Как сверстать веб-страницу. Часть 2 — Bootstraptutorial

[HTML](https://habrahabr.ru/hub/html5/)\*, [CSS](https://habrahabr.ru/hub/css/)\*, [Веб-разработка](https://habrahabr.ru/hub/webdev/)\*

Введение

Уважаемый читатель, эта статья является второй частью цикла статей, посвященных вёрстке.  
  
В [первой части](http://habrahabr.ru/post/202408/) мы верстали шаблон [Corporate Blue](http://www.pcklab.com/item.php?id=16) от студии Pcklaboratory с помощью стандартных средств на чистом HTML и CSS. В данной статье мы попробуем сверстать этот же шаблон, но с помощью CSS фреймворка [Bootstrap 3](http://getbootstrap.com/).  
  
  
  
Преимущество использования CSS фреймворков заключается в том, что верстальщику не нужно думать о многих нюансах верстки, которые за него уже продумали создатели фреймворков. К таким нюансам относятся кроссбраузерность, поддержка различных разрешений экранов и многое другое. Верстальщик лишь указывает, что, как и когда нужно показать, остальное фреймворк делает сам. Данный подход может сильно ускорить вёрстку сайта. К преимуществам Bootstrap относится и его популярность. Это означает, что другому верстальщику будет проще поддерживать ваш код.  
  
Недостатком использования фреймворков является тот факт, что странице придется целиком «нести» за собой лишние стили фреймворка, даже если она использует лишь их малую часть. Фреймворк является отличным инструментом для прототипирования и создания страниц, для которых дизайн вторичен, например страницы администрирования. Если же у вас есть очень специфический дизайн, то сверстать его с помощью фреймворка может оказаться сложнее, чем нативными средствами. Тем не менее, и это возможно.

Об использовании Bootstrap

В настоящее время есть несколько способов работы со стилями Bootstrap.

Без использования LESS

Для новичков сам Bootstrap [рекомендует](http://getbootstrap.com/getting-started/#customizing) следующий подход: нужно скачать с сайта скомпилированный Bootstrap и положить его в свой проект, ничего не изменяя. Затем нужно создать свой пустой CSS файл и подключить его после bootstrap.css.

<link href="css/bootstrap.min.css" rel="stylesheet">

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

После этого, для того чтобы изменить стили Bootstrap вам нужно перебить их в своем styles.css примерно в таком виде:

a { color: #beceda; }

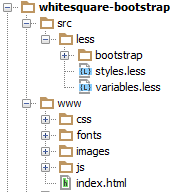
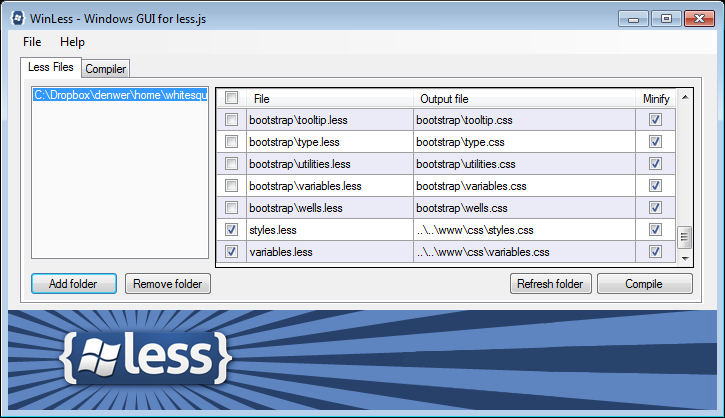
Очевидным минусом данного подхода является то, что вам придется вручную искать нужные стили, которые требуется перебить и не всегда это будет тривиально, т.к. некоторые параметры Bootstrap применяются ко многим селекторам в изменённом виде, например через формулы. Небольшую помощь здесь может оказать инструмент [Customize](http://getbootstrap.com/customize), он поможет скомпилировать правильно ваши изменения, но только один раз. Если в будущем вы захотите изменить какой-то параметр, то придется заново вбивать изменненные значения для всех полей, чтобы скомпилировать свои стили.

С использованием LESS

Данный способ подразумевает, что все переменные Bootstrap хранятся в .less файлах. Разработчик работает с этими переменными и по необходимости вручную или автоматически компилирует их в CSS файлы, а сам HTML подключает только скомпилированные CSS файлы. Именно этот вариант и будет рассматриваться в статье, как самый гибкий.  
  
Существует большое количество способов скомпилировать LESS файлы и Bootstrap оставляет это на усмотрение разработчика. Сам Bootstrap использует для компиляции [Grunt](http://gruntjs.com/), вы можете предпочесть [плагин](http://plugins.jetbrains.com/plugin?pr=idea&pluginId=7059) для продуктов JetBrains, а мы, поскольку статья ориентирована на новичков, посмотрим в сторону более простых решений. Такими решениями являются программы [WinLess](http://winless.org/)для Windows, [SimpLESS](http://wearekiss.com/simpless) для Mac или [Koala](http://koala-app.com/) для Linux. Все эти программы делают примерно одно и то же: получают на вход папку с LESS файлами и слушают изменения в них. Как только вы вносите изменения в любой файл – тут же он компилируется в указанный CSS файл. Таким образом вам нет необходимости запускать компиляцию руками после каждого изменения. Вы изменяете LESS файл, сохраняете его и тут же видите изменения на сайте в уже скомпилированном, сжатом виде.

