

# Введение

Итоговая оценка разбита на две составляющие: базовая и дополнительная.

Базовая оценка предполагает перепроверку критериев допуска. При условии, что они выполнены, выставляется базовая оценка и наставник переходит к дополнительной оценке.

Дополнительная оценка выставляется по дополнительным критериям. Дополнительные критерии оценивают проект с точки зрения шлифовки качества и оптимизации, и выстроены по принципу перфекционизма. Мы не требуем сделать проект идеально, но набрать максимальное количество баллов возможно.

Для подачи проекта на защиту нужно сверстать мобильное, планшетное и десктопное состояния трёх страниц проекта.

## Базовые критерии

### Разметка

- **B1. Выполнена HTML-разметка всех страниц проекта и всех элементов на этих страницах.**

Этот 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.

- **B2. Грубые ошибки в разметке отсутствуют.**

Грубые ошибки:

- Ссылки сделаны не тегом `<a>`, а другими тегами;
- 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>`.

Поэтому в ссылку можно включать целые секции, если `<a>` вложен в потоковый блок:

Верно:

```
<aside>
  <h2>Ограниченные предложения</h2>
  <a href="/buy">
    <section>
      <figure>
        
        <figcaption>
          Набор для путешествий «Baxter of California»
        </figcaption>
      </figure>
      <p>Всего за 100 рублей!</p>
    </section>
  </a>
</aside>
```

Неверно:

```
<span>
  <a href="/buy">
    <section>
      <h2>Акция!</h2>
      <p>Всего за 100 рублей!</p>
    </section>
  </a>
</span>
```

- Б3. Документ проходит проверку на валидность <https://validator.w3.org/nu/>.
- Б4. В разметке отсутствует дублирование кода для одного и того же элемента, с помощью которых элемент отображается в разных местах страницы на разных версиях: мобильной, десктопной, планшетной. Этот критерий не касается элементов, которые скрываются или показываются в разных версиях.

Неверно:

```
<div class="block block--for-mobile">
  <h2 class="block__title">Заголовок</h2>
```

```
<p class="block__text">Произвольный текст</p>
</div>

<div class="block block--for-desktop">
  <h2 class="block__title">Заголовок</h2>
  <p class="block__text">Произвольный текст</p>
</div>
```

- **Б5. 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.

*Неверно:*

```
<div class="block__element1">
  <div class="block__element1__element2"></div>
</div>
```

*Верно:*

```
<div class="block__element1">
  <div class="block__element2"></div>
</div>
```

- Создание модификатора для модификатора.

*Неверно:*

```
<div class="block block--mod1--mod2"></div>
```

*Верно:*

```
<div class="block block--mod1 block--mod2"></div>
```

- 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).

*Неверно:*

```
<div class="block--mod"></div>
```

*Верно:*

```
<div class="block block--mod"></div>
```

- **Б32. Названия полей форм привязаны к своим полям с помощью `<label>`.**

Для улучшения взаимодействия пользователя с элементами форм должна быть реализована активация поля при нажатии 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="spb">
<label for="spb">Санкт-Петербург</label>
```

*Верно:* элемент формы `radio` и подпись обернуты в `<label>`

```
<label>
  <input type="radio"> Санкт-Петербург
</label>
```

*Неверно:* описание никак не связано с элементом формы.

```
<input type="radio" id="spb"> Санкт-Петербург
```

## Стилизация

- **Б6. 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.

*Верно:* основная сетка сделана с использованием `flex` элементов.

```
<div class="features">
  <div class="features-item">
    ...
  </div>
  <div class="features-item">
    ...
  </div>
  <div class="features-item">
    ...
  </div>
</div>

.features {
  display: flex;
  justify-content: space-between;
}
.features-item {
  width: 300px;
```

```
}
```

*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 абсолютного позиционирования.

```
<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 {
  position: absolute;
  top: 0;

  width: 300px;
}
.features-1 {
  left: 0;
}
.features-item {
  left: 300px;
}
.features-item {
  right: 0;
}
```

- **Б7. В CSS отсутствует `!important`.**

Допускается использование `!important` при обосновании его необходимости в комментарии.

- **Б8. Подключены правильные шрифты, их размеры, цвет и толщина равны соответствующим параметрам в макетах и техническом задании.**

Необходимо, чтобы в личном проекте использовались именно те шрифты, которые указаны в макете. При этом их размеры, цвета, и жирность должны быть равны соответствующим параметрам в макетах и техническом задании.

- **Б9. Нестандартные шрифты подключены локально. Формат шрифтов должен быть `woff` и `woff2`.**

Все нестандартные шрифты должны быть подключены в CSS из папки `fonts` с помощью правила `@font-face`. Подключаемые шрифты должны быть в форматах `woff` и `woff2`.

Подключение шрифтов с помощью внешних ресурсов, таких как Google Fonts считается ошибкой.

Перечисление форматов шрифтов должно начинаться с более современного.

- **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](#).

**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;  
}
```

