

Требования к верстке

В результате работы должна всегда быть адаптивная, кроссбраузерная, валидная верстка, с использованием html5, css3 и jquery, которая корректно выглядит во всех современных браузерах:

- Opera
- Mozilla Firefox
- Google Chrome
- Internet Explorer 10.0+

Стиль написания кода

Не оставляем пустые блоки стилей

НЕ верно

```
.a {  
    color: tomato;  
}  
.b{}
```



Верно

```
.a {  
    color: tomato;  
}
```



В конце каждого CSS свойства ставится точка с запятой ;

НЕ верно

```
.a {  
    color: tomato  
}
```



Верно

```
.a {  
    color: tomato;  
}
```



Все селекторы пишутся в нижнем регистре

НЕ верно

```
LI > a {  
  color: tomato;  
}
```



Верно

```
li > a {  
  color: tomato;  
}
```



ПЕРЕД двоеточием в CSS свойствах НЕ СТАВИТСЯ пробел

НЕ верно

```
.a {  
  color :tomato;  
  top :0;  
}
```



Верно

```
.a {  
  color: tomato;  
  top: 0;  
}
```



ПОСЛЕ двоеточия в CSS свойствах СТАВИТСЯ пробел

НЕ верно

```
.a {  
  color:tomato;  
  top:0;  
}
```



Верно

```
.a {  
  color: tomato;  
  top: 0;  
}
```



В дочерних селекторах ПЕРЕД и ПОСЛЕ знака больше > СТАВИТСЯ пробел

НЕ верно

```
.a>span {  
  color: tomato;  
}
```

Верно

```
.a > span {  
  color: tomato;  
}
```

Каждое CSS свойство пишется на новой строке

НЕ верно

```
.a {  
  color: tomato; top: 0; height: 0;  
}
```

```
.a {color: tomato; top: 0; height: 0;}
```

Верно

```
.a {  
  color: tomato;  
  top: 0;  
  height: 0;  
}
```

ПЕРЕД открывающей фигурной скобкой { в блоке свойств СТАВИТСЯ пробел

НЕ верно

```
.a{  
  color: tomato;  
}
```

Верно

```
.a {  
  color: tomato;  
}
```

ПОСЛЕ открывающей фигурной скобкой { в блоке свойств СТАВИТСЯ перенос строки

НЕ верно

```
.a{color: tomato;
}
```



Верно

```
.a {
  color: tomato;
}
```



ПЕРЕД закрывающей фигурной скобкой `}` в блоке свойств СТАВИТСЯ перенос строки

НЕ верно

```
.a{
  color: tomato; }
```



Верно

```
.a {
  color: tomato;
}
```



КАЖДАЯ новая группа свойств разделяется ПУСТОЙ СТРОКОЙ

НЕ верно

```
.a {
  color: tomato;
}
.b {
  color: tomato;
}
```



Верно

```
.a {
  color: tomato;
}

.b {
  color: tomato;
}
```



При использовании перечисления селекторов ПЕРЕД запятой НЕ СТАВИТСЯ пробел, после ставится

НЕ верно

```
.a , .b {  
  color: tomato;  
}
```

Верно

```
.a, .b {  
  color: tomato;  
}
```

Вендорные префиксы не пишутся руками в SCSS, они подставляются автоматически сборщиком

НЕ верно

```
.a {  
  -webkit-border-radius: 3px;  
  -moz-border-radius: 3px;  
  border-radius: 3px;  
}
```

Верно

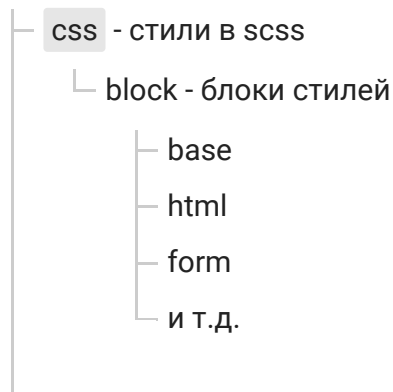
```
.a {  
  border-radius: 3px;  
}
```

