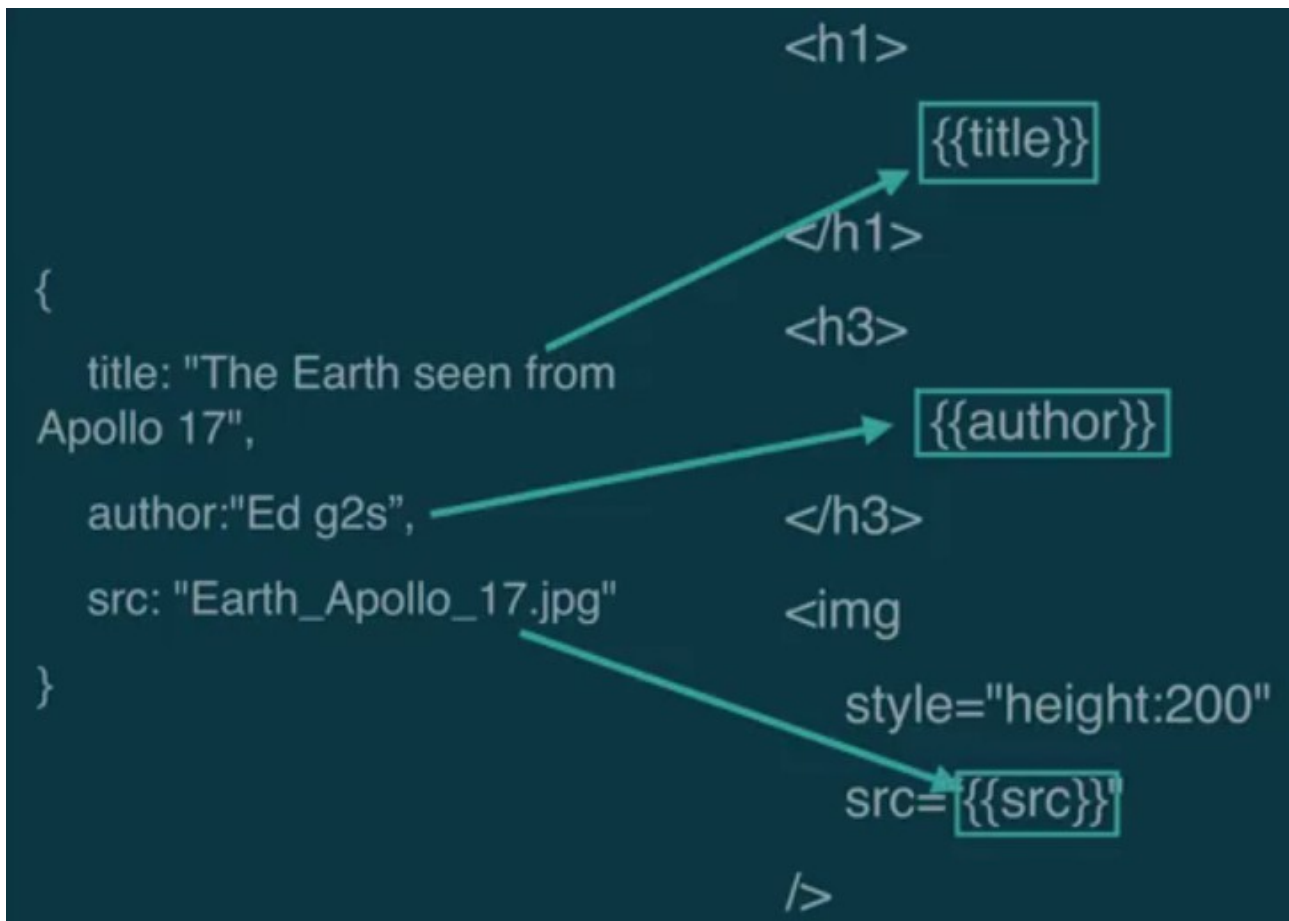


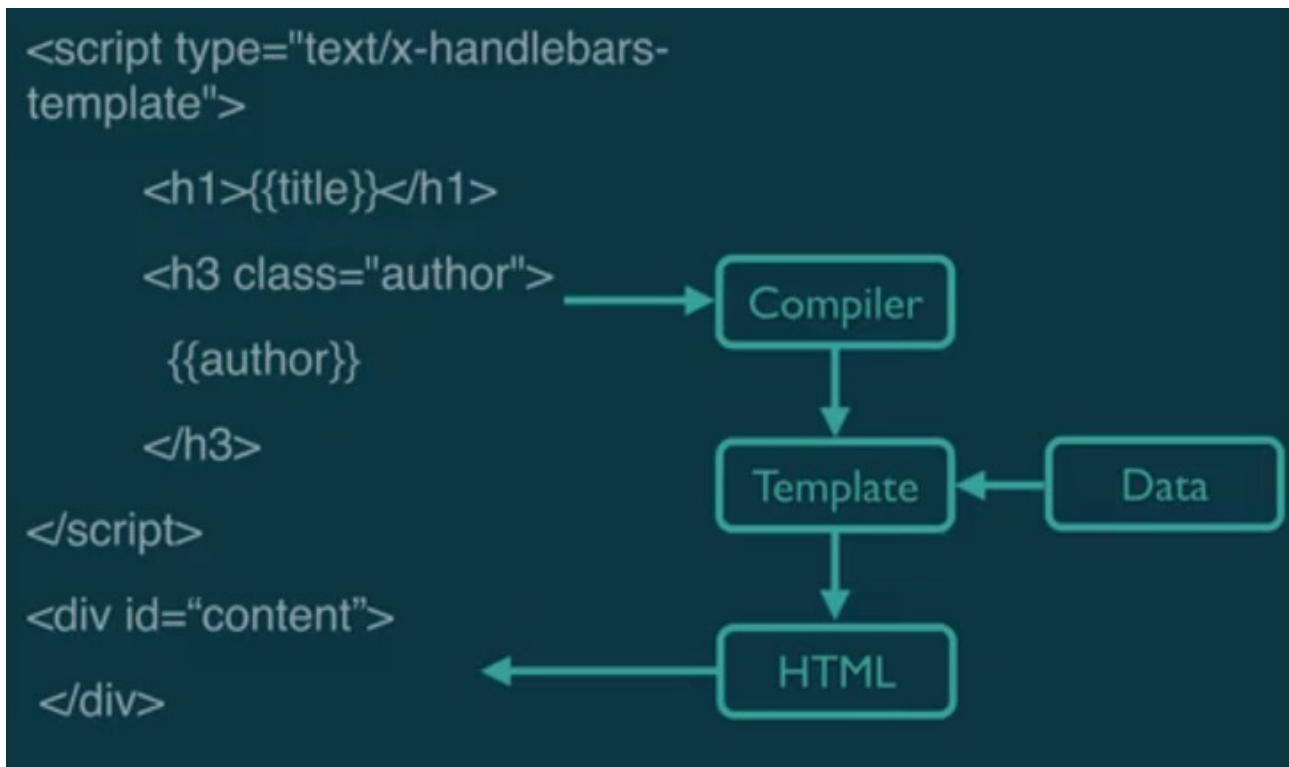
2.3.1.4: Putting data into templates

Now we've found out about objects and templates, the really powerful techniques for separating content and structure. And these are going to enable you to make some really interesting complex web pages. But there's one last step we need to do before we can use this, which is actually put the two of those things together and find out how to do that. And the way we do that is using JavaScripts.