- **Б11. При наполнении контентом (как в макете) элементы каждой версии страницы (мобильной, планшетной и десктопной) соответствуют макету.**

Верстка проверяется на соответствие макету только на указанных выюпортах:

- **Пинк:**
  - **320px** — мобильная версия,
  - **660px** — планшетная версия,
  - **960px** — десктопная версия.
- **Седона:**
  - **320px** — мобильная версия,
  - **768px** — планшетная версия,
  - **1200px** — десктопная версия.
- **Мишка:**
  - **320px** — мобильная версия,
  - **768px** — планшетная версия,
  - **1150px** — десктопная версия.

Размеры выюпортов определены по крайним направляющим в макетах.

**Важно:** в ширину выюпорта не входит полоса прокрутки.

При проверке допускаются:

- Вертикальная погрешность не более 10 пикселей, горизонтальная погрешность не более 5 пикселей;
- Различия в отображении шрифтов, связанные со сглаживанием на различных платформах.

## CSS-препроцессор

- **Б12. Использован CSS-препроцессор.**

Писать CSS-код можно только с помощью CSS-препроцессора. Можно использовать: Less или Sass.

Стилевой код, написанный без препроцессора, является ошибкой.

- **Б13. 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.

Вариант *верного* разбиения файла стилей:

```
less/  
  normalize.less  
  grid.less  
  blocks/  
    block1.less  
    block2.less  
    ...  
  style.less
```

Вариант *неверного* разбиения файла стилей:

```
less/  
  normalize.less  
  global.less  
  index-page.less  
  header.less  
  photo-page.less  
  style.less
```



Вариант *неверного* содержания файла `header.less` со стилями блока. Внутри файла с блоком `header` находятся стили другого блока — `logo`:

```
.header {  
  padding: 20px;  
}  
  
.header__top {  
  background-color: #cccccc;  
}  
  
.logo {  
  width: 200px;  
}
```

## Адаптивность

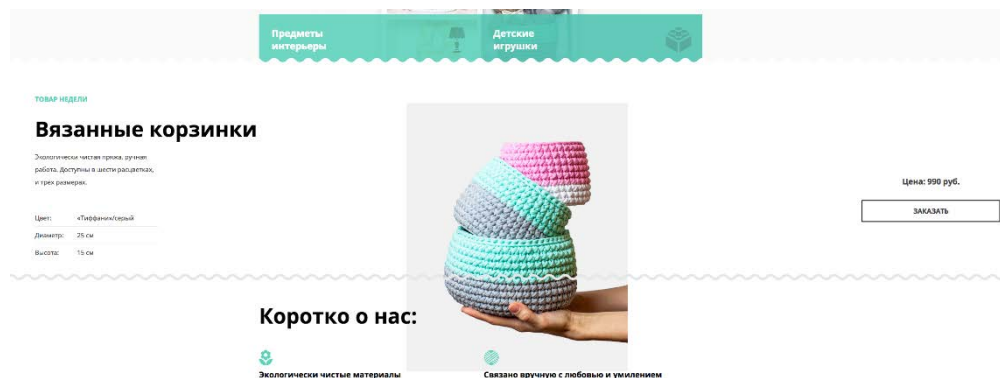
- **Б14.** Выполнена вёрстка трёх состояний каждой страницы: мобильной, планшетной и десктопной.

Сетка в промежутках между основными точками может быть фиксированной или резиновой.

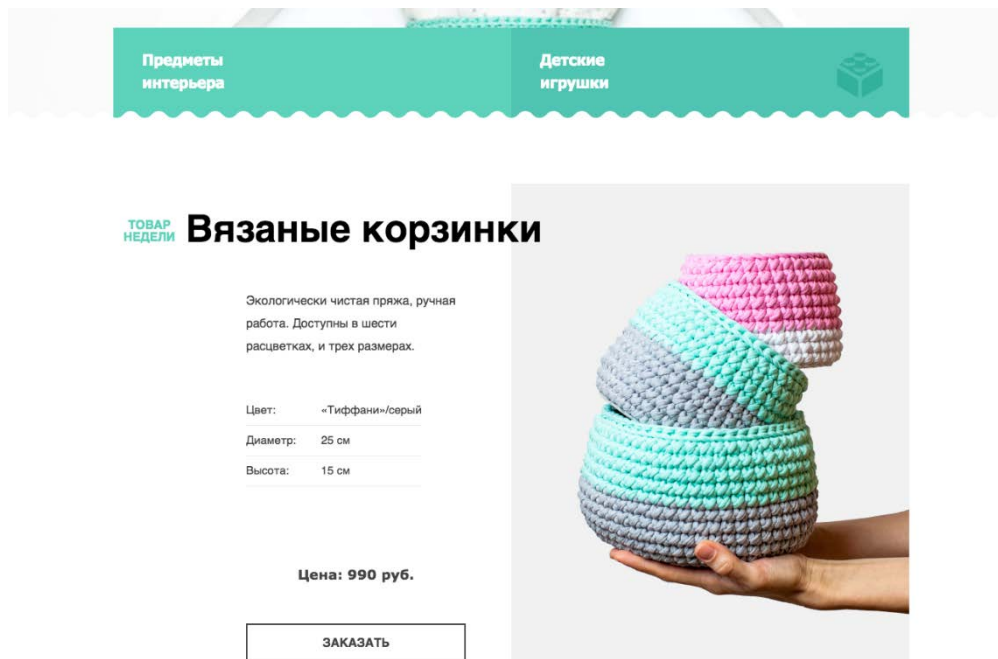
**Важно:** вне указанных точек вёрстка не должна «разваливаться»:

- все элементы продолжают располагаться по сетке, заданной в макете;
- блоки не выпадают из родителей и из общего центрирующего блока;

*Неверно:*



*Верно:*



- **Б15. В разметке есть правильный выюпорт тег.**

Адаптивность должна работать на планшетах и мобильных устройствах. То есть не только при изменении ширины окна браузера.

- **Б16. Для микросеток использованы флексбоксы.**

Следующие компоненты должны быть реализованы с помощью флексбокса:

- **Пинк:**
  - разделы формы, поля и подписи формы на странице «Конкурс на безбашенное путешествие»;
  - раскладка фотографий на странице «Фотографии пользователей»;
  - блок описания фотографии (автор, описание, лайки) на странице «Фотографии пользователей»;
  - блок постинга фотографии на странице «Фотографии пользователей».
- **Седона:**
  - разделы формы, поля и подписи формы на странице отзыва;
  - раскладка фотографий на странице «Фото и видео»;
  - блок описания фотографии (название, автор, лайки) на странице «Фото и видео»;
  - видеоплеер на странице «Фото и видео».
- **Мишка:**

- шапка сайта (лого, основное меню, поиск, корзина, новые поступления, распродажа, доставка);
- разделы формы, поля и подписи формы на странице «Вязание на заказ»;
- раскладка товаров и блок описания товара (название, описание, цена, корзина) на странице «Каталог товаров».

## • Б17. Выполнена ретинизация растровой графики.

Графика должна быть подготовлена как минимум для двух вариантов экранов: с обычной и двойной плотностью пикселей. Обязательны для ретинизации следующие компоненты:

### ◦ Пинк:

- задний фон в промоблоке под главным меню на всех страницах;
- рука с телефоном в промоблоке под главным меню на главной странице;
- изображения телефонов в блоке преимуществ на главной странице;
- контентные изображения на странице «Фотографии пользователей».

### ◦ Седона:

- задний фон в промоблоке под главным меню на всех страницах;
- фоны (или изображения) в промоблоках «Настоящий городок» и «Там есть Мост Дьявола» на главной странице;
- контентные изображения на странице «Фото и видео».

### ◦ Мишка:

- задний фон в промоблоке на главной странице;
- изображение товара недели на главной странице;
- задний фон в блоке отзывов на главной странице;
- контентные изображения товаров в каталоге.

## • Б19. Логотип должен адаптироваться.

Логотип должен адаптироваться к мобильному, планшетному и десктопному состоянию проекта.

Логотип можно сделать с помощью тега `<picture>` или с помощью встроенного SVG-изображения.

## • Б20. Выполнено кадрирование контентных изображений.

С помощью тега `<picture>` выполнено кадрирование контентных изображений:

- **Пинк:**
  - на странице «Фотографии пользователей».
- **Седона:**
  - на странице «Фото и видео».
- **Мишка:**
  - на странице «Каталог товаров».

Кадрирование в данном случае подразумевает загрузку разных фотографий на разном разрешении экрана.

*Например:*

```
<picture>
  <source media="(min-width: 1200px)" srcset="img/photo-1-
desktop.jpg">
  <source media="(min-width: 768px)" srcset="img/photo-1-
tablet.jpg">
  
</picture>
```

## Графика

### • Б18. Использована векторная графика.

Обязательно должны быть использованы следующие векторные изображения:

- **Пинк:**
  - логотип Пинк;
  - логотип Академии;
  - иконки приложений на главной странице.
- **Седона:**
  - логотип Седоны;
  - логотип Академии;
  - тексты в промоблоке под главным меню;
  - иконки «жильё», «еда» и «сувениры» на главной странице.
- **Мишка:**
  - логотип Мишки;
  - логотип Академии;
  - иконки в блоке «Коротко о нас» на главной странице.