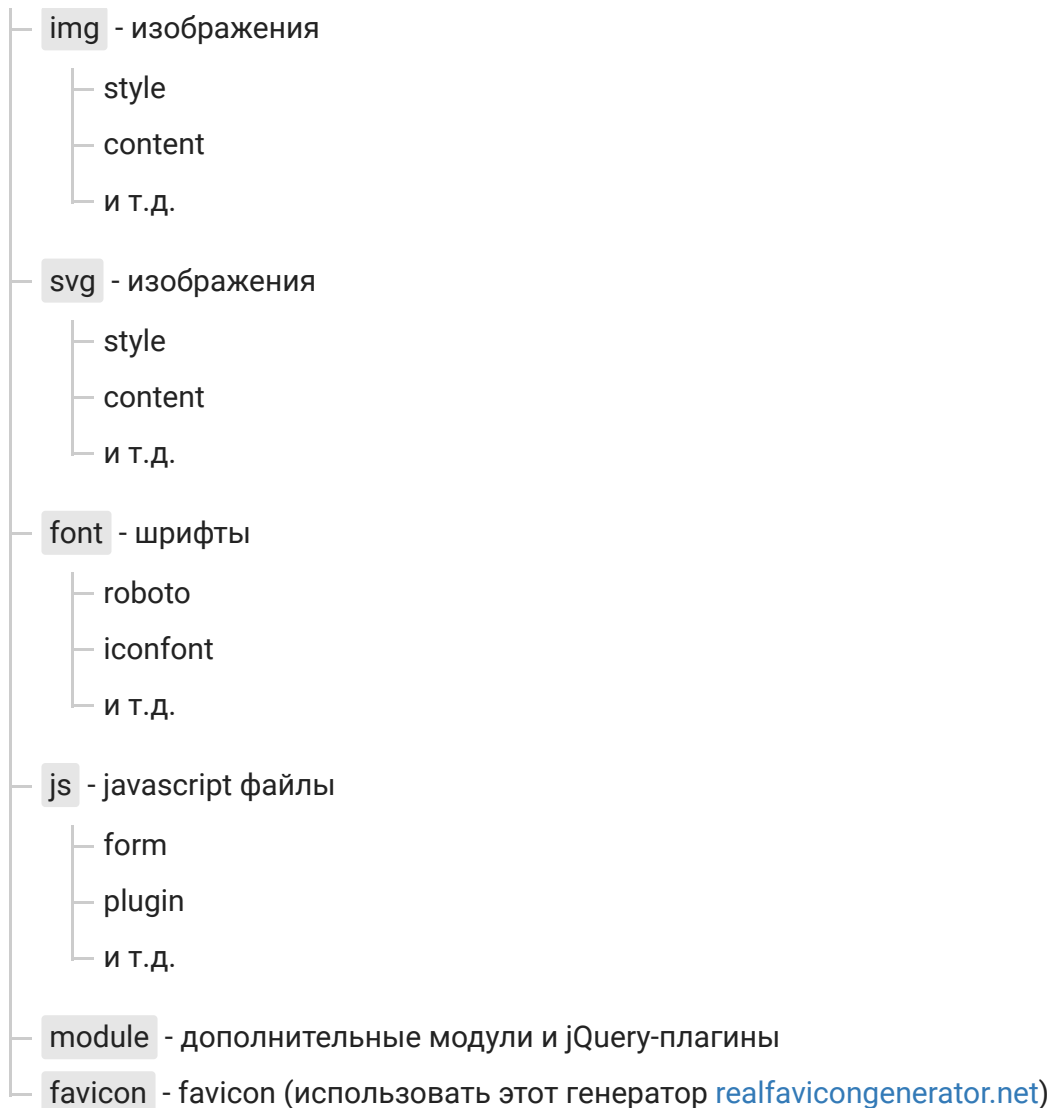
Структура проекта

Проект должен иметь следующую структуру:

`_assets` - скомпилированные исходники

`_source` - исходники





HTML

Обязательно соблюдение следующих требований к HTML-разметке

- Доктайп html5 https://www.w3schools.com/tags/tag_doctype.asp
- Семантические теги для блоков (`<header>`, `<footer>`, `<article>`, `<section>`, `<aside>`, `<nav>`) https://www.w3schools.com/html/html5_semantic_elements.asp
- Микроформаты <https://yandex.ru/support/webmaster/hcard/general.html?lang=ru>
- Валидность https://validator.w3.org/#validate_by_input
- НЕ используем комментарии `<!-- -->`
- НЕ используем теги без их семантической принадлежности:
 - Если ссылка не является ссылкой (`href=""`, `href="#"`, `href="javascript:void(0);"`) не используем тег `<a>`
 - Если список не выглядит как список, не используем теги `` (``) и ``
 - и т.д.