So let's recap a minute what we're doing. We're taking variables from an object and putting them inside a template onto the template tags.

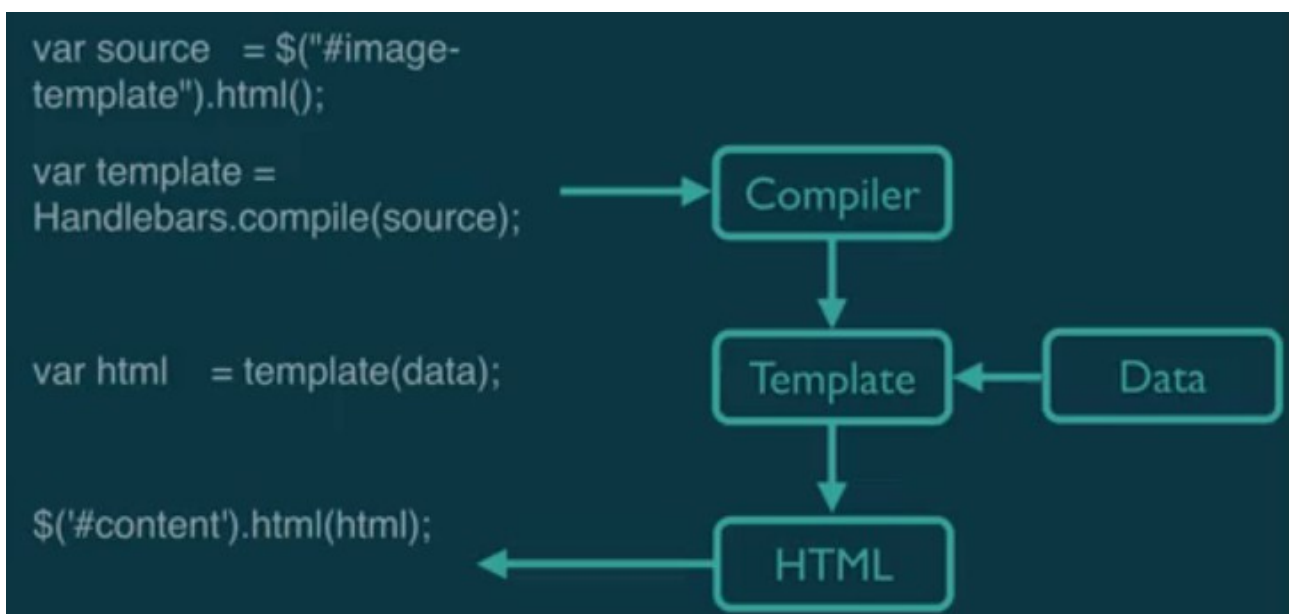


Now, in order to do that, we first need to do a bunch of stuff to the template. As I said earlier the template is not, JavaScript is not built in, it's in the cisternal library handlebars. So we need to do a couple of special things to turn it into some JavaScript we can actually use so.