## Оптимизация

- **Б21. «Нежная» ретинизация.**

При ретинизации изображений учитывается плотность пикселей устройства пользователя и возвращаются разные изображения. Для контентных изображений должен быть использован `srcset`, а для фонов — медиавыражения.

Например, ретинизация контентного изображения:

```

```

Ретинизация фонового изображения:

```
.page-header__logo {
  background-color: #000000;
  background-image: url("img/bg-header@1x.jpg");
}

@media
  (min-resolution: 144dpi),
  (min-resolution: 1.5dppx) {
  .page-header__logo {
    background-image: url("img/bg-header@2x.jpg");
    background-size: 1338px auto;
  }
}
```

## Сборка проекта

- **Б22.** Процесс сборки личного проекта настроен с помощью Grunt или Gulp.
- **Б23.** Все зависимости проекта должны быть указаны в файле `package.json`. Команда `npm i` должна установить всё необходимое для того, чтобы сборка проекта работала.
- **Б24.** Сборка проекта должна запускаться командой `npm run build`.

При сборке не должно возникать никаких ошибок.

- **Б25. В результате сборки должна получаться папка `build` со всеми необходимыми файлами.**

Особенности сборки:

- папка `build` должна находиться в корне проекта;
  - папка `build` должна автоматически создаваться в момент сборки;
  - в папке `build` должна быть папка `css` с скомпилированными и минифицированными стилевыми файлами;
  - в папке `build` должна быть папка `fonts` в которой лежат только шрифты в формате `woff` и `woff2`;
  - в папке `build` должна быть папка `img` в которой лежат только оптимизированные изображения и собранные спрайты;
  - в корне папки `build` должны лежать только HTML-файлы проекта.
- **Б26. Папка `build` со всем её содержимым не должна попадать в Гитхаб.**

Это значит, что папка `build` должна находиться в файле `.gitignore`.

## Разное

- **Б27. Вёрстка идентично отображается в последних версиях браузеров Chrome, Opera, Firefox, Safari, Edge, а также в Internet Explorer 11.**

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.

- **Б28. 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;

- о если размер отступа в два пробела, то таким он должен быть везде;
- о если для отступов используются табы или пробелы, то для всех отступов должны быть либо табы, либо пробелы;
- о названия классов должны быть оформлены единообразно;
- о 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.

- **Б29. Отсутствует транслит в названиях классов, атрибутах, переменных CSS-препроцессора, названиях примесей и так далее.**

Для названий необходимо использовать английские слова и термины.

*Верно:* используются только английские слова.

```
.login-button {}
.container {}
.footer {}
```

*Неверно:* транслит с русского языка.

```
.knopka-vxoda {}
.kontainer {}
.podval {}
```

- **Б30. Мобильное меню должно быть работоспособным на мобильной версии при отключённом JavaScript.**

При реализации без JS:

1. меню в мобильной версии должно быть открыто по умолчанию и находиться в потоке (то есть не закрывать остальной контент);
2. на мобильной и планшетной версии крестик должен быть скрыт;
3. в комментариях HTML-файлов должны быть указаны классы (или класс), которые переключают состояния меню и внешний вид кнопки.

В HTML- и CSS-коде должны быть предусмотрены элементы и правила для трёх возможных состояний:

1. открытое меню в потоке при отключённом JavaScript;
2. открытое меню вне потока при включённом JavaScript;
3. закрытое меню.

Во всех трёх личных макетах достаточно двух состояний, открытое и закрытое меню, так как открытое меню расположено в потоке.

Скрипт открытия и закрытия меню писать необязательно. Но если он написан, то работать должен так:

1. при инициализации скрипта меню должно убираться из потока (если в макете открытое меню не в потоке) и закрываться;
2. при клике на гамбургер, меню открывается. Появляется кнопка-закрытие, убирается кнопка-гамбургер;
3. при клике на закрытие, меню закрывается, убирается кнопка-закрытие, появляется гамбургер.

## • Б31. Проект соответствует техническому заданию

Техническое задание каждого проекта можно посмотреть на [странице ТЗ](#).

В некоторых пунктах есть необязательные реализации, которые выполняются по желанию.

Обратите внимание, что:

- о должна быть сделана вёрстка всех модальных окон, но реализовывать открытие/закрытие необязательно;
- о должна быть сделана разметка всех слайдов, но оживлять их переключение в слайдере необязательно.

При конфликте технического задания и критериев, считать техническое задание более приоритетным.

# Дополнительные критерии

## Разметка

- Д1. У всех векторных изображений размер прописан в теге `<img>`, у встроенных SVG-изображений размер прописан в теге `<svg>`.

В HTML-коде должны быть указаны ширина и высота **векторных** картинок, так как векторные картинки могут растянуться на всю ширину родителя.

