***Carousel* –** блок в котором выводятся какие либо данные (контент сайта, реклама, фотографии или любая возможная информация). Основной сутью которого есть взаимозаменяемость блоков по очередном порядке с использованием выбранного интервала времени в автономном режиме а также возможностью ручной пользовательской навигации(серфингу) по элементам(блокам) самой карусели как в перед так и назад.

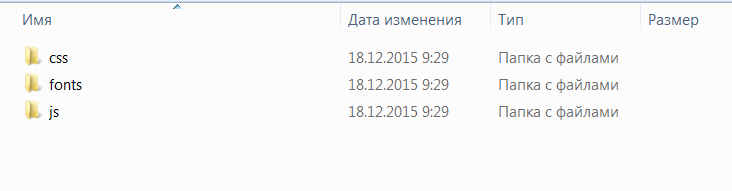
В данном случае мы будем использовать библиотеку Bootstrap для создания «карусели». Для этого нам нужно скачать Bootstrap и еще несколько библиотек.

***1. Скачиваем библиотеки***

Скачиваем архив с последней версией библиотеки, актуально использовать последнюю версию так как постоянно выходят обновления и улучшается функционал и

поддержка на разных устройствах и браузерах

Bootstrap - http://mybootstrap.ru/

После скачивания распаковываем архив. 

После распаковки архива у нас есть три папки:

**CSS** – папка в которой хранятся стили оформления страницы   
**FONTS** – шрифты используемые библиотекой, они хранятся в отдельной папке так как на устройстве пользователя может не быть шрифтов которые используются на странице   
**JS** – скрипты используемые для нашего модального окна и многого другого что используются в библиотеке Bootstrap.

Также в нашем случаем мы тут же создали папку img куда добавили три картинки. Которые будут выводится в блоках как айтемы <div class=**"item"**><img src=**"img/3.jpg"**></div></div>.

***2. Скачав и распаковав библиотеку у нас есть весь функционал нужный для реализации работы.***

Теперь создаем HTML страничку.

<!DOCTYPE html>  
<html>  
<head lang=**"en"**>  
 <meta charset=**"UTF-8"**>  
 <title>**Carousel**</title>  
  
 *<!-- Bootstrap core CSS -->* <link href=**"css/bootstrap.min.css"** rel=**"stylesheet"**>  
  
 *<!-- Custom styles for this page -->* <link href=**"css/style.css"** rel=**"stylesheet"**>  
</head>  
<body>  
*<!-- div main -->*<div class=**"content"**>  
  
</div>  
  
  
*<!-- Bootstrap core JavaScript  
================================================== -->  
<!-- Placed at the end of the document so the pages load faster -->*<script src=**"js/jquery-1.11.3.min.js"**></script>  
<script src=**"js/bootstrap.min.js"**></script>  
*<!-- Custom script for this page -->*<script src=**"js/main.js"**></script>  
</body>  
</html>

Организация кода на страничке такова что при ее запуске ***обязательно* должны** грузится стили и скрипты библиотеки, а потом ваши собственные. Потому из написанного выше кода видно, что вначале выполняется загрузка сначала стилей библиотеки потом собственные стили чтоб выполнялось правило наследования стилей и ваши стили меняли стили которые были написаны в библиотеке по наследованию. После чего в мы также подключаем наши скрипты. В это делается в конце <body> чтоб скрипты загружались после загрузки самого HTML контента станицы.