- If we look at the template, the first thing we need to do is grab out the actual text of the template from the script tag, and compile it, put it into the compiler. What that means is you're taking this mixture of HTML and these weird double-curly bracket things, and turning them into some JavaScript code that can actually implement the template.
- The output of that is what we call the real template, which is JavaScript code.
- Once we've got our JavaScript code, we can put some data into there.
- That data is coming from the object and it gives us some HTML. And we can put that HTML back into the DOM inside a div like this.

So how do we do that? Let's look at the actual code we would use to do all that.



- So the first thing we need to do is get the text of the template out of the DOM, so we can just jQuery to do that. Grabbing the HTML out of the script tag it was in.
- Then we compile the template. So we call `handlebars.compile`, that's a function. We parse in the source, which is the text of the template, and we get out this variable template, which is this JavaScript code, which will actually turn objects into HTML.
- Then the next step is to take the data, which is your object, put it through the template. So that's a function call, we're calling the template function in passing in data. And then what that gives us is some HTML code,
- and the last step is to use jQuery again to copy back the HTML into the door of our content element.

So this is a sequence of operations we need to do to turn our template text and our data into actual HTML code. And I think now we can go through, we're ready to go through more complete examples of HTML code that does that. So, first off, I just want to show you what the code looks like. It's a very simple HTML page. It's nothing very different from what we've seen before. The only difference is what's going on behind the scenes. So, the thing I really want to show you is the code.

```
<head>

<!-- include our libraries and css files -->

    <script src="js/jquery-2.1.4.min.js"></script>
    <script src="js/handlebars-v3.0.3.js"></script>

    <script src="js/bootstrap.min.js"></script>

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

</head>
```

The first thing I want to show you is in the top in the header, we're including a bunch of script. So we're including jQuery as we've done before. But, we also have to include the handlebar script. So handlebars is a bunch of JavaScript. We have to bring that in. And also with bringing in Bootstrap which is JavaScript in CSS. So everything in Bootstrap we've seen apart from Handlebars which is implementing in templates.

```
<!-- the content of the web page starts off empty
      because we will fill it later from the template -->
<div id="content">
</div>
```

Down here we've got this empty div. So we've just got a div called content which isn't, hasn't got anything in it. Normally, that would look really strange in a webpage, having a div with no content, because it would do nothing. But, that's here because later on, we're gonna fill it with stuff, in that content. We need something there with nothing in, so that when we're creating stuff with our templates, we have somewhere to put it.

```

<!-- this is our template
it displays an image with a title and author headings
the bits in curly brackets {{}} are template expressions -->
<script id="image-template" type="text/x-handlebars-template">
  <div class="title">
    <h1>{{title}}</h1>
    <h3 class="author">
      {{author}}
    </h3>
    
  </div>
</script>

```

Here's our template itself. This is exactly the code I've shown you on the slides.

- The important things to notice is in the script. It's got type handlebars, template, and we've got the template expressions `{{author}}` which are where we're filling in the data.

And then down at the bottom, we've got another script. This is just standard, normal JavaScript, and this is where we're actually putting the templates in the dense data together and it's going through all the steps I've shown you.

```

<!-- javascript code to fill the template -->
<script type="text/javascript">

  // grab our template code from the DOM
  var source = $("#image-template").html();

  // compile the template so we can use it
  var template = Handlebars.compile(source);

  // create some data
  var data = {
    src:
"https://upload.wikimedia.org/wikipedia/commons/thumb/9/97/The\_Earth\_seen\_from\_Apollo\_17.jpg/600px-The\_Earth\_seen\_from\_Apollo\_17.jpg ",
    title:"The Earth seen from Apollo 17",
    author:"Ed g2s"
  };

  // generate HTML from the data
  var html = template(data);

  // add the HTML to the content div
  $('#content').html(html);
</script>

```

We're grabbing the source out of our script tag. We're compiling it to create a template. Here I'm creating the actual data itself which is an object, got source title, author, you'll notice source is actually grabbing stuff from the URL's, grabbing stuff from Wikipedia, which is all creative comment, so we can use it. But it could also be like local file name and then here we're actually putting that data into the template, generating an HTML which we're putting in this variable HTML. This is just a text variable, HTML is just text so it's not a problem to have HTML in the variable. And then finally we're using the jQuery HTML command to put the HTML into this div content. Which as we see at the top here is a empty div that we created at the beginning to put stuff into so all of this together will allow us to sort of pull the data in, put it through the template, and then add it to our web page.