**Важно:** в атрибутах `width` и `height` нельзя использовать `px`, так как по спецификации размеры в пикселях задаются без единиц измерения. Проценты использовать можно, если картинка должна тянуться.



Для адаптивных изображений, у которых есть версии с разными размерами и пропорциями (например, заданы через тег `sources` в элементе `picture`), размеры в `img` в HTML можно не задавать.

*Верно:* картинке заданы размеры.

```
<div class="page-header__logo">
  
</div>
```

*Неверно:* картинке не заданы размеры.

```
<div class="page-header__logo">
  
</div>
```

*Неверно:* картинке заданы размеры с использованием `px`.

```
<div class="page-header__logo">
  
</div>
```

*Верно:* тегу `svg` заданы размеры.

```
<svg width="200" height="200">
  <rect x="0" y="0" width="200" height="200" fill="#FAFAA2"
stroke="#000"/>
</svg>
```

*Верно:* адаптивному изображению может быть не задан размер в HTML.

```
<picture>
  <source media="(min-width: 1150px)" srcset="img/logo-
desktop.svg">
  <source media="(min-width: 768px)" srcset="img/logo-
tablet.svg">
  
</picture>
```

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

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

## Стилизация

- Д4. 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.

```
.main-nav {}  
.page-footer {}
```

*Неверно:* стилизация элементов строится через идентификаторы.

```
#main-nav {}  
#page-footer {}
```

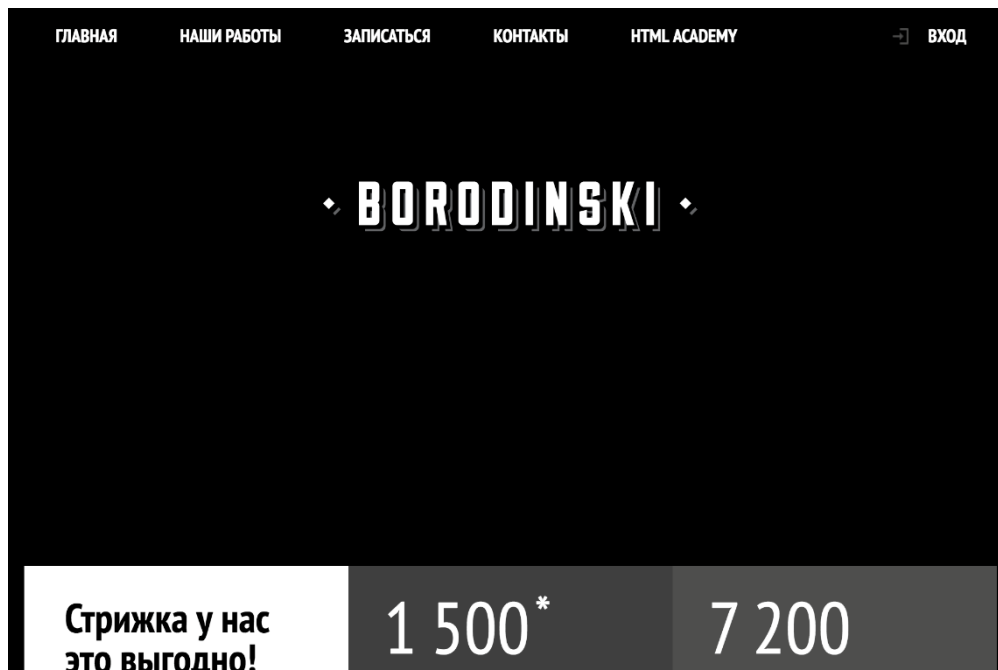
- Д6. Для блока, у которого есть фоновое изображение, прописан фоновый цвет, который соответствует преобладающему цвету изображения (пока изображение не загружено, страница выглядит похоже на макет).

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

*Верно:* указан цвет фона перед основным изображением.

```
body {  
  background: #000000 url("../img/background-body.jpg") no-repeat  
  center top;  
}
```

