сеточные и смысловые блоки - это ошибкой не является.

* Д3. Все скрипты подключены непосредственно перед </body>.

Подключение скриптов в самом низу страницы необходимо для того, чтобы загружающийся скрипт не блокировал содержимое страницы.

Верно: скрипт подключается перед закрывающим тегом <body>

<!DOCTYPE html>

<html>

<head>

...

</head>

<body>

...

<script src="js/script.js"></script>

</body>

</html>

Неверно: скрипт подключается в секции <head>

<!DOCTYPE html>

<html>

<head>

...

<script src="js/script.js"></script>

</head>

<body>

...

</body>

</html>

***Указания к проверке***

В случае, если в макете отсутсвует js, этот критерий не следует засчитывать.

* Д4. Названия полей форм привязаны к своим полям с помощью <label>.

Для улучшения взаимодействия пользователя с элементами форм должна быть реализована активировация поля в случае нажатия на его описание. Для этого необходимо связывать элемент формы с его описанием, используя тег </label> и идентификаторы.

Верно: Элемент формы radio связан с его описанием через идентификатор.

<input type="radio" id="radio1">

<label for="radio1">Радио кнопка</label>

Верно: Элемент формы radio и подпись обёрнуты в <label>

<label>

<input type="radio"> Радио кнопка

</label>

Неверно: Описание никак не связано с элементом формы.

<input type="radio" id="radio1"> Радио кнопка

**Стилизация**

* Д5. Использованы normalize.css или reset.css (но не оба сразу).

Для выполнения этого критерия необходимо, чтобы был использован файл normalize.css или reset.css.

Верно: Файл normalize.css подключается отдельным файлом перед основным файлом стилей.

<link rel="stylesheet" href="css/normalize.css">

<link rel="stylesheet" href="css/style.css">

**Неверно:** Файл сброса стилей не подключается.

<link rel="stylesheet" href="css/style.css">

По желанию содержимое normalize.css или reset.css можно включить в начало основного стилевого файла.

***Указания к проверке***

В макете необходимо использовать только один из файлов сброса normalize.css или reset.css. Одновременное использование этих файлов является ошибкой. Студент может включить файл сброса в состав своего основного файла стилей, поэтому при проверке необходимо проверить его содержимое. Стоит обратить внимание, что стили сброса должны идти до основных стилей, вне зависимости от того, как организовано их подключение.

* Д6. Для стилизации не использованы #id.

Для стилизации объектов лучше использовать селекторы по классам или тегам. Использовать id для стилизации недопустимо.

Важно: Использовать атрибут id в HTML-разметке можно и нужно. Например, для привязки полей к подписям в формах.

Верно: Стилизация элементов через классы.

.feedback{}

.feedback p{}

.main-header{}

Неверно: Стилизация элементов и каскад строится через идентификаторы.

#feedback{}

#feedback p{}

#main-header{}

***Указания к проверке***

Критерий рассматривает именно запрет на стилизацию через id, но использовать этот атрибут для связывания label и элементов форм не считается ошибкой. Также не считается ошибкой использование этого атрибута для реализации функционала, например, в слайдере.

* Д7. Нет вложенности селекторов больше двух.

Длинные цепочки вложенных селекторов усложняют код и его поддержку. Хорошим подходом считается использовать вложенность 2 или 3 уровня. Селекторы с вложенностью не более 2 — это наш маленький, но важный каприз.

Псевдоэлементы или псевдоклассы не увеличивают уровень вложенности.

**Важно:** допускается использовать селекторы с вложенностью больше 2 при оживлении слайдеров на чистом CSS.

Верно: Максимальная глубина селекторов не превышает два элемента.

.catalog{}

.catalog-item img{}

.catalog-item .item-description{}

.catalog-item h3{}

.catalog-item:nth-child(odd):hover .item-description::after{} /\* это тоже 2 уровня вложенности \*/

Неверно: Глубина селекторов больше двух, но при этом их можно сократить.

.catalog{}

.catalog .catalog-item img{}

.catalog .catalog-item .item-description{}

.catalog .catalog-item h3{}

***Указания к проверке***

Псевдоэлементы и псевдоклассы не добавляют уровень вложенности.

