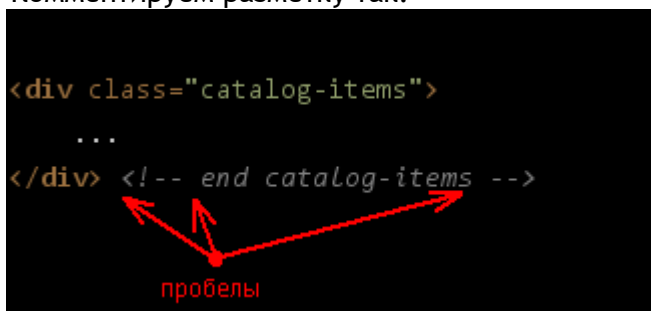


Во всех проектах необходимо использовать наш CSS Reset (выдадим позже).

Верстать нужно валидно, с обязательно проверкой на <http://validator.w3.org/>

### Требования по разметке HTML и CSS:

1. Не использовать id, совсем. Все исключительно на классах.
2. Имена классам давать строго по смыслу сайта, на английском языке, никаких коротких названий и сокращений типа "fl-1", "mb-5", никаких ВЕМ и тому подобных методологий. Имя класса разделять дефисом (-), никаких символов подчеркивания и т.п. Например: "left-bar", "main-content", "main-banner", "our-benefits", "catalog-items", "catalog-item", "news", "news-item", "search-filter", "add-to-cart".
3. <header> только для хедера, <footer> только для футера сайта, больше нигде не должны встречаться.
4. У большинства блоков отступы стараемся делать через margin-bottom, т.е. вниз от блока, а не перед ним (сверху). Для footer можно сверху.
5. Комментируем разметку так:



```
<div class="catalog-items">
  ...
</div> <!-- end catalog-items -->
```

пробелы

Такой комментарий необходимо добавлять в конце каждого элемента, имеющего три и более строки внутри.

6. При вставке картинок (<img src="" alt="" />) в большинстве случаев необходим оборачивающий блок с жестким заданием ширины и/или высоты, плюс overflow:hidden. Картинке при этом прописываем в css конкретную высоту и/или ширину, max-height и/или max-width. Нужно для того, чтобы какую картинку не вставили в код, она не должна вылезти за пределы своего блока и обрушить дизайн.
  7. Пунктирные ссылки должны быть ссылками — <a href="#" class="ajax-link"></a>, а не <span> или чем-либо еще. Подчеркивание при наведении убирать не нужно.
  8. Каждый тег обязательно должен быть закрыт. Пример:  
<input type="text" name="" value="" />  
<img src="" alt="" />  
<br />
- И не важно, что стандарт html5 разрешает этого не делать. Наша верстка по-прежнему используется в XML системах, где данное требование обязательно.
9. В целом надо стараться html код делать проще, т.е. использовать как можно меньше элементов (тегов). Все что можно сделать на одном диве — делаем. Не нужно в него еще один-два-три вкладывать.
  10. Активно используем теги h1-h6. Размеры задаем на свое усмотрение, согласно логике и дизайну, обычно это понятно и особых трудностей не вызывает.
  11. Пример оформления CSS:

```

/* -----
   Стили блоков, содержимого страниц | Other styles
   ----- */

.wrapper { /* Обертка страницы */
  width:900px;
  margin:0 auto;
}

header {
  position:relative;
  height:175px;
  background:url(../images/header-line.png) 0 0 repeat-x;
}

header .logo {
  position:absolute;
  top:0;
  left:0;
  width:149px;
  height:139px;
  background:url(../images/logo.png) 0 0 no-repeat;
}

header .logo a {
  display:block;
  width:100%;
  height:100%;
}

header .phone {
  position:absolute;
  top:20px;
  right:0;
  font-size:18px;
  line-height:24px;
}

header .phone a {
  font-weight:normal;
  font-size:24px;
}

.content {
  margin-bottom:20px;
}

```

пробел

кавычек нет

Все значения ТОЛЬКО в px или процентах.  
У нулей "px" не пишем.  
У шрифтов только в пикселях.

наследуем родительские классы,  
но не стоит все-все перечислять, достаточно 3-4 основных.