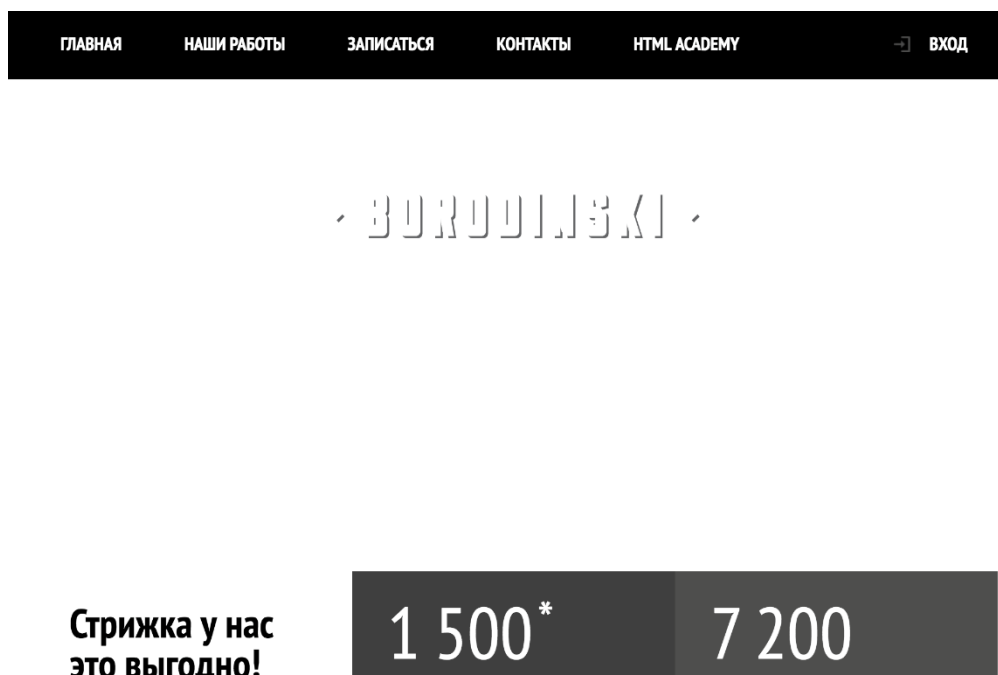
*Результат:* картинка не загрузилась, но при этом цвет фона остался тёмным.



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.



- Д7. Все состояния элементов (смотрите [styleguide.psd](#)) прописаны в стилевом файле.

В соответствии с файлом [styleguide.psd](#), 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 радиокнопок прописаны состояния `disabled`.

- **Д16. Нет глобальных стилей тегов.**

Для задания стилей используются только селекторы по классам или каскад.

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` (смотри пример в колонке «Верно»)

*Верно:*

```
body { margin: 0 }
img { max-width: 100% }
a { text-decoration: none }

.feedback { color: black }
.feedback ul { list-style: none }
.feedback p { margin-bottom: 0 }

html { box-sizing: border-box; }
*, *:before, *:after { box-sizing: inherit; }
```

*Неверно:*

```
ul { list-style: none }
p { margin: 0 }
div { font-size: 80% }
* { box-sizing: border-box }
```

## CSS-препроцессор

- **Д8. 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 `opacity()` и `transparentize()` в Sass.

- **Д9. Примеси не используются для генерации правил с вендорными префиксами.**

Этот критерий запрещает использовать миксины для генерации правил с вендорными префиксами. *Например:*

```
.box-sizing {  
  -webkit-box-sizing: border-box;  
  -moz-box-sizing: border-box;  
  box-sizing: border-box;  
}  
  
.class {  
  .box-sizing();  
}
```

Использование таких миксинов нецелесообразно, так как в проекте используется Автопрефиксер, который расставляет вендорные префиксы самостоятельно.

- **Д10. Нет вложенности больше двух уровней.**

Большая глубина вложенности усложняет читабельность кода и его поддержку. Хорошим подходом считается использовать вложенность не больше двух уровней.

Псевдоэлементы, псевдоселекторы, селекторы с `+` и `~` не увеличивают вложенность.

*Плохо:* вложенность больше двух уровней.

```
.block {  
  width: 400px;  
  
  .title {  
    font-size: 26px;  
  
    span {  
      font-weight: 600;  
    }  
  }  
}
```

*Хорошо:* вложенность не увеличена.

```
.checkbox__value:checked ~ .checkbox__state::before {  
  display: block;  
}
```

- **Д11. Верное использование `&` в стилевых файлах.**

Использование `&` в стилевых файлах (не примесях) допускается только:

- Для добавления псевдоэлементов и псевдоклассов в контекст селектора.  
*Например:*

```
.block {
  &::before {
    content: '';
  }

  &:hover {
    ...
  }
}
```

- В местах разделителя модификатора (перед разделителем). *Например:*

```
.block {
  ...

  &--mod {
    ...
  }
}

.block__element {
  ...

  &--mod {
    ...
  }
}
```

- **Д12. Не используются расширения (`extend`).**

Этот критерий запрещает использовать функцию `&:extend(.class);` для Less и функцию `@extend .class;` для Sass.

## Тестирование

- **Д13. Вёрстка проходит тест на переполнение контентом.**
  - Не ломается при добавлении в элементы большего количества текста.  
*Верно:* текст растягивает блок.



### БЫСТРО

Мы делаем свою работу быстро!  
Два часа пролетят незаметно  
и вы — счастливый обладатель  
стильной стрижки-минутки. Мы  
делаем свою работу быстро! Два  
часа пролетят незаметно и вы —  
счастливый обладатель стильной  
стрижки-минутки. Мы делаем  
свою работу быстро! Два часа  
пролетят незаметно и вы —  
счастливый обладатель стильной  
стрижки-минутки.