Допускается использовать селекторы с вложенностью больше 2 при оживлении слайдеров на чистом CSS. При наличии таких селекторов просто игнорируйте их при оценке этого критерия.

* Д8. Явно указано подходящее vertical-align для потоковых блоков с display: inline-block.

По умолчанию элементы inline-block имеют выравнивание baseline, которое может приводить к нежелательному поведению элементов, например, когда внутри них разное количество строк.

Поэтому нужно явно и осознанно задавать выравнивание таким элементам.

Например, если вы хотите, чтобы элементы вели себя как текст, стоит явно указать значение baseline. Такое указание позволит другим понять, что вы знаете об особенностях поведения элементов с этим выравниванием и задали его намеренно.

**Верно:** Вертикально выравнивание задано явно.

.login-button{

display: inline-block;

vertical-align: baseline;

}

**Неверно:** Вертикально выравнивание не задано.

.login-button{

display: inline-block;

}

* Д9. Для CSS-свойств с префиксом указан его вариант без префикса и это указание идёт последним.

Вендорные префиксы обеспечивают кросбразерную работу некоторых свойств CSS. Какие именно свойства необходимо использовать с префиксами, можно посмотреть [здесь](http://shouldiprefix.com/).

Свойство без префиксов, обязательно должно идти после всех свойств с префиксами.

**Верно:**

.class{

-webkit-box-sizing: border-box;

-moz-box-sizing: border-box;

box-sizing: border-box;

}

**Неверно:**

.class{

box-sizing: border-box;

-webkit-box-sizing: border-box;

-moz-box-sizing: border-box;

}

***Указания к проверке***

В случае, если в проекте не используются префиксы, этот пункт необходимо засчитывать и не считать ошибкой.

* Д10. Для блока у которого есть фоновое изображение прописан фоновый цвет, который соответствует преобладающему цвету изображения (пока изображение не загружено, страница выглядит похоже на макет).

Такой метод использует для перестраховки, как и в случае со шрифтом. Только в данном случае, если фоновое изображение не загрузится, на заднем фоне останется преобладающий цвет.

Верно: Указан цвет фона перед основным изображением.

body{

background: #000 url('../img/background-body.jpg') no-repeat center top;

}

Результат: Картинка не загрузилась, но при этом цвет фона остался темным.

Неверно: Указано только основное изображение.

body{

background: url('../img/background-body.jpg') no-repeat center top;

}

Результат: Картинка не загрузилась, а цвет фона стал белым.

***Указания к проверке***

Для того, чтобы этот критерий был засчитан, необходимо не просто указать цвет фона, но и подобрать цвет, который преобладает в самом изображении. Для картинок с полупрозрачностью или прозрачностью данный критерий применять не надо.

* Д11. Все состояния элементов (смотрите styleguide.psd) прописаны в стилевом файле.

В соответствии с файлом styleguide, который присутствует в каждом проекте, все указанные в нём элементы должны иметь соответствующие эффекты при наведении и нажатии. Должны присутствовать все активные состояния, а для чекбоксов и радио кнопок прописаны состояния disabled.

***Указания к проверке***

При проверке этого критерия следует оценивать не только наличие эффектов, но и их правильность.

* Д12. Файл стилей представлен в двух вариантах: с форматированием и минимизированный, к странице подключён минимизированный.

Обычный CSS код.

.feedback p{

font-size: 22px;

color: #000;

line-hight: 30px;

}

.feedback p{

font-size: 14px;

line-hight: 20px;

}

Сжатый CSS код

.feedback p{font-size: 22px;color: #000;line-hight: 30px;}.feedback p{font-size: 14px;line-hight: 20px;}

**JavaScript**

* Д13-1. Собственные скрипты собраны в отдельном JavaScript-файле, который подключён в HTML перед закрывающим тегом body.
* Д13. JavaScript-файл представлен в двух вариантах: с форматированием и минимизированный, к странице подключён минимизированный.