глобальные блоки отделяем отступом

Основной принцип - это каждое новое правило на отдельной строке (допускается и на одной, если на класс 1-2 правила), а также делать отступы для вложенных элементов.

Если в CSS правила на одной строке, скобки отделяем пробелами:

```
.button:hover { background-position:0 -30px; }
```

Если селекторов несколько — каждый с новой строки:

```
.button:hover a,  
.button:hover input[type=submit],  
.button:hover input[type=button] {  
    background-position:0 -30px;  
}
```

У селекторов в большинстве случаев обязан присутствовать уникальный родительский класс:

```
.page-contacts .contacts li { ... }  
.our-benefits .item .photo i { ... }
```

Простые селекторы вида `.contacts li`, `.photo i` без родительского класса не допускаются!

**12.** Активно используем CSS спрайты. Особенно это обязательно для различных hover эффектов. Ведь при наведении браузер должен немедленно показать другую картинку, а если она сделана отдельной, то он её только при первом наведении начнет загружать...

**13.** Шрифт задавать так и только так: `font:normal normal 12px/15px Arial, Helvetica, sans-serif`;

В дальнейшем, если семейство не меняется: `font-size`, `line-height`, `font-weight` и т.д. Все значения исключительно в пикселях.

При указании `font-size` обязательно задаем и `line-height`!

**14.** Имена картинкам давать осмысленные, согласно их положению в блоках.

Хорошо: `manager-help-photo.jpg`, `main-slider-arrows.png`, `our-benefit-economy.jpg`

Плохо: `photo.jpg`, `arrows.png`, `economy.jpg`

Ужасно: `pic1.jpg`, `pic2.png`, `pic3.png` и т.д.

Для перечисляющихся картинок контента допускается:

```
main-slider-img1.jpg, main-slider-img2.jpg, main-slider-img3.jpg;  
catalog-item-img1.jpg, catalog-item-img2.jpg, catalog-item-img3.jpg.
```

**15.** Из браузеров — ориентация на свежие. IE 7 вообще не смотрим, если об этом не сказано в задании, IE 8 чтобы все на своих местах было.

**17.** Активно используем CSS3: `:after`, `:before`, `border-radius`, `box-shadow`, `text-shadow`, `linear-gradient` и другие. По градиентам неплохой генератор: <http://css3please.com/> по сложным градиентам: <http://www.css3factory.com/linear-gradients/>

Используя эти правила, в современных браузерах все будет красиво, а в старых — квадратным, но при этом все на своих местах.

Иными словами — **все, что можно сделать на CSS — делаем**, в том числе различные уголки и другие фигуры на сайте. Документ Вам в помощь — <http://css-tricks.com/examples/ShapesOfCSS/>

**16. Верстаем Pixel Perfect.** Например, с помощью утилиты Vtrite (для Windows, позволяет делать окна прозрачными -> накладываем браузер на макет и подгоняем) либо другим удобным Вам способом.

Требование обязательное, но в то же время ситуации разные бывают, особенно в части шрифтов, поэтому иногда допускаются незначительные расхождения.

Отчасти это даже проще, чем линейкой все в точности мерить в фотошопе. Удобней сперва прописать "на глаз", а затем все выровнять по макету.

## Нюансы по jQuery:

1. `next()`, `parent()`, `children()` и т.п. использовать в случае крайней необходимости, вместо них всегда: `find()`, `closest()`, `closest().find()`.

2. Для функции `.click(function(){ ... })` желательно не использовать конструкцию `return false;` Вместо нее делаем так:

```
$('.a.button').click(function(e) {  
    e.preventDefault();  
});
```

3. Стандартное время анимации: 200-300мс. В jQuery по умолчанию 400мс, что много, люди уже насытились различными эффектами на сайтах и им важнее скорость работы. Данный пункт может не касаться слайдеров и других сложных штук. Рассчитан скорее на время появления попапов, раскрывающихся/выпадающих списков и др.

4. Для слайдеров желательно использовать скрипт <http://kenwheeler.github.io/slick/>. Сам скрипт вставляем в конец файла `/js/scripts.js`, инициализацию в секцию `$(document).ready()`

5. Активно используем переменные. Если селектор вида `$('.some-class')` используется более одного раза, то нужно вынести в переменную: `var some_class = $('.some-class');`