### КРУТО

Забудьте, как вы стриглись  
раньше. Мы сделаем из вас  
звезду футбола или кино.  
Во всяком случае внешне.



### ДОРОГО

Наши мастера — профессионалы  
своего дела и не могут стоять  
дёшево. К тому же, разве цена  
не даёт определённый статус?

## НОВОСТИ И АКЦИИ

[ПОКАЗАТЬ ВСЕ](#)

*Неверно:* текст выпадает за пределы блока с тёмным фоном.



### БЫСТРО

Мы делаем свою работу быстро!  
Два часа пролетят незаметно  
и вы — счастливый обладатель  
стильной стрижки-минутки. Мы  
делаем свою работу быстро! Два  
часа пролетят незаметно и вы —  
счастливый обладатель стильной  
стрижки-минутки. Мы делаем  
свою работу быстро! Два часа  
пролетят незаметно и вы —



### КРУТО

Забудьте, как вы стриглись  
раньше. Мы сделаем из вас  
звезду футбола или кино.  
Во всяком случае внешне.



### ДОРОГО

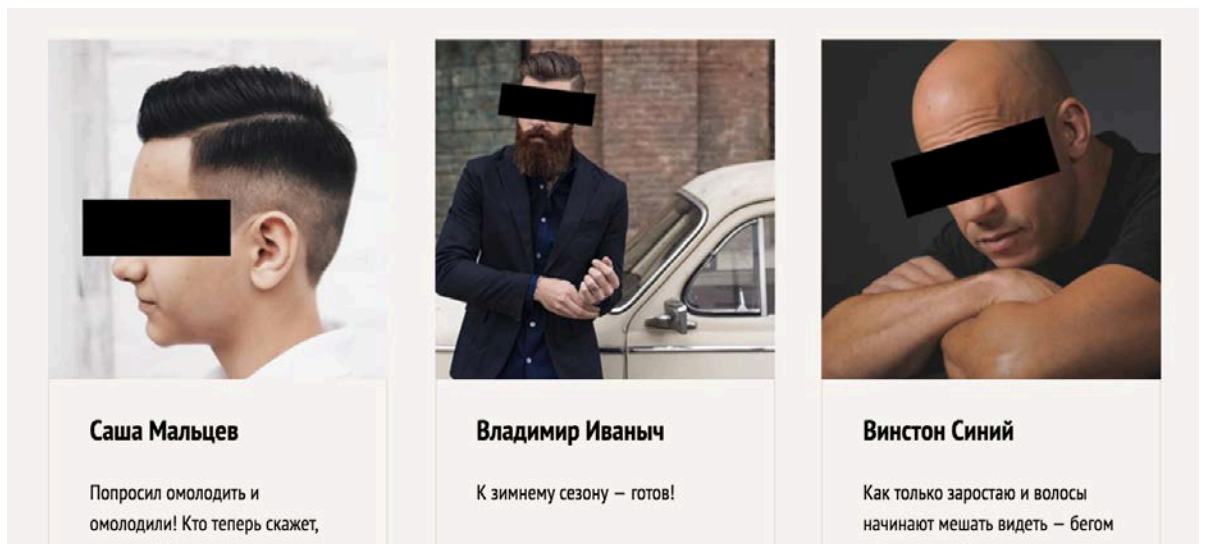
Наши мастера — профессионалы  
своего дела и не могут стоять  
дёшево. К тому же, разве цена  
не даёт определённый статус?