Создание проекта

Первым шагом давайте создадим простую структуру файлов для нашего проекта. 

* Создаем папку с названием проекта, например whitesquare-bootstrap.
* В ней нужно создать две вложенные папки: src – для исходных файлов и www – для файлов конечного сайта.
* В папке www создаем пустую папку images и пустой файл index.html.
* Затем нужно [скачать сам Bootstrap](https://github.com/twbs/bootstrap/releases/download/v3.1.0/bootstrap-3.1.0-dist.zip) и скопировать содержимое архива в папку www нашего проекта.
* Поскольку мы решили использовать LESS в нашем проекте, то придется еще [скачать исходники Bootstrap](https://github.com/twbs/bootstrap/archive/v3.1.0.zip) и скопировать оттуда папку less в папку src нашего проекта.
* Рядом с появившейся папкой less/bootstrap создадим два пустых файла styles.less и variables.less. В них мы будем перебивать переменные Bootstrap и описывать свои стили. Такой подход позволит потом быстро обновить Bootstrap.  
    
  
* Затем нужно настроить компиляцию LESS файлов в CSS. Посмотрим, как это делается в WinLess. Сначала нажмите «Add folder» и укажите путь до папки с LESS файлами:  
  C:\whitesquare-bootstrap\src\less  
  Затем у вас появится список всех файлов этой папки. Можно убрать все галочки. Нас интересуют последние два файла styles.less и variables.less. Кликните на них правой кнопкой и выберите из контекстного меню «Select output file» и укажите путь, куда будут компилироваться CSS файлы:  
  ..\..\www\css\styles.css  
  ..\..\www\css\variables.css  
  После этого любое изменение этих LESS файлов с сохранением будет приводить к перекомпиляции CSS файлов.  
    
  

Предварительный осмотр

После создания структуры файлов открываем psd файл в Photoshop. Важно внимательно осмотреть шаблон и оценить его. Нам нужно понять следующие вещи:

* Как будут нарезаться изображения?
* Какие будут использоваться компоненты?
* Какими будут основные стили?
* Какой макет страницы у нас получится?

Только после того, как вы мысленно себе ответите на эти вопросы, можно переходить к вёрстке. Давайте рассмотрим эти вопросы по-порядку.

Общие изображения

На данном этапе нужно нарезать и сохранить только общие изображения, которые будут на всех страницах сайта и не относятся к контенту. В нашем случае это будет светло-серый фон страницы, фон заголовка, изображение карты, два логотипа и кнопки социальных сетей.  
  
Сохраняем изображение карты:  
  
images/map.png  
  
Сохраним логотипы следующим образом:  
  
images/logo.png  
images/footer-logo.png  
  
Повторяющиеся фоновые изображения необходимо вырезать минимальным кусочком достаточным для образования полного изображения повторением по вертикали и горизонтали.  
  
/images/bg.png  
/images/h1-bg.png  
  
Иконки социальных сетей с одинаковыми размерами удобно сохранить в один файл и использовать как спрайты для более быстрой загрузки. Более подробно про склейку изображений описано в первой части. В итоге получится два файла:  
  
/images/social.png  
/images/social-small.png

Компоненты

Основное отличие вёрстки с помощью Bootstrap от вёрстки нативными средствами заключается в том, что Bootstrap вводит такое понятие, как компоненты. Компоненты представляют из себя часто используемые готовые HTML блоки с предопределенными стилями. Иногда компоненты используют JavaScript. Верстальщик может использовать как готовый компонент, так и определить свой внешний вид для него. Для этого часто нужно лишь поменять значение переменных в Bootstrap. Если нужны более гибкие изменения, верстальщик всегда может изменить HTML и CSS по своему усмотрению.  
  
Если окинуть взглядом наш шаблон, то можно увидеть, что нам понадобятся следующие компоненты: 

1. Для верстки колонками — сеточная система (row, col)
2. Для поиска – инлайновая форма (form-inline), сгруппированные контролы (input-group), кнопка (btn)
3. Для навигации — навигационная панель (navbar) и сама навигация (nav)
4. Для отображения подменю – группированный список (list-group)
5. Для блока карты – визуальная панель (panel)
6. Для отображения большого центрального блока – jumbotron
7. Для отображения рамок фотографий – миниатюры (thumbnail)

Более подробно на каждом компоненте мы остановимся, когда он нам встретится в вёрстке.

Основные стили

В Bootstrap все стили по умолчанию уже заданы, нам нужно только их перебить, если они отличаются от нашего дизайна. Сделаем это в файле src/less/variables.css.  
  
В первую очередь нужно добавить переменные, которых нет в настройках Bootstrap для того, чтобы можно было их использовать в дальнейшем. У нас это только специфический шрифт дизайна.

@brand-font: 'Oswald',sans-serif;

Если вы хотите использовать шаблон для русских сайтов, то шрифт Oswald можно попробовать заменить на наиболее близкий Cuprum, который поддерживает кириллицу.  
  