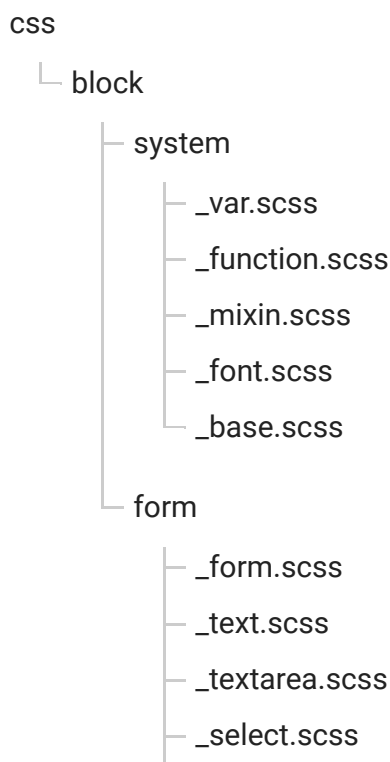
- Подключение стилей влияющих на отображение первого экрана сайта оставляем в `<head>`, остальные переносим перед `</body>`
- В `<head>` подключаются скрипты фреймворков (jQuery и др.) и влияющие на отображение первого экрана сайта, остальное переносим перед `</body>`
- НЕ используем CDN. Сайт должен корректно отображаться без наличия интернета
- Теги `<h1>` - `<h6>` используются ТОЛЬКО для выделения разделов и подразделов. Для визуального выделения текста используются классы.
- Тег `<h1>` обязательно должен присутствовать на странице и только один
- Для размещения текста на странице используем тег `<p>`, НЕ нужно размещать абзацы текста в `<div>`
- Для выделения жирным и курсивом в интерфейсных элементах используем классы и CSS свойства, для выделения в тексте используем теги `` и `` соответственно.
- Для разметки контента (сетки) используются строго теги `<div>`, использование других тегов недопустимо.

CSS

Общие требования

- Обязательно использование актуальной версии [normalize.css](https://n3u3.github.io/normalize.css/)
- Для верстки используем препроцессор SCSS (*.scss) <http://sass-lang.com/>
- Стили делятся на логические блоки и подключаются с сохранением единой области видимости

Например:



```
├── _checkbox.scss
├── _radio.scss
└── _button.scss
```

Общий файл стилей:

```
//system
@import 'block/system/var';
@import 'block/system/function';
@import 'block/system/mixin';
@import 'block/system/font';
@import 'block/system/base';

//form
@import 'block/form/form';
@import 'block/form/button';
@import 'block/form/text';
@import 'block/form/checkbox';
@import 'block/form/radio';
@import 'block/form/select';
@import 'block/form/textarea';
```

- НЕ используем краткое написание свойств background и font. Например, background: red изменяет не один, а сразу 10 стилей, включая background-color, background-size, background-repeat и другие.
- НЕ используем **!important** без крайней необходимости

Все что делается посредством использования !important можно сделать без него, потому использование оправдано только в крайних редких случаях.

- НЕ используем селекторы полностью:

- **Универсальный (*)**

Потому что есть наследование стилей.

Исключение:

```
* {
  box-sizing: border-box;
}
```

- **Идентификаторов (#id)**

Потому что:

1. Идентификаторы устарели
2. Использование идентификаторов делает невозможным дублирование кода
3. Использование идентификаторов делает код невалидными при невнимательности

4. Использование идентификаторов существенно повышает приоритет кода

◦ **Атрибутов** (input[type="text"])

Потому что запись:

```
input[type=text], select, textarea {  
  cursor: pointer;  
  display: block;  
  width: 100%;  
}
```

Не включает в себя:

- input[type="password"]
- input[type="tel"]
- input[type="number"]
- input[type="email"]
- input[type="date"]
- и т.д.

Для тега `<input>` атрибут тип на мобильной версии должен быть изменен динамически на корректный.

Правильно в данном случае использовать классы, например `.form__input`.

◦ **Тегов** (div)

Потому что определяется слишком много свойств, что в дальнейшем крайне сложно контролировать и переопределять.

Исключение:

- `<html>`, `<body>` - базовые свойства для наследования
- `<a>`, `<p>` - для всего сайта
- Основные теги контентного оформления ТОЛЬКО для блока контента:
 - Списки: `` `` ``
 - Ссылки: `<a>`
 - Текст: `<h1>` `<h2>` `<h3>` `<h4>` `<h5>` `<h6>` `<p>` `` `` `` `<quote>`
 - Изображения: ``
 - Таблицы: `<table>` `<tbody>` `<thead>` `<tfoot>` `<tr>` `<td>` `<th>`

- Используем селекторы контекстные (вложенные) и дочерние умеренно, МАКСИМУМ 2 уровня вложенности.

