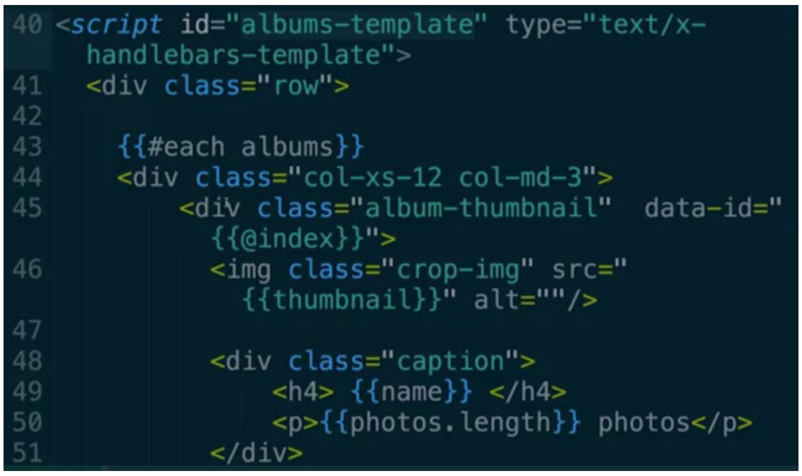
2.4.1.6: Writing the templates for the gallery

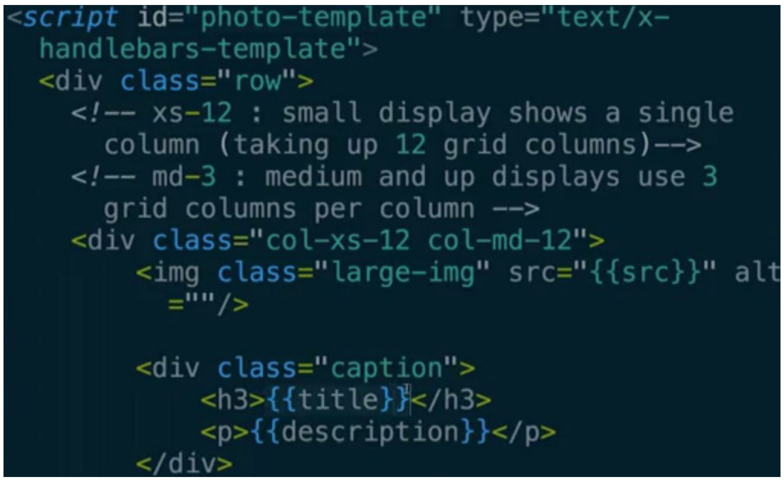
In the last video I showed you how we can create quite a complex structure to represent an entire photo album site. We've got albums, albums contain photos, and photos have data associated with them. Now, I'm going to look at how we can use templates to represent that. So, in this video I'm gonna look specifically at the templates, and in the next video I'll show you how to link the data and the templates together with some JavaScript. So, we've got, we've split up our app, as I showed you before. I've already showed you albums.js which contains all the data. In the next video I'm gonna show you gallery.js which maps data onto templates.

But let's take have look at index.html, which contains the actual templates themselves. The templates we're gonna use in this example are actually pretty simple. They're not so different from the ones you have seen before, we've just have a number of different ones. And this is the first one, the albums template If we look at gallery object, the first thing we've got is this member variable albums, which is a bigger way of photo albums, and we're just using an each template expression to display all of these.



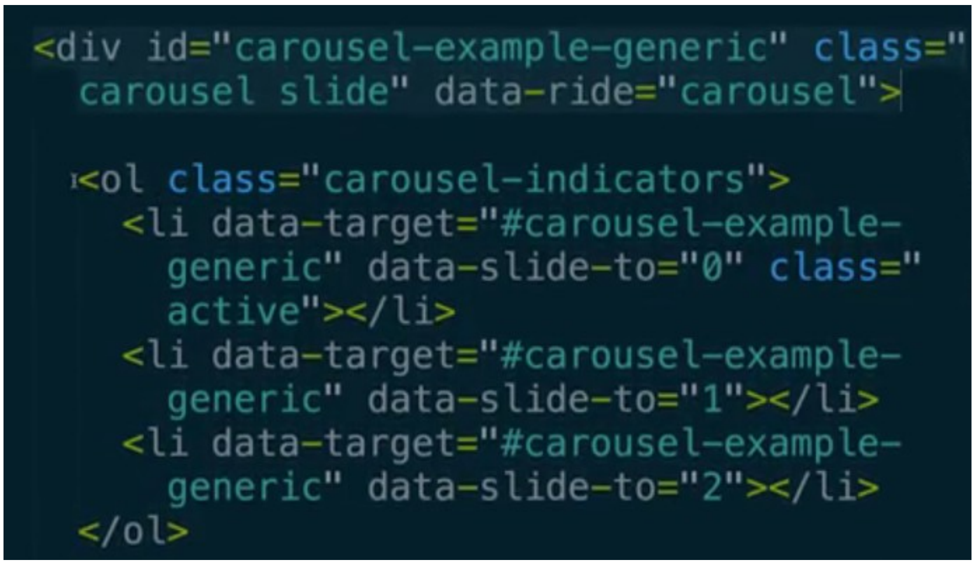
```
<script id="albums-template" type="text/x-
  handlebars-template">
  <div class="row">

    {{#each albums}}
    <div class="col-xs-12 col-md-3">
        <div class="album-thumbnail"  data-id="
          {{@index}}">
          <img class="crop-img" src="
            {{thumbnail}}" alt=""/>

          <div class="caption">
              <h4> {{name}} </h4>
              <p>{{photos.length}} photos</p>
          </div>
```