А теперь заменим настройки Bootstrap на свои:

/\*серый фон страницы\*/

@body-bg: #f8f8f8;

/\*специфический голубой фон данного дизайна\*/

@ brand-primary: #29c5e6;

/\*фон панелей\*/

@panel-bg: #f3f3f3;

/\*цвет рамки панелей\*/

@panel-inner-border: #e7e7e7;

/\*убираем скругление у блоков\*/

@border-radius-base: 0;

/\*первичные кнопки имеют голубой фон\*/

@btn-primary-bg: @brand-primary;

/\*если ширина экрана больше 992px, то ширина контейнера 960px\*/

@container-md: 960px;

/\*если ширина экрана больше 1200px, то ширина контейнера так же 960px\*/

@container-lg: @container-md;

/\*основным шрифтом будет Tahoma\*/

@font-family-base: Tahoma, sans-serif;

/\*задаем базовый размер\*/

@font-size-base: 12px;

/\*основной цвет текста на странице\*/

@text-color: #8f8f8f;

/\*серый фон текстовых полей\*/

@input-bg: @panel-bg;

/\*серая рамка текстовых полей\*/

@input-border: @panel-inner-border;

/\*серый цвет текста в полях\*/

@input-color: #b2b2b2;

Все переменные, которые есть в Bootstrap можно посмотреть на странице<http://getbootstrap.com/customize/>  
  
После того, как мы закончили с переменными, давайте начнем прописывать стили нашего дизайна в файле styles.less. Сначала подключим сам Bootstrap и наши переменные:

@import "bootstrap/bootstrap.less";

@import "variables.less";

Не все стили, заданные Bootstrap по умолчанию можно изменить переменными, давайте сделаем это вручную:

p {

margin: 20px 0;

}

.form-control {

box-shadow: none;

}

.btn {

font-family: @brand-font;

}

Здесь мы убрали тень у элементов формы, а текстам в кнопках указали специфический шрифт страницы.  
  
Затем опишем фон страницы и верхнюю полоску:

body {

border-top: 5px solid #7e7e7e;

background-image: url(../images/bg.png);

}

Далее в тексте не будет указываться в какой файл пишутся стили. Просто запомните, что все переменные мы сохраняем в файл variables.less, а CSS стили будем хранить в styles.less.

Каркас HTML

Вёрстку сайта начинаем традиционно с каркаса HTML. Вставляем в файл index.html код простейшего шаблона со страницы [Getting started](http://getbootstrap.com/getting-started/" \l "template), предварительно убрав всё лишнее:

<!DOCTYPE html>

<html>

<head>

<title>Whitesquare</title>

<meta name="viewport" content="width=device-width, initial-scale=1.0">

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

<!--[if lt IE 9]>

<script src="https://oss.maxcdn.com/libs/html5shiv/3.7.0/html5shiv.js"></script>

<script src="https://oss.maxcdn.com/libs/respond.js/1.3.0/respond.min.js"></script>