НЕ верно:

```
#mapModal {
  .modal-dialog {
    .modal-content {
      .mark-info {
        max-width: 655px;
        min-height: 240px;
        margin: 0 auto 0 auto;
        top: 30px;

        .mi-content {
          padding: 30px 15px 0 15px;
        }
      }
    }
  }
}
```

Верно:

```
.map-modal {
  .mark-info {
    max-width: 655px;
    min-height: 240px;
    margin: 0 auto;
    top: 30px;
  }

  .mi-content {
    padding: 30px 15px 0 15px;
  }
}
```

- Обязательно использовать селекторы классов (например: .menu), кроме исключений указанных выше
- Обязательно использовать BEM синтаксис именования классов
<https://ru.bem.info/methodology/naming-convention/>

CSS

```
<style>
  //block
  .menu {}

  //element
  .menu__item {}

  //modifier
  .menu__item_active {}
</style>
```

SCSS

```
<style>
  //block
  .menu {

    //element
    &__item {

      //modifier
      &_active {

      }

    }
  }
</style>
```

- Обязательно в свойстве font-family записывать запасные семейства шрифтов: Arial, sans-serif;
- Для повторяющихся значений, таких как семейства шрифтов, цветов и т.д. необходимо использовать переменные препроцессора.

Контент

- Стили для основных тегов оформления контента обязательно должны быть прописаны при помощи использования вложенных селекторов тегов для контейнера контента.

В системах администрирования нет возможности корректно размечать текст классами верстки.

Пример:

```
.content-block {
  ul {
    margin: 10px 0 20px 0;
  }

  li {
    margin-bottom: 6px;
  }
}
```

Основные теги контентного оформления:

- Ссылки: `<a>`
- Изображения: ``

- Списки: `` `` ``
 - Текст: `<h1>` `<h2>` `<h3>` `<h4>` `<h5>` `<h6>` `<p>` `` `` `` `<quote>`
 - Таблицы: `<table>` `<tbody>` `<thead>` `<tfoot>` `<tr>` `<td>` `<th>`
- ОБЯЗАТЕЛЬНО необходимо прилагать к результатам верстки отдельную страницу с примером всех основных элементов сайта. Даже если нет сводного макета со всеми функциональными элементами.

Страница должна включать в себя:

- Ссылки:
 - `<a>`
- Списки:
 - ``
 - ``
 - ``
- Заголовки:
 - `<h1>`
 - `<h2>`
 - `<h3>`
 - `<h4>`
 - `<h5>`
 - `<h6>`
- Текст:
 - `<p>`
 - ``
 - ``
 - ``
 - `<quote>`
- Изображения со всеми выравниваниями:
 - ``
- Таблицы:
 - `<table>`
 - `<tbody>`
 - `<thead>`
 - `<tfoot>`

- `<tr>`
- `<td>`
- `<th>`

- Формы:

- `<form>`
- `<input[type="text"]>`
- `<input[type="checkbox"]>`
- `<input[type="radio"]>`
- `<input[type="password"]>`
- `<textarea>`
- `<select>`
- `<input[type="submit"]>`
- `<button>`

Группировка

- Файлы стилей группируются по их логическому назначению, а не по названию блоков
- НЕ нужно группировать файлы стилей по названию страниц на сайте

Классы

- Для имен классов необходимо использовать правильные английские слова без артиклей в единственном числе.

Написание имен классов строго в нижнем регистре.

ЗАПРЕЩЕНО

- Танслит

НЕ правильно: `.verhnee-menu`

Правильно: `.menu-top`

- Множественное число

НЕ правильно: `.items`

Правильно: `.item-list`

- Артикли

НЕ правильно: `.member-of-the-team`

Правильно: `.team-member`

- Верхний регистр

НЕ правильно: `.headerTopMenu`

Правильно: `.header-top-menu`

- ЗАПРЕЩЕНО использовать имена классов отличающихся одной буквой

НЕ правильно

```
.basic-service  
.basic-services
```



Правильно

```
.basic-service  
.basic-service__item
```



```
.basic-service-item  
.basic-service-list
```



- ЗАПРЕЩЕНО использовать имена классов длинее трех слов **.header-top-menu-item-link**
- ДОПУСТИМЫЕ базовые слова для использования в классах (не обязательно ограничиваться ими)
 - **block** , **section** - блок
 - **item** - пункт, элемент
 - **link** - ссылка
 - **aside** , **sidebar** - боковая колонка, сайдбар
 - **menu** , **nav** - навигационное меню
 - **list** - список, набор
 - **inner** - внутренний блок
 - **content** - содержимое, контент
 - **top** , **right** , **bottom** , **left** - верх, низ, лево, право
 - **active** , **deactive** , **disabled** - активность, неактивность, заблокировано
 - **visible** , **invisible** , **hidden** - видимый, невидимый, скрытый
- ЗАПРЕЩЕННЫЕ слова для использования в названиях классах, поскольку элементы вырезаются со страницы AdBlock
 - ads, ad, adv
 - cookie
 - banner

Больше тут: [список запрещенных к использованию классов и идентификаторов](#)

Переменные

- Для имен переменных необходимо использовать правильные английские слова без артиклей в единственном числе.

