<!--

以下容器就是整个轮播图组件的整体，

注意该盒子必须加上 class="carousel slide" data-ride="carousel" 表示当前是一个轮播图

bootstrap.js会自动为当前元素添加图片轮播的特效

-->

<div id="轮播图的ID" class="carousel slide" data-ride="carousel">

<!-- ol标签是图片轮播的控制点 -->

<ol class="carousel-indicators">

<!--

每一个li就是一个单独的控制点

data-target属性就是指定当前控制点控制的是哪一个轮播图，其目的是如果界面上有多个轮播图，便于区分到底控制哪一个

data-slide-to属性是指当前的li元素绑定的是第几个轮播项

注意，默认必须给其中某个li加上active，展示的时候就是焦点项目

-->

<li data-target="#轮播图的ID" data-slide-to="0" class="active"></li>

<li data-target="#轮播图的ID" data-slide-to="1"></li>

<!-- ...更多的 -->

</ol>

<!--

.carousel-inner是所有轮播项的容器盒子，

注意role="listbox"代表当前div是一个列表盒子，作用就是给当前div添加一个语义

-->

<div class="carousel-inner" role="listbox">

<!-- 每一个.item就是单个轮播项目，注意默认要给第一个轮播项目加上active，表示为焦点 -->

<div class="item active">

<!-- 轮播项目中展示的图片 -->

<img src="example.jpg" alt="示例图片">

<div class="carousel-caption">

<!-- 标题或说明性文字，如果不需要，直接删除当前div.carousel-caption -->

</div>

</div>

<div class="item">

<!-- ... -->

</div>

<!-- ... -->

</div>

<!-- 图片轮播上左右两个控制按钮，分别点击可以滚动到上一张和下一张 -->

<!-- 此处需要注意的是 该a链接的href属性必须指向需要控制的轮播图ID -->

<!-- 另外a链接中的data-slide="prev"代表点击该链接会滚到上一张，如果设置为next的话则相反 -->

<a class="left carousel-control" href="#轮播图的ID" role="button" data-slide="prev">

<span class="glyphicon glyphicon-chevron-left" aria-hidden="true"></span>

<span class="sr-only">上一张</span>

</a>

<a class="right carousel-control" href="#轮播图的ID" role="button" data-slide="next">

<span class="glyphicon glyphicon-chevron-right" aria-hidden="true"></span>

<span class="sr-only">下一张</span>

</a>

</div>

#### 由于轮播图片超宽造成的影响

- 美工为了在不同屏幕下更好地展示将图片两边做的非常宽，但是我们大多数情况的页面宽度都无法满足这样的图片宽度

- 而且Bootstrap的样式中默认将图片的max-width设置为100%；

- 造成界面图片缩放

- 想在一个较小屏幕下展示一个超宽的图片，并让图片居中显示

+ 两种办法：

\* 换用背景图的方式，background-position: center center;

\* 使img元素绝对定位，left:50%，margin-left: -width/2

### background使用

- 将容器的高度固定（410px）

- 将轮播图改为背景显示

- 由于可能图片的高度不一定是410px

- 所以需要设置css3中的background-size

#### background-size

- length

+ 如 background-size: 100px 100px，将背景图固定到多大尺寸

- percentage

+ 如 background-size: 90% 90%，以百分比的形式设置背景大小

- cover

+ 1.背景图片等比例缩放

+ 2.让背景图相对较小边放大到目标容器大小结束

\* 如：一张100\\*200的背景图放到一个300\\*400的盒子中，最终背景图片的大小是300\\*600

\* 因为背景图的较小边为100，将100放大到目标容器300的宽度，放大了3倍，最终结果300\\*600

- contain

+ 1.背景图片等比例缩放

+ 2.让背景图相对较大边放大到目标容器大小结束

\* 如：一张100\\*200的背景图放到一个300\\*400的盒子中，最终背景图片的大小是200\\*400

\* 因为背景图的较大边为200，将200放大到目标容器400的高度，放大了2倍，最终结果200\\*400

##### demo

###### cover

<div style="width: 400px;height: 200px;border:1px dashed #c0c0c0;background-image: url('data:image/svg+xml;base64,');background-repeat:no-repeat;background-size: cover;"></div>

###### contain

<div style="width: 400px;height: 200px;border:1px dashed #c0c0c0;background-image: url('data:image/svg+xml;base64,');background-repeat:no-repeat;background-size: contain;"></div>

### 图片响应式

- 目的

+ 各种终端都需要正常显示图片

+ 移动端应该使用更小（体积）的图片

- 实现方式

+ 将元素中直接设置的图片背景删除，换成两个data-属性（如：data-img-sm="小图路径"，data-img-lg="大图路径"）

+ 通过JS的方式获取屏幕的宽度；

+ 判断屏幕宽度是否小于一定的值（如：768）

+ 根据判断情况决定使用具体的大图还是小图

javascript

// 获取屏幕宽度

var windowWidth = $(window).width();

// 判断屏幕属于大还是小

var isSmallScreen = windowWidth < 768;

// 根据大小为界面上的每一张轮播图设置背景

// $('#main\_ad > .carousel-inner > .item') // 获取到的是一个DOM数组（多个元素）

$('#main\_ad > .carousel-inner > .item').each(function(i, item) {

// 因为拿到是DOM对象 需要转换

var $item = $(item);

// var imgSrc = $item.data(isSmallScreen ? 'image-xs' : 'image-lg');

var imgSrc =

isSmallScreen ? $item.data('image-xs') : $item.data('image-lg');

// 设置背景图片

$item.css('backgroundImage', 'url("' + imgSrc + '")');

});