<![endif]-->

</head>

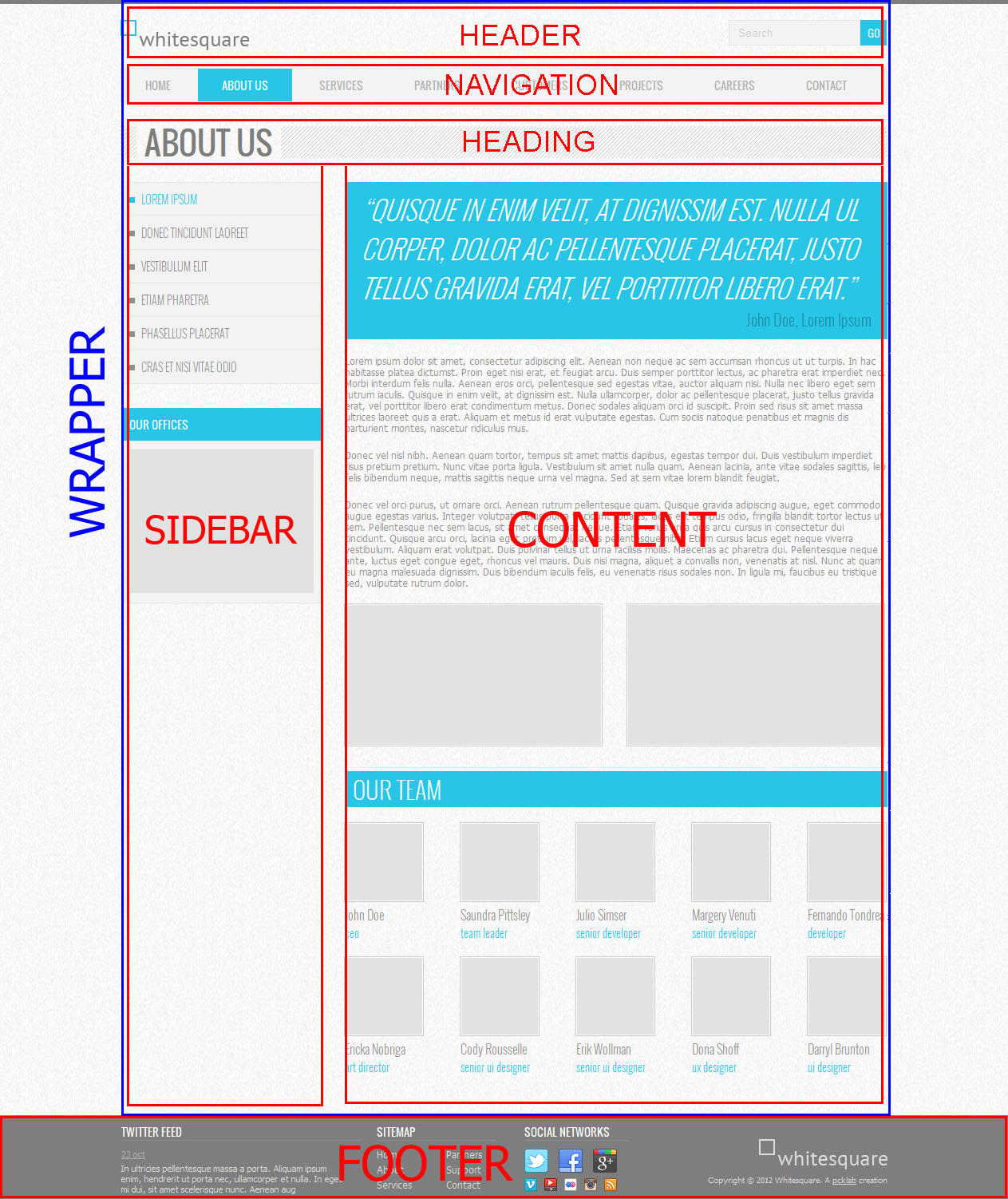
<body>

</body>

</html>

В этом блоке создается HTML5 структура документа. В title указываем название нашей страницы – Whitesquare. Метатегом viewport указываем, что ширина страницы на мобильных устройствах будет равна ширине экрана и начальный масштаб будет 100%. Затем подключается файл стилей. И для версий Internet Explorer меньше девятой подключаем скрипты, позволяющие правильно отображать нашу верстку.

Макет

В данном случае, мы видим, что сайт состоит из двух частей: основного контейнера с содержимым, который центрируется на экране и тянущегося футера. Основной контейнер состоит из двух колонок: основного контента и сайдбара. Над ними находится шапка (header), навигация (nav) и название страницы (.heading).   
  
  
  
Давайте добавим в body следующий код:

<body>

<div class="wrapper container">

<header></header>

<nav></nav>

<div class="heading"></div>

<div class="row">

<aside class="col-md-7"></aside>

<section class="col-md-17"></section>

</div>

</div>

<footer></footer>

Здесь нам встречается первый компонент Bootstrap – [колонки](http://getbootstrap.com/css/#grid). Родительскому элементу колонок задается класс «row», а классы колонок начинаются с префикса «col-», затем идет размер экрана (xs, sm, md, lg), а заканчиваются относительной шириной колонки.  
  
Колонке можно задавать одновременно различные классы со значениями для экранов, например class=«col-xs-12 col-md-8». Эти классы просто задают ширину колонке в процентах для определенного размера экрана. Если колонке не задан класс определенного экрана, то применится класс для минимально определенного экрана, а если и он не указан – то никакая ширина не применится и блок займет максимально возможную ширину.  
  
У нас классы «col-md-7» и «col-md-17» указывают, что блоки представляют из себя колонки шириной 7 и 17 относительно родительского контейнера. По умолчанию сумма ширин колонок в Bootstrap равняется 12, однако мы увеличили это число вдвое для достижения нужной нам гибкости. 

@grid-columns: 24;

Далее опишем нужные нам отступы:

body {

…

.wrapper {

padding: 0 0 50px 0;

}

header {

padding: 20px 0;

}

}

Данную конструкцию мы поместили внутрь body. Синтаксис LESS позволяет вкладывать правила друг в друга, которые потом скомпилируются в такие конструкции: 

body .wrapper {…}

body header {…}

Такой подход позволяет видеть структуру HTML прямо внутри CSS и дает некую «область видимости» правилам.

Логотип

https://habrastorage.org/getpro/habr/post_images/ad1/e15/d92/ad1e15d92086ee83c396cb458509df2d.png  
  
Вставляем логотип в тег header:

<header>

<a href="/"><img src="" alt="Whitesquare logo"></a>

</header>

Дополнительных стилей не требуется.

Поиск

https://habrastorage.org/getpro/habr/post_images/d87/d55/940/d87d5594013f6e7b6632e278ae34ef8c.png  
  
Для того чтобы создать поиск, нам понадобятся следующие компоненты Bootstrap: [инлайновая форма](http://getbootstrap.com/css/" \l "forms-inline), [сгруппированные контролы](http://getbootstrap.com/components/#input-groups) и [кнопка](http://getbootstrap.com/css/#buttons).  
В теге header создаем инлайновую форму, выровненную по правому краю. Поля такой формы обязательно должны иметь класс «form-control» и label.  
  