And we're going through with all of these. We've got, in this line here, we've got our data ID attribute, which is our index, which we saw just a couple of videos ago, will enable us to link, click on this thumbnail and bring up the whole album. We're displaying, our album contains this thumbnail variable, which is the source of an image we want to display to upsend that album, and we're just putting that in the image tag here. And then, we're displaying the name with the album. And this is just one other thing that we're using I wanted to show you. We're displaying the number of photos in it. So, in this case, photos is member variable of album. We can see it here. But we're not using array. What we want to do is, later on we're going to display the whole array, but actually what we want to just know is how many images there are. And this photos.length will give us the number of images, the size of the array, so that we can tell us how many photos there are. And we've got a number of other templates like this. We've got a photos template. Very, very similar. Basically, the same template we were using in our previous examples. We're gonna display this once you've clicked on an album, and we can see those photos in those albums. Again, it's got an each, image source, title, and description.



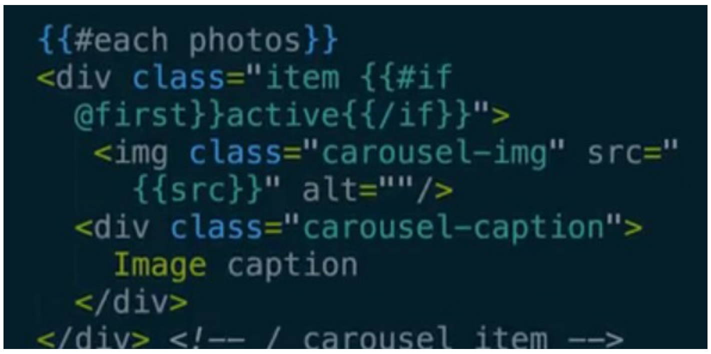
```
<script id="photo-template" type="text/x-
  handlebars-template">
  <div class="row">
    <!-- xs-12 : small display shows a single
      column (taking up 12 grid columns)-->
    <!-- md-3 : medium and up displays use 3
      grid columns per column -->
    <div class="col-xs-12 col-md-12">
        <img class="large-img" src="{{src}}" alt
          =""/>

        <div class="caption">
            <h3>{{title}}</h3>
            <p>{{description}}</p>
        </div>
```

And the next template is a photo template. And this an individual photo. So, this is the kind of template we were using last week. No need for each, or anything like that. It's just title, description, source for this one individual photo.



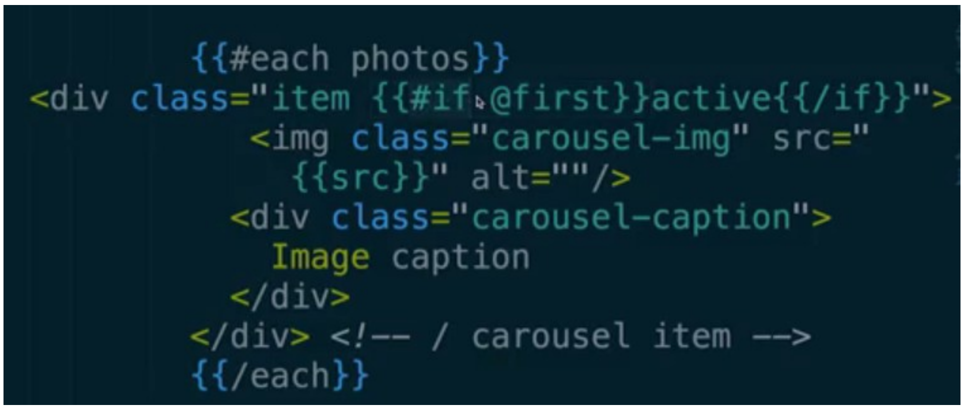
```
<div id="carousel-example-generic" class="
  carousel slide" data-ride="carousel">

  <ol class="carousel-indicators">
    <li data-target="#carousel-example-
      generic" data-slide-to="0" class="
      active"></li>
    <li data-target="#carousel-example-
      generic" data-slide-to="1"></li>
    <li data-target="#carousel-example-
      generic" data-slide-to="2"></li>
  </ol>
```

The last template we're gonna use is a template for the slideshow. Now, this is, in the first MOOC I showed you how to make a simple slideshow. In this example I'm gonna use built in jQuery and Bootstrap to create a more complex slideshow that has nice fading and various things like that.