- Написание имен переменных строго в нижнем регистре.
- Все переменные должны состоять минимум из двух слов, разделитель между словами символ подчеркивания `_`
- Именованние переменных необходимо начинать со слова-префикса в котором указывается назначение переменной.

Например:

- `$font_text`
- `$color_hover`
- `$grid_hover`
- и т.д.

- Переменные с одинаковой обработкой необходимо объединять в тар. [Документация](#)

```
//список размеров
$grid_size: (
  's': $grid_s,
  'm': $grid_m,
  'l': $grid_l,
  'xl': $grid_xl
);
```

JS

- Обязательно использовать jQuery
- Все события взаимодействия с объектной моделью документа должны быть повешены на специальные селекторы с префиксом js.

Например: `.js-tab-change`

- Список плагинов для основных нужд
 - [FormStyler](#) - стилизация выпадающих списков
 - [Ion.RangeSlider](#) - ползунки (слайдер диапазонов)
 - [OwlCarousel](#) - слайдер-карусель
 - [maskedinput](#) - маска ввода (например, для телефонов)
 - [FlipClock](#) - таймер обратного отсчета или часы
 - [Moment](#) - для работы с датами

Favicon

- Обязательно наличие корректно сформированной группы файлов для отображения favicon (генератор [realfavicongenerator.net](#))

Исходное изображение для генератора должно быть квадратным и минимально 260x260 пикселей размером.

Список необходимых файлов:

- favicon.ico
- android-chrome-96x96.png
- apple-touch-icon.png
- favicon-16x16.png
- favicon-32x32.png
- mstile-150x150.png
- browserconfig.xml
- safari-pinned-tab.svg
- site.webmanifest

Пример подключения на страницу:

```
<link rel="apple-touch-icon" sizes="120x120" href="/_assets/favicon/apple-touch-  
icon.png">  
<link rel="icon" type="image/png" sizes="32x32" href="/_assets/favicon/favicon-  
32x32.png">  
<link rel="icon" type="image/png" sizes="16x16" href="/_assets/favicon/favicon-  
16x16.png">  
<link rel="manifest" href="/_assets/favicon/site.webmanifest">  
<link rel="mask-icon" href="/_assets/favicon/safari-pinned-tab.svg" color="#5bbad5">  
<link rel="shortcut icon" href="/_assets/favicon/favicon.ico">  
<meta name="msapplication-TileColor" content="#2b5797">  
<meta name="msapplication-config" content="/_assets/favicon/browserconfig.xml">
```

- Положение favicon в структуре проекта: _source/favicon/
- Необходимо также дополнительно задавать цвет шапки вкладки для устройств на Android 5+. Цвет #0099da указан ДЛЯ ПРИМЕРА.

```
<meta name="theme-color" content="#0099da">
```

Шрифты

- Каждый шрифт должен быть размещен в отдельной подпапке, в папке _source/font/Имя шрифта
- В папке со шрифтом должен быть размещен файл css в котором производится подключение данного шрифта
- Каждый шрифт должен быть представлен как минимум следующими файлами *.ttf, *.woff, *.eot

В случае если предоставлен только один формат файла шрифта, остальные необходимо получить самостоятельно, используя конвертеры.

Например:

- Для всех расширений: <http://www.font2web.com/>
- Для отдельных расширений: <https://everythingfonts.com/woff-to-ttf>

Пример структуры папки шрифта (_source/font/audiext):

audiext

- audiext.css
- AudiType-ExtendedBold.eot
- AudiType-ExtendedBold.ttf
- AudiType-ExtendedBold.woff
- AudiType-ExtendedNormal.eot
- AudiType-ExtendedNormal.ttf
- AudiType-ExtendedNormal.woff

Содержимое файла audiext.css:

```
<style>
  @font-face {
    font-family: 'AudiTypeExt';
    src: url('AudiType-ExtendedNormal.eot');
    src: url('AudiType-ExtendedNormal.eot?#iefix') format('embedded-opentype'),
          url('AudiType-ExtendedNormal.woff') format('woff'),
          url('AudiType-ExtendedNormal.ttf') format('truetype');
    font-weight: normal;
    font-style: normal;
  }

  @font-face {
    font-family: 'AudiTypeExt';
    src: url('AudiType-ExtendedBold.eot');
    src: url('AudiType-ExtendedBold.eot?#iefix') format('embedded-opentype'),
          url('AudiType-ExtendedBold.woff') format('woff'),
          url('AudiType-ExtendedBold.ttf') format('truetype');
    font-weight: bold;
    font-style: normal;
  }
</style>
```

Подключаем шрифт в файле со стилями (scss), например _source/css/block/_font.scss:

```
@import url('../font/audiext/audiext.css');
```

Изображения

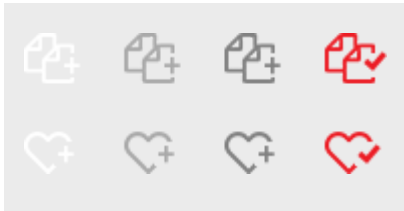
- Все изображения используемые в верстке должны быть предварительно обработаны для уменьшения размера.