В форму мы помещаем компонент «сгруппированные контролы». Группировка контролов позволяет убрать отступ между вводом текста и кнопкой и как бы слить их в единый элемент.  
Он представляет из себя div с классом «input-group» и полями, а кнопка такого компонента помещается в блок с классом «input-group-btn».  
  
Поскольку нам не нужно показывать label для поля поиска — скроем его классом «sr-only». Это нужно для специальных устройств чтения с экрана.  
  
Кнопке добавляется класс «btn-primary», означающий, что это первичная кнопка данной формы. 

<header>

…

<form name="search" action="#" method="get" class="form-inline form-search pull-right">

<div class="input-group">

<label class="sr-only" for="searchInput">Search</label>

<input class="form-control" id="searchInput" type="text" name="search" placeholder="Search">

<div class="input-group-btn">

<button type="submit" class="btn btn-primary">GO</button>

</div>

</div>

</form>

</header>

Всё, что нам осталось — это задать в стилях ширину форме поиска.

body {

…

.wrapper {

…

header {

…

.form-search {

width: 200px;

}

}

}

}

Меню

https://habrastorage.org/getpro/habr/post_images/11f/1e2/bf4/11f1e2bf47e901fa1ffc3af899deefcf.png  
  
Для отображения меню возьмем компонент «[навигационная панель](http://getbootstrap.com/components/#navbar)» и поместим в него компонент «[навигация](http://getbootstrap.com/components/#nav)», которая представляет из себя список со ссылками. Для навигации добавляется класс «navbar-nav», который применяет специальные стили навигации внутри навигационной панели.

<nav class="navbar navbar-default">

<ul class="nav navbar-nav">

<li><a href="/home/">Home</a></li>

<li class="active"><a href="/about/">About us</a></li>

<li><a href="/services/">Services</a></li>

<li><a href="/partners/">Partners</a></li>

<li><a href="/customers/">Customers</a></li>

<li><a href="/projects/">Projects</a></li>

<li><a href="/careers/">Careers</a></li>

<li><a href="/contact/">Contact</a></li>

</ul>

</nav>

Для того, чтобы привести это меню к нашему дизайну зададим следующие значения переменным:

/\*высота навигационного меню\*/

@navbar-height: 37px;

/\*задаем побольше горизонтальный отступ\*/

@nav-link-padding: 10px 30px;

/\*фон для пунктов меню\*/

@navbar-default-bg: @panel-bg;

/\*цвет текста в пунктах меню\*/

@navbar-default-link-color: #b2b2b2;

/\*и при наведении мышки – такой же\*/

@navbar-default-link-hover-color: @navbar-default-link-color;

/\*фон активного пункта меню – наш специфический голубой цвет\*/

@navbar-default-link-active-bg: @brand-primary;

/\*цвет текста активного пункта меню\*/

@navbar-default-link-active-color: #fff;

Помимо настраиваемых параметров, опишем дополнительные в стилях – это текст в верхнем регистре и наш специфический шрифт:

body {

…

.wrapper {

…

.navbar a {

text-transform: uppercase;

font: 14px @brand-font;

}

}

}

Заголовок страницы

https://habrastorage.org/getpro/habr/post_images/728/e3b/83b/728e3b83b915a000db114952b08c9b4c.png  
  
Заголовок страницы помещается в div с классом «heading».

<div class="heading">

<h1>About us</h1>

</div>

И имеет следующие стили:

body {

…

.wrapper {

…

.heading {

height: 40px;

background: transparent url(../images/h1-bg.png);

margin: 30px 0;

padding-left: 20px;

h1 {

display: inline-block;

color: #7e7e7e;

font: normal 40px/40px 'Oswald', sans-serif;

background: url(../images/bg.png);

margin: 0;

padding: 0 10px;

text-transform: uppercase;

}

}

}

}

Здесь мы рисуем серую полоску фоном на div’e, и в нее вкладываем инлайновый h1 с нужным шрифтом и фоном цвета страницы, чтобы создалось впечатление прозрачного фона для h1.

Подменю

  
  
При создании подменю, мы не будем использовать компонент «навигация», так как по стилям он нам не очень подходит, гораздо больше нам подойдет компонент «[группированный список](http://getbootstrap.com/components/#list-group)». Каждый элемент такого компонента имеет класс «list-group-item».  
  
Подменю нужно разместить в теге aside. Список ссылок создаем аналогично главному меню.