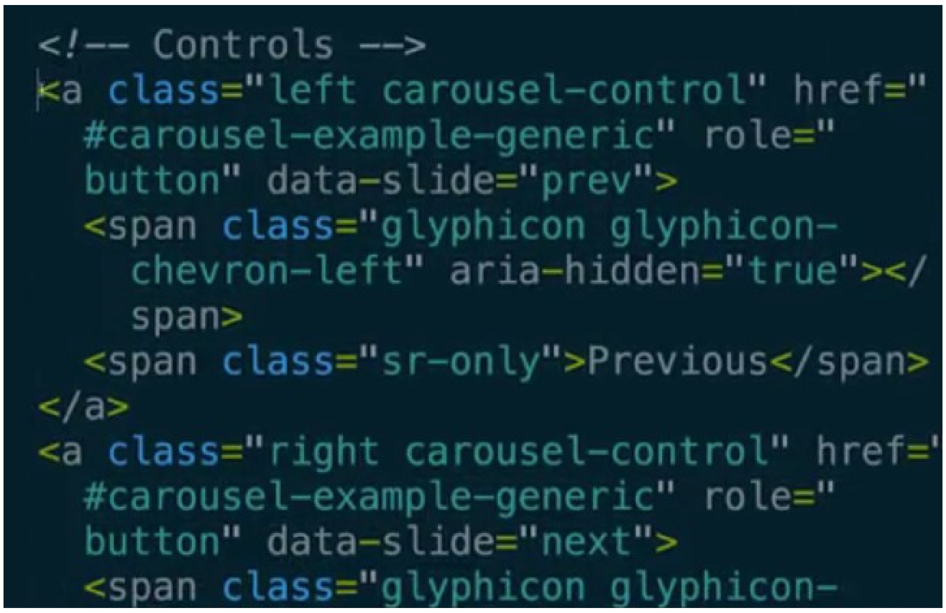
```
{{#each photos}}
<div class="item {{#if
  @first}}active{{/if}}">
   <img class="carousel-img" src="
     {{src}}" alt=""/>
  <div class="carousel-caption">
    Image caption
  </div>
</div> <!-- / carousel item -->
```

The important thing is how you get the actual photo data into this slideshow carousel. And again, we're using each template var expressions. And this is the little bit of code that tells us exactly how to display an image. It's got an image tag, and a div with a caption, and each of these divs with class item is just added to the carousel and displayed. So, if you want lots of them we just use an each expression as usual. The only one thing I will say is that I'm doing something a little bit special here. So, the standard class for this is item (**div class="item {{...**) And the class item means that it's a member of the slideshow, it's an item of the slideshow, and be displayed in the slideshow. But we also need to know which is the current showing, becoming the active item on the slideshow. And this jQuery carousel code uses an extra class, called active, for that particular item that's currently displaying. So, to display an item in the slideshow we need to set the active class. But how do I know which one is set active if I'm in a template and I'm creating all of them? I need to know, for example, if I want to start on the first one, I need to know which, when I am creating template for the first one, so I can add this active class to that one, and to just the first item that I'm displaying and not all the others. And this is how I'm doing it.



```
        {{#each photos}}
<div class="item {{#if @first}}active{{/if}}">
            <img class="carousel-img" src="
              {{src}}" alt=""/>
           <div class="carousel-caption">
             Image caption
           </div>
        </div> <!-- / carousel item -->
        {{/each}}
```

Let me move it over so we can see the entire line a bit better. I'm using an if template expression. So, we've seen if statements in JavaScript, this is kind of the same thing. So, I'm saying if something is true, then I'm doing this, otherwise do nothing. So, if this is true, at first, display active. What is at first? At first is true only if the current template item we're looking at in the template is the first item in the array we're displaying. So, that tells us exactly the right information. So, if we're displaying the first element in the array, then do active and the class becomes item active. Finish if. Otherwise, we're not gonna display active, and it's just an ordinary item. So that's just a special little bit of code, shows you a little bit more Handlebar templating.



```
<!-- Controls -->
<a class="left carousel-control" href="
  #carousel-example-generic" role="
  button" data-slide="prev">
  <span class="glyphicon glyphicon-
    chevron-left" aria-hidden="true"></
    span>
  <span class="sr-only">Previous</span>
</a>
<a class="right carousel-control" href='
  #carousel-example-generic" role="
  button" data-slide="next">
  <span class="glyphicon glyphicon-
```

And then, the rest of the template is some controls that enables the slideshow to do what it is. Index HTML, therefore, just contains a number of templates that we can use in different contexts to display different aspects of the data that I've just shown you in our albums.js.