Обычный JS код.

var open = document.querySelector(".map-btn");

var popup = document.querySelector(".feedback");

var close = popup.querySelector(".feedback-circle-btn");

open.addEventListener("click", function (event) {

event.preventDefault(event);

popup.classList.add("feedback-show");

});

close.addEventListener("click", function (event) {

event.preventDefault(event);

popup.classList.remove("feedback-show");

});

Сжатый JS код

var open=document.querySelector(".map-btn"),popup=document.querySelector(".feedback"),close=popup.querySelector(".feedback-circle-btn");open.addEventListener("click",function(e){e.preventDefault(e),popup.classList.add("feedback-show")}),close.addEventListener("click",function(e){e.preventDefault(e),popup.classList.remove("feedback-show")});

***Указания к проверке***

Существуют различные степени сжатия. В данном критерии степень сжатия не принципиальна.

* Д14. С помощью JavaScript реализовано открытие/закрытие окна с формой (без вспомогательных библиотек).
* Д15. С помощью JavaScript добавлена анимация формы (без вспомогательных библиотек).
* Д16. Добавлена интерактивная карта (допускается подключение с помощью iframe).

***Указания к проверке***

Карта может быть реализована как через iframe так и через API Google или любых других карт. При любой реализации не обязательно ставить на карте такой маркер, который указан в макете.

**Тестирование**

* Д17. Вёрстка проходит тест на переполнение контентом:
  + Не ломается при добавлении в элементы большего количества текста;

**Верно:** Текст растягивает блок.

**Неверно:** Текст выпадает за пределы блока со светлым фоном.

* + Не ломается при использовании картинок с неподходящими размерами;

Верно: Добавление картинки не сломало сетку, она вписалась в соответствующий размер.

Неверно: Добавлений большей картинки сломало сетку.

***Указания к проверке***

При проверке переполнения в браузере Safari необходимо добавлять текст не через инспектор браузера, а через редактор кода, редактируя страницу HTML. В противном случае текст будет добавляться в одну длинную строчку.

* + Не ломается при изменении количества потоковых блоков;

Верно: Переполнение верхнего блока растягивает его и двигает нижний.

Неверно: Переполнение верхнего блока ломает всю сетку документа.

* *Текст не выпадает из блоков, нижерасположенные блоки не скрываются, смещение блоков в потоке сохраняет логику потока (не приводит к нарушению сетки).*
* Д18. Критический функционал сайта работоспособен без JavaScript (использовано прогрессивное улучшение), например:
  + Все формы являются работоспособными без JavaScript;
  + Элементы, вызывающие появление попапов, являются ссылками, ведущими на отдельные страницы;
  + Интерактивная карта без JavaScript показывает статичную картинку с картой.

**Разное**

* Д19 Отсутствуют непонятные сокращения и транслит в названиях классов, атрибутах, названиях файлов и так далее.

Для написания классов необходимо использовать английские слова и термины.

Верно: Используются только английские слова.

.login-button{}

.container{}

.footer{}

Неверно: Транслит с русского языка.

.knopka-vxoda{}

.kontainer{}

.podval{}

* Д20. При взаимодействии с элементами (наведение, нажатие) ни сам элемент, ни окружающие его блоки не меняют своего положения (если иное не прописано в ТЗ или slyleguide.psd).
* Д21. Проведена базовая оптимизация: минифицированы стили и скрипты (при наличии), использован растровый спрайт. Если в макетах были иконки в растре, то из них нужно сделать растровый спрайт.

***Указания к проверке***

Критерий следует засчитывать только в случае, если выполнены все требования. Все иконки, которые представлены в макете сделаны с помощью иконочного шрифта или с помощью спрайта, а файлы js и css минифицированы.

Д24 Страницы должны ссылаться друг на друга из главного меню.

Д25 Нет глобальных стилей тегов.

Д26 У интерактивных элементов при нажатии или фокусе с клавиатуры есть активное состояние

**Выполнение ТЗ**

Общие технические требования

● 1.1. Стандарты вёрстки: HTML5, CSS3, прогрессивное улучшение.

● 1.2. Сетка: определена в макете.

● 1.3. Адаптивность вёрстки: нет.