<aside class="col-md-7">

<ul class="list-group submenu">

<li class="list-group-item active">Lorem ipsum</li>

<li class="list-group-item"><a href="/donec/">Donec tincidunt laoreet</a></li>

<li class="list-group-item"><a href="/vestibulum/">Vestibulum elit</a></li>

<li class="list-group-item"><a href="/etiam/">Etiam pharetra</a></li>

<li class="list-group-item"><a href="/phasellus/">Phasellus placerat</a></li>

<li class="list-group-item"><a href="/cras/">Cras et nisi vitae odio</a></li>

</ul>

</aside>

В настройках компонента укажем, что все группированные списки нужно показывать с фоном и рамкой компонента «панель»:

@list-group-bg: @panel-bg;

@list-group-border: @panel-inner-border;

И применяем к подменю следующие стили:

body {

…

.wrapper {

…

.submenu {

margin-bottom: 30px;

li {

display: list-item;

font: 300 14px @brand-font;

list-style-position: inside;

list-style-type: square;

padding: 10px;

text-transform: uppercase;

&.active {

color: @brand-primary;

}

a {

color: @text-color;

text-decoration: none;

&:hover {

color: @text-color;

}

}

}

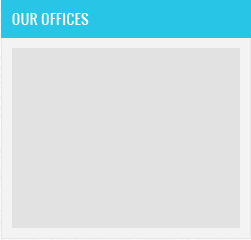
}

}

}

Сначала возвращаем элементам списка стандартные стили, так как Bootstrap их перебил на свои. Добавляем отступ снизу. Для подменю применяется более тонкий шрифт и квадратные маркеры. А для ссылок задаем цвета, верхний регистр и убираем подчеркивание. Амперсанд в коде «&.active» по синтаксису LESS во время компиляции заменится на родительский селектор: «.submenu li.active».

Контент сайдбара

В контенте сайдбара помимо подменю располагается также изображение с расположением офисов.  
  
  
  
Для его отображения нам подойдет компонент «[панель](http://getbootstrap.com/components/#panels)», а точнее его вариация «основная панель» (panel-primary) для раскраски заголовка. Этот компонент содержит блок заголовка (panel-heading) и блок содержания панели (panel-body). Изображению карты добавляем класс «img-responsive», который позволит картинке уменьшаться при маленькой ширине экрана.

<aside class="col-md-7">

…

<div class="panel panel-primary">

<div class="panel-heading">Our offices</div>

<div class="panel-body">

<img src="/images/map.png" class="img-responsive" alt="Our offices">

</div>

</div>

</aside>

В переменных Bootstrap мы уже задали цвет для фона панели (panel-bg), а теперь укажем, что «primary» панель будет иметь серую рамку дефолтной панели, а не голубую, как задано по умолчанию:

@panel-primary-border: @panel-inner-border;

Теперь в стилях сайта нужно изменить стандартные настройки панелей, которые не изменяются через переменные:

.panel {

box-shadow: none;

.panel-heading {

font: 14px @brand-font;

text-transform: uppercase;

padding: 10px;

}

.panel-body {

padding: 10px;

}

}

Здесь мы убрали тень у панелей, поставили свои отступы и задали свой шрифт заголовка.

Цитата

Вёрстку контента начнём с добавления цитаты.   
  
  
  
Этот элемент страницы больше всего похож на компонент [Jumbotron](http://getbootstrap.com/components/" \l "jumbotron). Добавим его в колонку контента:

<section class="col-md-17">

<div class="jumbotron">

<blockquote>

<p>

“Quisque in enim velit, at dignissim est. nulla ul corper, dolor ac pellentesque placerat, justo tellus gravida erat, vel porttitor libero erat.”

</p>

<small>John Doe, Lorem Ipsum</small>

</blockquote>

</div>

</section>

Через переменные для компонента jumbotron зададим белый цвет текста и брэндовый голубой фон:

@jumbotron-bg: @brand-primary;

@jumbotron-color: #fff;

И опишем наши стили:

body {

…

.wrapper {

…

.jumbotron {

border-radius: 0;

padding: 0;

margin: 0;

blockquote {

border-left: none;

p {

font: 300 italic 33px @brand-font;

text-transform: uppercase;

margin-bottom: 0;

}

small {

text-align: right;

color: #1D8EA6;

font: 300 20px @brand-font;

&:before {

content: '';

}

}

}

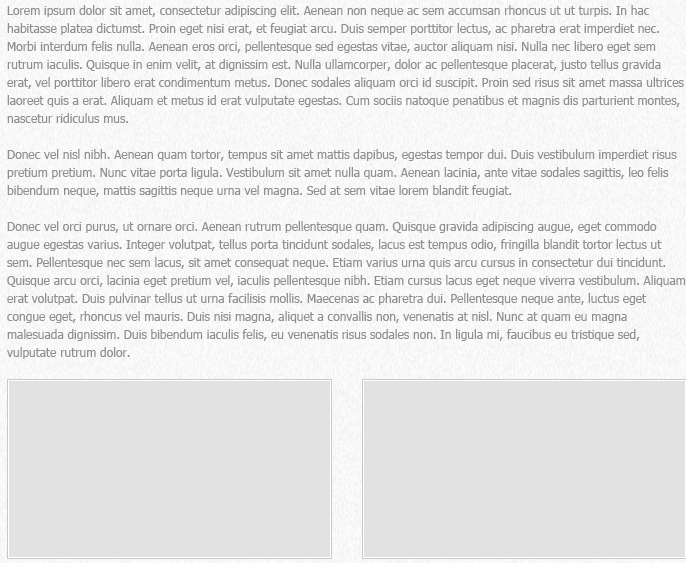
}

}

}

В них мы убираем скругление углов, отступы компонента и декорации цитаты, заданные Bootstrap по умолчанию. Также добавим стили наших шрифтов.