Используем tinypng.com

Стандартный функционал Photoshop "Save for web" (Сохранить для Web) не оптимизирует изображения должным образом.

- Иконки в виде изображений объединяются в спрайты

Пример спрайта иконок:



- Иконки в виде SVG объединяются в SVG спрайты АВТОМАТИЧЕСКИ СБОРЩИКОМ

Пример SVG спрайта:

```
<svg xmlns="http://www.w3.org/2000/svg">
  <g id="check" viewBox="0 0 24 24">
    <path d="M0 0h24v24H0z" fill="none"/>
    <path d="M9 16.17L4.83 12l-1.42 1.41L9 19 21 7l-1.41-1.41z"/>
  </g>
  <g id="phone" viewBox="0 0 24 24">
    <path d="M0 0h24v24H0z" fill="none"/>
    <path d="M6.62 10.79c1.44 2.83 3.76 5.14 6.59 6.59l2.2-2.2c.27-.27.67-.36
1.02-.24 1.12.37 2.33.57 3.57.57 0 1 .45 1 1v20c0 .55-.45 1-1 1-9.39 0-17-7.61-17-17
0-.55.45-1 1-1h3.5c.55 0 1 .45 1 1 0 1.25.2 2.45.57 3.57.11.35.03.74-.25 1.02l-2.2
2.2z"/>
  </g>
  <g id="pin" viewBox="0 0 24 24">
    <path d="M12 2C8.13 2 5 5.13 5 9c0 5.25 7 13 7 13s7-7.75 7-13c0-3.87-3.13-7-7-7-
7zm0 9.5c-1.38 0-2.5-1.12-2.5-2.5s1.12-2.5 2.5-2.5 2.5 1.12 2.5 2.5-1.12 2.5-2.5 2.5z"/>
    <path d="M0 0h24v24H0z" fill="none"/>
  </g>
</svg>
```

Использование на странице:

```
<svg viewBox="0 0 24 24" class="icon icon_service">
  <use xmlns:xlink="http://www.w3.org/1999/xlink" xlink:href="#pin"></use>
</svg>
```

- Иконки из иконочных шрифтов НЕ вставляются как изображения

Текст

- Текст статических элементов должен быть типографирован <https://www.artlebedev.ru/typograf/>
- Текст статических элементов должен быть без ошибок <https://www.artlebedev.ru/orfograf/>

Активные элементы

Состояния

Если в макете не представлены все состояния - необходимо запрашивать их у дизайнера.

Все состояния должны быть в верстке в обязательном порядке.

- Обычное
- Наведение (:hover)
- Нажатие (:active)
- Фокус (:focus)
- Ссылка посещенная (:visited)

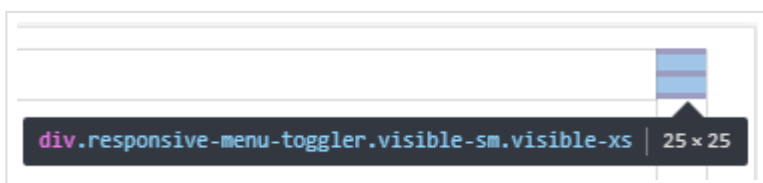
Дополнительно

- Курсор в виде руки вместо стрелки (`cursor: pointer;`)
- Для всех нестандартных активных элементов (кроме `<a>`, `<button>`, `<input>`, `<select>`, `<textarea>`) нужно использовать атрибут `role` со значениями `button` или `link` (`role="button"` `role="link"`)
- Убрать случайное выделение текста (`user-select: none;`) для кнопок, выпадающих списков и т.д.
- Добавить обработчик нажатия на Enter при наличии фокуса на элементе
- Предусмотреть спиннер (отображение загрузки), если состояние элемента зависит от асинхронного события

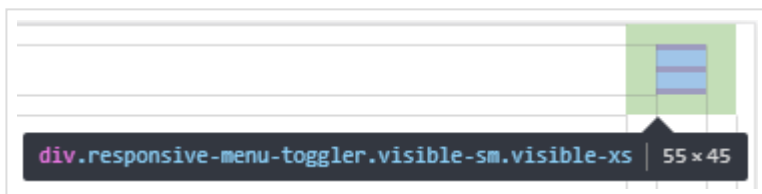


- Область клика активного элемента должна быть размером большим либо равным размеру всего активного элемента

НЕ правильно:



Правильно:



Ссылки

На электронную почту ссылка `mailto:`

```
<a href="mailto:example@domain.com">Обратная связь</a>
```

На телефон ссылка `tel:`

В номере телефона указываемом в атрибуте href используются только цифры, без пробелов и форматирования.