● 1.4. Используемые фреймворки: нет.

● 1.5. Кроссбраузерность: IE11+, Chrome, Firefox, Opera, Safari.

● 1.6. Типографика: частично определена в макете (прочее — на усмотрение

разработчика).

● 1.7. Используемые шрифты: Gilroy, Open Sans (есть в папке с макетом и на

WebFont.ru).

● 1.8. С макетом предоставлен styleguide.psd, содержащий прорисовку состояний элементов интерфейса. При любых расхождениях с макетами он должен иметь наивысший приоритет.

2. Пояснения для учащихся

● 2.1. В макетах есть скрытые слои с всплывающими окнами. Такие слои в блоке слоёв фотошопа выделены синим цветом.

● 2.2. Макеты верстаются постепенно, не нужно сразу выполнять все требования.

● 2.3. Ниже в разделе «Обязательные требования» описано поведение блоков, которое должно быть реализовано для получения допуска. Требования из раздела

«Дополнительные требования» можно реализовать по желанию для выполнения дополнительных критериев.

3. Обязательные требования

Все макеты:

● 3.1. Контентная область центрируется и не может быть уже макетной ширины.

● 3.2. Авторизованному и неавторизованному посетителю показывается разный вид блока авторизации (смотрите device-catalog.psd и device-index.psd).

● 3.3. В блоке авторизованного посетителя имя и иконка пользователя являются ссылкой на профайл.

● 3.4. Логотип Академии в подвале ведёт на лендинг интенсива «Профессиональный HTML и CSS, уровень 1» .

device-index.psd:

● 3.5. Блоки с названиями разделов («виртуальная реальность», «моноподы для селфи», ...) являются ссылками на разделы каталога.

● 3.6. Промо-слайдер: слайдер. Вёрстка всех слайдов обязательна. Оживление слайдера необязательно, принцип оживления описан в 4 разделе.

● 3.7. Блок «Сервисы» («доставка», «гарантии», «кредит»): слайдер. Вёрстка всех слайдов обязательна. Оживление слайдера необязательно, принцип оживления описан в 4 разделе.

● 3.8. Блок карты — достаточная реализация — обычное изображение, клик по ней приводит к переходу на сервис карт.

● 3.9. Вёрстка модального окна обязательна (смотрите папку слоёв «write us»). device-catalog.psd:

● 3.10. Логотип — это ссылка на главную страницу. (в футере добавить)

● 3.11. Фильтр: верстать с помощью формы, кнопка «Показать» отвечает за отправку формы.

● 3.12. Блок «Стоимость» — при наведении на любой из маркеров появляется указатель cursor: pointer, делать маркеры подвижными не обязательно, цена меняться не должна.

● 3.13. При наведении на карточку товара появляются кнопки «В корзину» и

«Добавить к сравнению» (смотрите styleguide.psd). Ссылкой на отдельную страницу товара является только название товара (отдельную страницу верстать не нужно).

● 3.14. Количество товаров в правом блоке может быть любым, добавление товаров не должно ломать страницу.

● 3.15. У любого товара может быть метка «new».

4. Дополнительные требования

device-index.psd:

● 4.1. Промо-слайдер: оживление слайдера. Смена слайдов в слайдере должна происходить мгновенно, без промежуточных состояний и анимации (некоторые слайды скрыты - смотрите слои выделенные красным цветом).

● 4.2. Блок «Сервисы» («доставка», «гарантии», «кредит»): оживление слайдера. Слайдер с табами работает аналогично промо-слайдеру: по клику на таб меняется слайд мгновенно.

● 4.3. Блок карты — реализация по желанию — часть интерактивной карты, клик на неё приводит к появлению полного размера интерактивной карты (смотрите папку слоёв «map hover»), окно позиционируется относительно вьюпорта, а не страницы.

● 4.4. При клике по кнопке «Напишите нам?..» возникает модальное окно (смотрите папку слоёв «write us»), окно позиционируется относительно вьюпорта, а не страницы.

device-catalog.psd:

● 4.5. Фильтр: по нажатию кнопки «Показать» осуществляется отправка формы.