Контент

  
  
Все стили для текста контента мы уже добавили. Поэтому остается добавить только три параграфа с самим текстом.

<p>Lorem ipsum dolor sit amet…</p>

<p>Donec vel nisl nibh…</p>

<p>Donec vel nisl nibh…</p>

Следующим шагом нужно добавить два изображения, которые находятся в конце текста контента. Делается это с помощью двух колонок:

<div class="row">

<div class="col-md-12">

<img src="/images/about-1.png" alt="" class="thumbnail">

</div>

<div class="col-md-12">

<img src="/images/about-2.png" alt="" class="thumbnail">

</div>

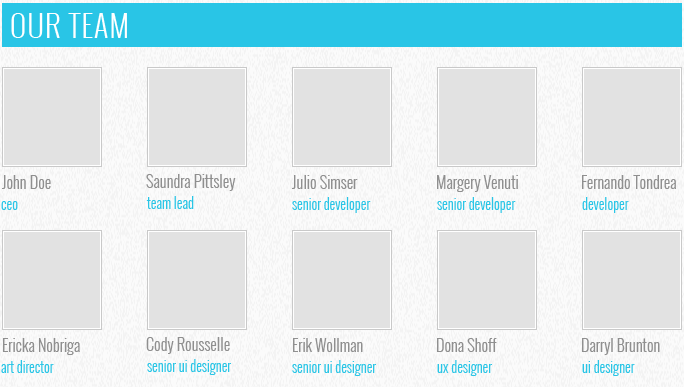
</div>

Класс «thumbnail» превращает изображения в компонент «[миниатюра](http://getbootstrap.com/components/#thumbnails)». Он сделает за нас всю работу по стилизации изображений. Единственное, что нам осталось — это установить свой отступ и цвет рамки в переменных для этого компонента:

@thumbnail-padding: 1px;

@thumbnail-border: #c9c9c9;

Блок «Our team»

  
  
При верстке этого блока добавим сначала заголовок:

<h2>Our team</h2>

со стилем:

body {

…

.wrapper {

…

h2 {

background: none repeat scroll 0 0 #29C5E6;

color: #fff;

font: 300 30px @brand-font;

padding: 0 10px;

text-transform: uppercase;

}

}

}

А затем добавим блок с классом «team», который состоит из двух строк, содержащих карточки сотрудников. Каждая карточка – это колонка. Карточка имеет ширину равную четырем колонкам нашей сетки. Все карточки кроме первой в строке имеют отступ слева, который создается классом «col-md-offset-1». Содержимое карточки состоит из изображения и описания (.caption)

<div class="team">

<div class="row">

<div class="col col-md-4">

<img src="/images/team/Doe.jpg" alt="John Doe" class="thumbnail">

<div class="caption">

<h3>John Doe</h3>

<p>ceo</p>

</div>

</div>

<div class="col col-md-4 col-md-offset-1">

<img src="/images/team/Pittsley.jpg" alt="Saundra Pittsley" class="thumbnail">

<div class="caption">

<h3>Saundra Pittsley</h3>

<p>team leader</p>

</div>

</div>

…

</div>

<div class="row">

<div class="col col-md-4">

<img src="/images/team/Nobriga.jpg" alt="Ericka Nobriga" class="thumbnail">

<div class="caption">

<h3>Ericka Nobriga</h3>

<p>art director</p>

</div>

</div>

<div class="col col-md-4 col-md-offset-1">

<img src="/images/team/Rousselle.jpg" alt="Cody Rousselle" class="thumbnail">

<div class="caption">

<h3>Cody Rousselle</h3>

<p>senior ui designer</p>

</div>

</div>

…

</div>

</div>

После создания разметки зададим этим элементам следующие стили:

body {

…

.wrapper {

…

.team {

.row {

margin-top: 20px;

.col {

white-space: nowrap;

.thumbnail {

margin-bottom: 5px;

}

}

.col-md-offset-1 {

margin-left: 3.7%;

}

.caption {

h3 {

font: 300 16px @brand-font;

margin: 0;

}

p {

font: 300 14px @brand-font;

color: @brand-primary;

margin: 0;

}

}

}

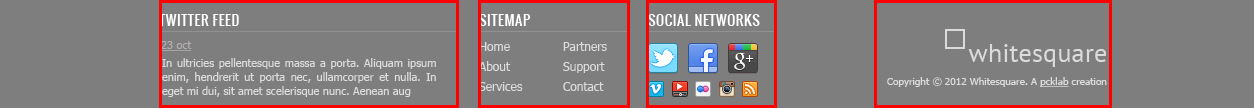
}

}

}

Помимо отступов и стилей шрифтов, которые тут задаются, мы изменили класс «col-md-offset-1». Ему пришлось задать отступ 3.7%, т.к. стандартный отступ был слишком большой.

Футер

Футер состоит из четырёх больших блоков: ленты Твиттера, карты сайта, социальных ссылок и логотипа с копирайтом.  
  