***3. После чего переходим к написанию HTML самой «карусели»***

*<!-- div main -->*<div class=**"content"**>  
 *<!-- Carousel -->* <div id=**"myCarousel"** class=**"carousel slide"**>  
 *<!-- Carousel items -->* <div class=**"carousel-inner"**>  
 <div class=**"active item"**><img src=**"img/1.jpg"**></div>  
 <div class=**"item"**><img src=**"img/2.jpg"**></div>  
 <div class=**"item"**><img src=**"img/3.jpg"**></div>  
 </div>  
 *<!-- Carousel nav -->* <a class=**"carousel-control left"** href=**"#myCarousel"** data-slide=**"prev"**>**&lsaquo;**</a>  
 <a class=**"carousel-control right"** href=**"#myCarousel"** data-slide=**"next"**>**&rsaquo;**</a>  
 </div>  
  
</div>

Этот пример нам показывает работу карусели и ее создание. Как она работает мы создаем общий блок <div class=**"content"**> класс **content** в означает что этот блок есть одним общим и не делится на колонки и т.п. В котором создаем сам блок с каруселью <div id=**"myCarousel"** class=**"carousel slide"**>  
 на сам общий блок с каруселью навешены общие стили 1 карусели и айди поторому в будущем будут подключатся скрипты управления, но об том дальше. Она в свою очередь может иметь множество дочерних блоков <div class=**"item"**> которые которые обязательно должны иметь клас class=**"item"** и один из элементов class=**"active item"**, содержащих какую-либо информацию которую нужно вывести, в нашем случае есть три картинки **"**><img src=**"img/2.jpg"**> картинка имеющая class=**"active"**  будет показываться стартовой при загрузке страницы.

К общему блоку <div id=**"myCarousel"** class=**"carousel slide"**> прописаны стили которые есть по умолчанию потому блок будет выводится на весь екран.

***4. При использовании без JavaScript части.***

«Карусель» в случае если под написан без *JavaScript* блоки будут выведены, но не будет работать автоматическая смена элементов так как их функционал сделан и написан на *JavaScript.* Но возможна ручная навигация при помощи кнопок на странице.

Итак перейдем к части кода который должен быть написан на скрипте *JavaScript.*

***5. С помощью JavaScript.***

Вызываем функцию carousel() на объекте DOM $('.carousel' которая будет автоматически перебирать картинки с интервалом по умолчанию 5000милисекунт.

$('.carousel').carousel();

### **Опции**

Опции могут быть заданы через data-атрибуты или JavaScript. Для data-атрибутов добавьте название опции к data-, как и к data-interval="".

| **Название** | **тип** | **по умолчанию** | **описание** |
| --- | --- | --- | --- |
| interval | number | 5000 | Количество времени между автоматической сменой слайдов. Если установить значение false, карусель не будет запущена автоматически. |
| pause | string | «hover» | Ставит показ слайдов на паузу при фокусе «карусели» и возобновляет показ при потере фокуса. |

### **Методы**

.carousel(options);  
  
*//Инициализирует «карусель» с дополнительными опциями object и начинает показ слайдов.*$('.carousel').carousel({  
 interval: 2000  
})  
  
 .carousel('cycle')  
  
*//Запускает цикл слева направо.* .carousel('pause')  
  
*//Останавливает показ элементов.* .carousel(*number*)  
  
*//Переход на указанный элемент (Начиная с 0, по аналогии с массивом).* .carousel('prev')  
  
*//Переход к предыдущему элементу.* .carousel('next');  
  
*//Переход к следующему элементу.*

В этом случае в отлии чии от варианта с $('.carousel').carousel(); в этом случае функция carousel() может принимать различные параметры и отрабатывать по разному самый частый вариант это когда нужно засетать другое тайминговое значение интервала смены айтемов.

$('.carousel').carousel({  
 interval: 2000  
})

В этом случае мы водим ключ interval: 2000 и заносим в него нужное значение по принципу работы с *JavaScript* объектами.

Остальные методы нам дают удобство работы или навигации по элементам в случае если нам нужно ходить по ним в каком-то порядке.

### **События**

«Карусель» Bootstrap имеет два события:

| **Событие** | **Описание** |
| --- | --- |
| slide | Срабатывает каждый раз при вызове метода slide. |
| slid | Срабатывает, когда «карусель» закончила показ слайдов. |