## НОВОСТИ И АКЦИИ

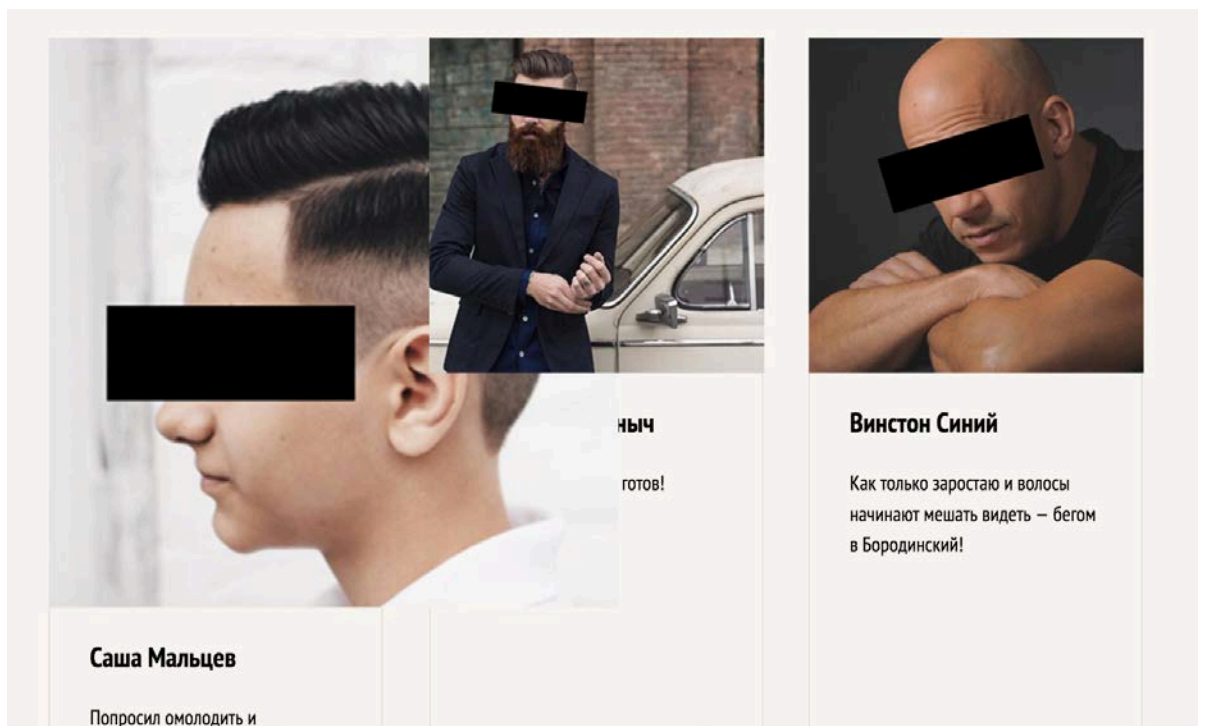
[ПОКАЗАТЬ ВСЕ](#)

- Не ломается при использовании картинок с неподходящими размерами.

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



*Неверно:* добавление большей картинки сломало сетку.



- Не ломается при изменении количества потоковых блоков. Текст не выпадает из блоков, нижерасположенные блоки не скрываются, смещение блоков в потоке сохраняет логику потока (не приводит к нарушению сетки).

*Верно:* переполнение верхнего блока растягивает его и двигает нижний.





*Неверно:* переполнение верхнего блока приводит к тому, что блоки скрываются под следующими элементами.



- Д14. Критическая функциональность сайта работоспособна без JavaScript (использовано прогрессивное улучшение).

*Например:*

- Все формы являются работоспособными без JavaScript.
- Элементы, вызывающие появление попапов, являются ссылками, ведущими на отдельные страницы.

Для выполнения этого критерия достаточно указать адрес на страницу, на которую будет происходить переход в случае неработоспособности JavaScript. При этом верстать саму страницу необязательно. *Пример:*

```
<a class="btn btn-open-form" href="form.html">Открыть форму</a>
```

- Интерактивная карта без JavaScript показывает статичную картинку с картой.
- Мобильное меню по умолчанию открыто.

## Доступность

- **Д17. У интерактивных элементов при нажатии или фокусе с клавиатуры есть активное состояние.**

Активное состояние интерактивных элементов при нажатии или фокусе с клавиатуры должно оставаться либо встроенным браузерным, либо быть равноценно переназначено. В таком случае по интерактивным элементам сайта можно передвигаться с клавиатуры клавишей **Tab** и видеть каждый текущий элемент в активном состоянии.

- **Д18. 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.

- **Пинк** — социальные кнопки, логотип Академии в подвале, лайки, элементы управления фильтром;
- **Седона** — социальные кнопки, логотип Академии в подвале, лайки, элементы управления видеоплеером;
- **Мишка** — кнопка поиска в шапке, социальные кнопки, логотип Академии в подвале, кнопка добавления товара в корзину.

## Форматирование и внешний вид

- **Д19. Код соответствует правилам в EditorConfig**

При выполнении автоматической проверки и консольной команды `npm run editorconfig` в корневой папке проекта не возникает ошибок.

- **Д20. Код соответствует правилам в Stylelint.**

При выполнении автоматической проверки и консольной команды `npm run stylelint` в корневой папке проекта не возникает ошибок.

## Оптимизация

- **Д21. Используются изображения в формате webp.**

Для браузеров, поддерживающих формат `webp`, контентные изображения подключаются в этом формате. Для браузеров, не поддерживающих `webp`, изображения подключаются в формате `jpeg` или `png`.

- Д22. Использован векторный спрайт.

Векторные изображения на странице объединены в спрайт.

## Разное

- Д15. При взаимодействии с элементами (наведение, нажатие) ни сам элемент, ни окружающие его блоки не меняют своего положения (если иное не прописано в техническом задании или [styleguide.psd](#)).