Начинается телефон с +7 (а НЕ 8).

```
<a href="tel:+74957851001">+7 (495) 785-10-01</a>
```

Формы

- Все формы должны быть на странице с использованием тега `<form>`
- Все элементы формы должны иметь `tabindex`
- Все инпуты на странице десктопа должны быть внешне соответствующими макету (выглядеть также как на макете)
- Все инпуты на странице мобильного устройства должны быть семантически верными, но внешне соответствующими макету (<http://mobileinputtypes.com/>)

На мобильной версии НЕ используются js плагины для изменения внешнего вида выпадающих списков (`<select>`), списков множественного выбора (`<select multiple="">`), только CSS стили.

- Все элементы формы являются активными элементами и должны соответствовать требованиям для активных элементов
- Чекбоксы и радиокнопки должны быть помещены в тег `<label>` вместе с описанием, кликабельным должен быть не только сам чекбокс (радиокнопка) но и текст описания

Пример неправильной разметки

- ☐ Неправильный чекбокс 1
- ☐ Неправильный чекбокс 2
- ☐ Неправильный чекбокс 3

```

<div class="form__row">
  <input type="checkbox" name="test[]" value="1" />
  <label>Неправильный чекбокс 1</label>
</div>
<div class="form__row">
  <input type="checkbox" name="test[]" value="2" />
  <label>Неправильный чекбокс 2</label>
</div>
<div class="form__row">
  <input type="checkbox" name="test[]" value="3" />
  <label>Неправильный чекбокс 3</label>
</div>

```

Пример правильной разметки

- ☐ Правильный чекбокс 1
- ☐ Правильный чекбокс 2
- ☐ Правильный чекбокс 3

```

<div class="form__row">
  <label>
    <input type="checkbox" name="test[]" value="1" />
    Правильный чекбокс 1
  </label>
</div>
<div class="form__row">
  <label>
    <input type="checkbox" name="test[]" value="2" />
    Правильный чекбокс 2
  </label>
</div>
<div class="form__row">
  <label>
    <input type="checkbox" name="test[]" value="3" />
    Правильный чекбокс 3
  </label>
</div>

```

Таблицы

Все таблицы на странице должны быть представлены в нескольких вариантах:

- Основной (на десктоп) из макета
- Уменьшенный (на мобильные устройства) состоящий максимум из двух столбцов

Таблица для десктопа

--	--	--	--

	Стандарт	Комфорт	Минивэн
Минимальный заказ	299 руб	450 руб	550 руб
Включено в минимальный заказ	15 минут	20 минут	20 минут

Таблица для мобильных устройств

Стандарт	
Минимальный заказ	299 руб
Включено в минимальный заказ	15 минут
Комфорт	
Минимальный заказ	450 руб
Включено в минимальный заказ	20 минут
Минивэн	
Минимальный заказ	550 руб
Включено в минимальный заказ	20 минут

В случае если таблицу нет возможности разложить для мобильных устройств, то необходимо отправить запрос об этом в момент оценки работы.

В самом крайнем случае (ТОЛЬКО если нет возможности подготовить адаптированную таблицу) для таблицы на мобильных устройствах необходимо задать фиксированную ширину и добавить горизонтальную прокрутку.

Сборка

Проект автоматически собирается при помощи Gulp

Модули (часть файла package.json)

```
"devDependencies": {
  "gulp": "^3.9.1",
  "gulp-autoprefixer": "^3.1.1",
  "gulp-cheerio": "^0.6.3",
  "gulp-concat": "^2.6.1",
  "gulp-cssnano": "^2.1.2",
```

```
"gulp-imagemin": "^3.4.0",
"gulp-rename": "^1.2.2",
"gulp-sass": "^2.3.2",
"gulp-svg-sprite": "^1.3.7",
"gulp-svgmin": "^1.2.4",
"gulp-uglify": "^1.5.4",
"gulp-watch": "^4.3.11",
"pump": "^2.0.0"
}
```