  
  
Для начала создадим контейнер футера с этим блоками:

<footer>

<div class="container">

<div class="row">

<div class="col-md-8 twitter"></div>

<div class="col-md-4 sitemap"></div>

<div class="col-md-6 social"></div>

<div class="col-md-6 footer-logo"></div>

</div>

</div>

</footer>

И применим к нему оформление: 

footer {

background: #7e7e7e;

color: #dbdbdb;

font-size: 11px;

.container {

height: 110px;

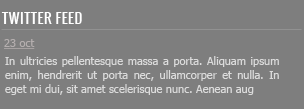
padding: 10px 0;

}

}

Тег footer задает серую область по всей ширине экрана, а контейнер внутри него отображает область по центру на больших экранах и задает высоту и отступ футера. Для выравнивания блоков внутри футера мы используем колонки.

Лента Твиттера

Верстаем содержимое ленты Твиттера:  
  


<div class="col-md-8 twitter">

<h3>Twitter feed</h3>

<time datetime="2012-10-23"><a href="#">23 oct</a></time>

<p>

In ultricies pellentesque massa a porta. Aliquam ipsum enim, hendrerit ut porta nec, ullamcorper et nulla. In eget mi dui, sit amet scelerisque nunc. Aenean aug

</p>

</div>

Стили:

body {

…

footer {

…

.container {

…

h3 {

border-bottom: 1px solid #919191;

color: #ffffff;

font-size: 14px;

line-height: 21px;

font-family: @brand-font;

margin: 0 0 10px;

text-transform: uppercase;

}

p {

margin: 5px 0;

}

.twitter {

p {

padding-right: 15px;

}

time a {

color: #b4aeae;

text-decoration: underline;

}

}

}

}

}

Для всех заголовков футера задаем шрифты и отступы, а также делаем подчеркивание через нижнюю рамку. Для параграфов указываем отступ. Ссылке, отображающую дату, задаем цвет и подчеркивание. 

Карта сайта

Карта сайта представляет собой две равные колонки со ссылками:  
  
 

<div class="col-md-4 sitemap">

<h3>Sitemap</h3>

<div class="row">

<div class="col-md-12">

<a href="/home/">Home</a>

<a href="/about/">About</a>

<a href="/services/">Services</a>

</div>

<div class="col-md-12">

<a href="/partners/">Partners</a>

<a href="/customers/">Support</a>

<a href="/contact/">Contact</a>

</div>

</div>

</div>

Ссылкам задаем цвет, шрифт и отступ между ними. 

body {

…

footer {

…

.container {

…

a {

color: #dbdbdb;

}

.sitemap a {

display: block;

font-size: 12px;

margin-bottom: 5px;

}

}

}

}

Социальные ссылки

  
  
Вставляем набор ссылок в блок с классом «social». 

<div class="col-md-4 social">

<h3>Social networks</h3>

<a href="http://twitter.com/" class="social-icon twitter"></a>

<a href="http://facebook.com/" class="social-icon facebook"></a>

<a href="http://plus.google.com/" class="social-icon google-plus"></a>

<a href="http://vimeo.com/" class="social-icon-small vimeo"></a>

<a href="http://youtube.com/" class="social-icon-small youtube"></a>

<a href="http://flickr.com/" class="social-icon-small flickr"></a>

<a href="http://instagram.com/" class="social-icon-small instagram"></a>

<a href="/rss/" class="social-icon-small rss"></a>

</div>

И стилизуем их:

body {

…

footer {

…

.container {

…

.social {

.social-icon {

width: 30px;

height: 30px;

background: url(../images/social.png) no-repeat;

display: inline-block;

margin-right: 10px;

}

.social-icon-small {

width: 16px;

height: 16px;

background: url(../images/social-small.png) no-repeat;

display: inline-block;

margin: 5px 6px 0 0;

}

.twitter { background-position: 0 0; }

.facebook { background-position: -30px 0; }

.google-plus { background-position: -60px 0; }

.vimeo { background-position: 0 0; }

.youtube { background-position: -16px 0; }

.flickr { background-position: -32px 0; }

.instagram { background-position: -48px 0; }

.rss { background-position: -64px 0; }

}

}

}

}

Здесь мы применили технику спрайтов – когда один файл с изображением применяется для разных картинок. Все ссылки разделились на большие иконки (.social-icon) и маленькие (.social-icon-small). Мы задали этим классам отображение в виде инлайнового блока с фиксированными размерами и одинаковым фоном. А затем с помощью CSS сдвинули этот фон так, чтобы на каждой ссылке отобразилось соответствующее изображение.

Копирайт

Блок с копирайтом и логотипом – это картинка со ссылкой и параграф с текстом под ней.  
  


<div class="col-md-8 footer-logo">

<a href="/"><img src="" alt="Whitesquare logo"></a>

<p>

Copyright © 2012 Whitesquare. A <a href="http://pcklab.com">pcklab</a> creation

</p>

</div>

Стили делают аналогично предыдущим блокам с той лишь разницей, что блок прибивается к правому краю и выравнивание внутри него так же по правому краю:

body {

…

.footer {

…

.container {

…

.footer-logo {

float: right;

margin-top: 20px;

font-size: 10px;

text-align: right;

a {

text-decoration: underline;

}

}

}

}

}