**Анализ макета Jetro**

****

Макет требуется сделать адаптивным и отображаемым хорошо на широком диапазоне устройств – от широкоформатных дисплеев до мобильных телефонов.

Исходя из этого, мы воспользуемся фреймворком **Boostrap 3** (далее BS), который имеет «на борту» адаптивную сетку, по которой можно удобно располагать все элементы.

Определим ширину макета – она составляет **960 пикселей**. Это его максимальная ширина. И это будет макет для брейкопоинта большого десктопа (large desktop), так как мы будем использовать BS. Классы колонок для различных устройств можно посмотреть в [этой таблице](http://getbootstrap.com/css/#grid-options).

Также, благодаря наличию в макете сетки, мы можем установить ширину BS-переменной **$gutter**, которая отвечает за **padding’**и слева и справа от стандартной BS-колонки и некоторых других элементов.

В макете используется один кастомный шрифт, который прилагается, и нужно будет его «прикрутить» через директиву **@font-face**.

Так как есть много однотипных элементов – размеров шрифтов, цветов, отступов и так далее, мы будем использовать препроцессор (в нашем случае **Less**) для упрощения работы. Учитывая, что требуется сделать макет адаптивным, **CSS-препроцессор** нам будет очень кстати. Без него работа увеличится в два и более раз.

Перейдем к разбору отдельных секций.

**1) Логотип**

Логотип будет в виде текста, так как нет смысла делать его картинкой. Для всего текста зададим оранжевый цвета и поместим в **<h1>**, а для слога RO укажем бледно-розовый цвет и этот слог поместим в тег **<span>**.

**2) Главное меню**

Оно будет представлять из себя обычный список **UL>LI >A**. Всем элементам будет задан серый цвет, а **active**- и **hover**- классы будут иметь оранжевый цвет.

Так как мы будем использовать BS, то два этих элемента (1 и 2) мы поместим в один контейнер **div.container**, который имеет фиксированную ширину (про типы контейнеров BS [здесь](http://getbootstrap.com/css/#overview-container)).

Затем разобъем его на колонки **div.col-xx-x** (ссылка на таблицу с классами колонок доступна выше) для размещения там логотипа и главного меню.

**3) Слайдер и**

**4) его фоновая полоска**

В качестве основы для слайдера нам сгодится стандартный слайдер BS (<http://getbootstrap.com/javascript/#carousel>). Нужно будет только немного переписать его HTML-разметку. Нам нужно будет стилизовать подпись к каждому слайду, настроить внешний вид кнопок листания влево и вправо, а также превратить точки, ссылающиеся на слайды, в превьюшки этих слайдов.

Касательно фоновой полоски, то в качестве общего контейнера для элементов 3) и 4) мы будем использовать **div.container-fluid**, который растягивается на всю ширину родительского элемента.

Внутри него мы разместим еще один **div** и залём его цветом этой полоски. Сверху наложим **div.container**, внутрь которого и поместим наш слайдер. Таким образом, цель будет достигнута – полоска всегда будет растянута на всю ширину экрана, а слайдер будет в любой момент времени центрирован или растянут на всю ширину области просмотра, если речь идет о мобильной версии.

**5) Кнопки перемщения влево-вправо и**

**6) описание слайда**

Нам нужно будет убрать стандартное «бутстраповское» затенение при наведении на конпки, а также подрегулировать их размер.

Слайдер с описанием будет спозиционирован абсолютно и иметь фоновый цвет, заданный через модель **RGBA**, что позволит сделать полупрозрачным фон родителя, не затронув параметр прозрачности для дочерних элементов – заголовка и, собственно, описания слайда.

**7) Превьюшки слайдов**

Мы используем разметку внутри слайдера BS для маркеров слайдов. В качестве фонового изображения мы зададим им картинки превьюшек наших слайдов.

Также, нам прийдется изменить позиционирование этой части слайдера. Взглянем на код стандартного BS-слайдера (<http://getbootstrap.com/javascript/#carousel>).

Мы видим, что нумерованный список **ol. carousel-indicators** имеет абсолютное позиционирование и расположен в самом верху. Мы спустим его вниз, под **div.carousel-inner** и зададим этому списку позиционирование **position:static;** заставив, таким образом, его находиться под секцией с полноразмерными изображениями. Таким образом, мы достигаем нашей цели и останется лишь подставить ссылки на изображения внутрь маркеров, не забывая правильно все это спозиционировать.

Изображения для превьюшек и полноформатного просмотра извлечем в формате JPEG.

**Секции статей (элементы 8, 9, 10)**

Каждая секция будет обернута в HTML5-тег **<article>**, что позволит поисковикам рассматривать эти части как отдельные от общей семантики страницы. Хотя логичнее будет все-таки использовать тег **<section>,** который делит контейнер на части, которые по смыслу нельзя вытянуть из контекста и использовать как самостоятельную единицу. Это логичнее еще и потому, что превью статей нельзя считать законченным смысловым элементом.

**8) Иконки статей и заголовки**

Будут иметь формат **PNG**, так как нам нужна прозрачность вокруг иконок (альфа-канал) и нам НЕ предоставили иконки в векторном формате. Можно преобразовать все три иконки в **SVG** (способы это сделать мы рассматривали в видео Дня 2), но из-за того, что качество таких преобразованных иконок будет невысоким, то мы откажемся от этой идеи, либо рассмотрим вариант её реализации на последнем этапе создания веб-страницы.

Заголовок будет иметь тег **<h1>**, потому как благодаря использованию нового алгоритма **HTML5 Outline**, заголовков первого уровня у нас на странице может быть более одного.

**9) Вводный текст статьи и**

**10) кнопка More..**

Вводный текст поместим в обычный параграф **<p>**, а кнопку с текстом “**More..**” – в тег **<a>**, который мы соответствующим образом стилизуем.

**11) Заголовок Recent Articles и 12) фоновая текстура рядом с ним**

Заголовок будет задан тегом **<h1>**. Мы поместим его в контейнер **div** и, задав ширину по размеру текста сделаем фон фелого цвета. В то же время, на фон контейнера-обертки мы поместим текстуру (кусочек, вырезанный из слоя с узором), и замостим его ею по горизонтали.

**13) Картинка и 12) текстовая подпись к ней**

Картинку вырежем в формате **JPEG**, а текстовую подпись изобразим в виде заголовка **<h2>** и параграфа **<p>**.

**15-18) Футер и 19) фоновая розовая линия**

Эти элементы мы сделаем по тому же принципу, как и слайдер с его фоновой линией, но тут будет проще – мы просто поместим **div.container** внутрь **div.container-fluid**, задав последнему розовый фон. Таким образом, футер окажется отцентрированным.

Блоки 15, 16 и 17 внутри оранжевого блока футера мы зададим с помощью колонок BS. Текст в них будет в теге **<p>**, а заголовок в **<h2>**.

Иконки 18 мы преобразуем в код **base64** и вставим непосредственно в **CSS-код**. Это позволит сэкономить на количестве **HTTP-запросов** к серверу при попытке пользователя открыть наш сайт.

**Для того, чтобы дополнить картину, описанную в этом документе, еще раз посмотри секцию видеозаписи Дня 2, где я анализирую этот макет.**