Конфигурация файла для Gulp (gulpfile.js)

```
<script>
'use strict';

var gulp = require('gulp'),
    watch = require('gulp-watch'),
    prefixer = require('gulp-autoprefixer'),
    sass = require('gulp-sass'),
    cssnano = require('gulp-cssnano'),
    uglify = require('gulp-uglify'),
    imagemin = require('gulp-imagemin'),
    rename = require('gulp-rename'),
    svgmin = require('gulp-svgmin'),
    cheerio = require('gulp-cheerio'),
    svgsprite = require('gulp-svg-sprite'),
    concat = require('gulp-concat'),
    pump = require('pump');

var path = {
  src: {
    js: './_source/js/**/*.js',
    style: './_source/css/**/*.scss',
    css: './_source/css/**/*.css',
    img: './_source/img/**/*.*',
    svg: './_source/svg/**/*.svg',
    font: './_source/font/**/*.*',
    module: './_source/module/**/*.*',
    favicon: './_source/favicon/*.%'
  },
  build: {
    js: './_assets/js/',
    style: './_assets/css/',
    css: './_assets/css/',
    img: './_assets/img/',
    svg: './_assets/svg/',
    font: './_assets/font/',
    module: './_assets/module/',
    favicon: './_assets/favicon/'
  }
};

gulp.task('js:work:build', function (cb)
```

```

        {
    pump([
        gulp.src(path.src.js),
        concat('bundle.min.js'),
        gulp.dest(path.build.js)
    ],
        cb
    );
});

gulp.task('js:build', function (cb)
    {
    pump([
        gulp.src(path.src.js),
        uglify(),
        concat('bundle.min.js'),
        gulp.dest(path.build.js)
    ],
        cb
    );
});

gulp.task('style:build', function (cb)
    {
    pump([
        gulp.src(path.src.style),
        sass({outputStyle: 'compressed'}),
        prefixer({browsers: ['last 3 version', '> 1%', 'ie 8', 'ie 9', 'Opera 12.1'],
cascade: false}),
        gulp.dest(path.build.style)
    ],
        cb
    );
});

gulp.task('css:build', function (cb)
    {
    pump([
        gulp.src(path.src.css),
        prefixer({browsers: ['last 3 version', '> 1%', 'ie 8', 'ie 9', 'Opera 12.1'],
cascade: false}),
        cssnano(),
        gulp.dest(path.build.css)
    ],
        cb
    );
});

gulp.task('img:build', function (cb)
    {
    pump([
        gulp.src(path.src.img),
        imagemin(),
        rename(function (path)

```



```

        {
            path.extname = (path.extname + "").toLowerCase();
        }),
        gulp.dest(path.build.img)
    ],
    cb
);
});

gulp.task('img:work:build', function (cb)
{
    pump([
        gulp.src(path.src.img),
        rename(function (path)
        {
            path.extname = (path.extname + "").toLowerCase();
        }),
        gulp.dest(path.build.img)
    ],
    cb
);
});

gulp.task('svg:build', function (cb)
{
    pump([
        gulp.src(path.src.svg),
        svgmin({
            js2svg: {
                pretty: true
            }
        }),
        cheerio({
            run: function ($)
            {
                $('[fill]').removeAttr('fill');
                $('[stroke]').removeAttr('stroke');
                $('[style]').removeAttr('style');
            },
            parserOptions: {xmlMode: true}
        }),
        svgsprite({
            mode: {
                symbol: {
                    sprite: "../sprite.svg"
                }
            }
        }),
        gulp.dest(path.build.svg)
    ],
    cb
);
});

gulp.task('font:build', function (cb)

```

```

        {
    pump([
        gulp.src(path.src.font),
        gulp.dest(path.build.font)
    ],
        cb
    );
});

gulp.task('module:build', function (cb)
    {
    pump([
        gulp.src(path.src.module),
        gulp.dest(path.build.module)
    ],
        cb
    );
});

gulp.task('favicon:build', function (cb)
    {
    pump([
        gulp.src(path.src.favicon),
        gulp.dest(path.build.favicon)
    ],
        cb
    );
});

gulp.task('build', [
    'js:build',
    'style:build',
    'css:build',
    'img:build',
    'svg:build',
    'font:build',
    'module:build',
    'favicon:build'
]);

gulp.task('watch', function ()
    {
    watch([path.src.js], function ()
        {
        gulp.start('js:work:build');
        });
    watch([path.src.style], function ()
        {
        gulp.start('style:build');
        });
    watch([path.src.css], function ()
        {
        gulp.start('css:build');
        });
    watch([path.src.img], function ()

```

```
        {
            gulp.start('img:work:build');
        });
        watch([path.src.svg], function ()
            {
                gulp.start('svg:build');
            });
        watch([path.src.font], function ()
            {
                gulp.start('font:build');
            });
        watch([path.src.module], function ()
            {
                gulp.start('module:build');
            });
    });

    gulp.task('default', ['build', 'watch']);
</script>
```