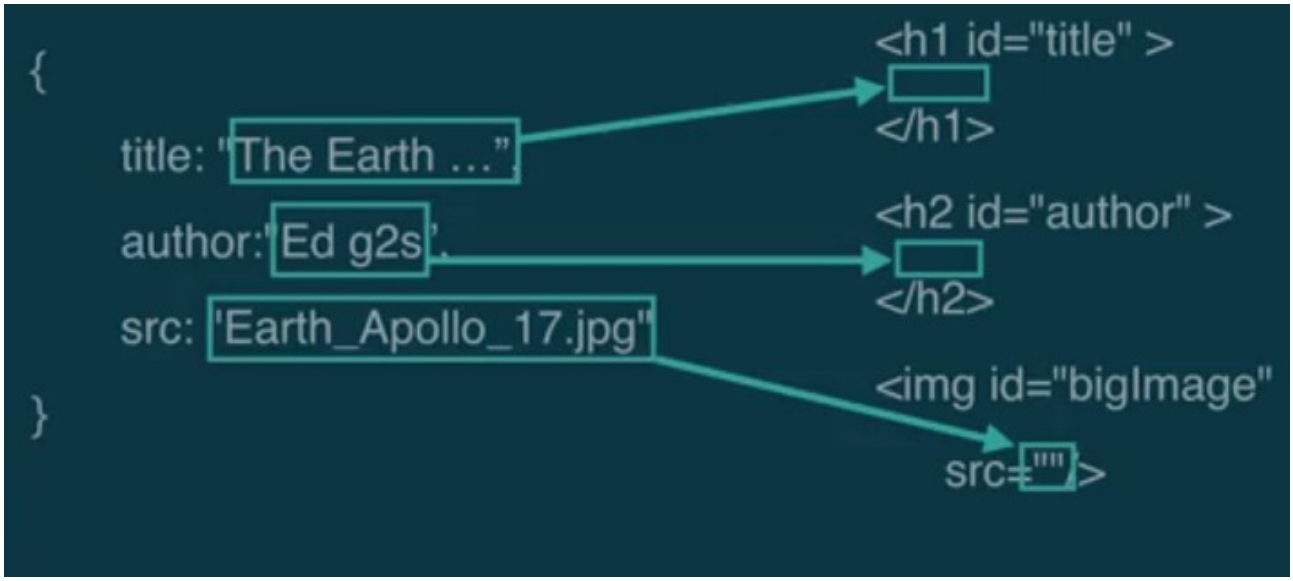
2.3.1.3:JavaScript templates with Handlebars

JavaScript reference site: https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference
Handlebars templates: http://handlebarsjs.com

So, you've learned one of the most important concepts in JavaScript and web programming, which is an object. And we've seen that it's a little bit different from what you might know in other programming languages. But it's a really powerful tool for creating structured content, and that's really what we need if we want to do this separation between the structure of the HTML and the content.
So, we've got the content of your webpage as an object with a bunch of variable for title, author, image file name. And then, we've got this HTML file and we need to put the content that comes from the object into that HTML, and we're gonna start looking at how to do that.



So, let's just remind ourselves. We've got this object which has got a title field that needs to go into an h1, or whatever, tag, an author field which needs to go into another text tag, and a source field which needs to go into an image tag. So how do we go about doing that? Well, let's look at that, we've looked in terms of the actual HTML. So, on the right here, we've got some HTML tags. The title needs to go inside this h1 tag, the author inside this h2 tag. And the source is not going inside the tag, it's actually going into the source attribute of the image tag. And what we've got here, I've kind of recreated these holes inside these HTMLs. So, we've got kind of this empty space in the HTML inside the h1 tag that needs to be filled with the content from our objects.

- And we need a way of creating those holes, and saying put some content in here. And the way we're gonna do that is use something called a template, HTML templates which, essentially, we can think of as a bunch of HTML with a bunch of holes in that can be filled in later using content from a JavaScript object.

- Now, JavaScript on its own doesn't support templates. They're not a built in part of JavaScript. So we need to use an external library to create these templates, and I'm gonna use one called Handlebars.

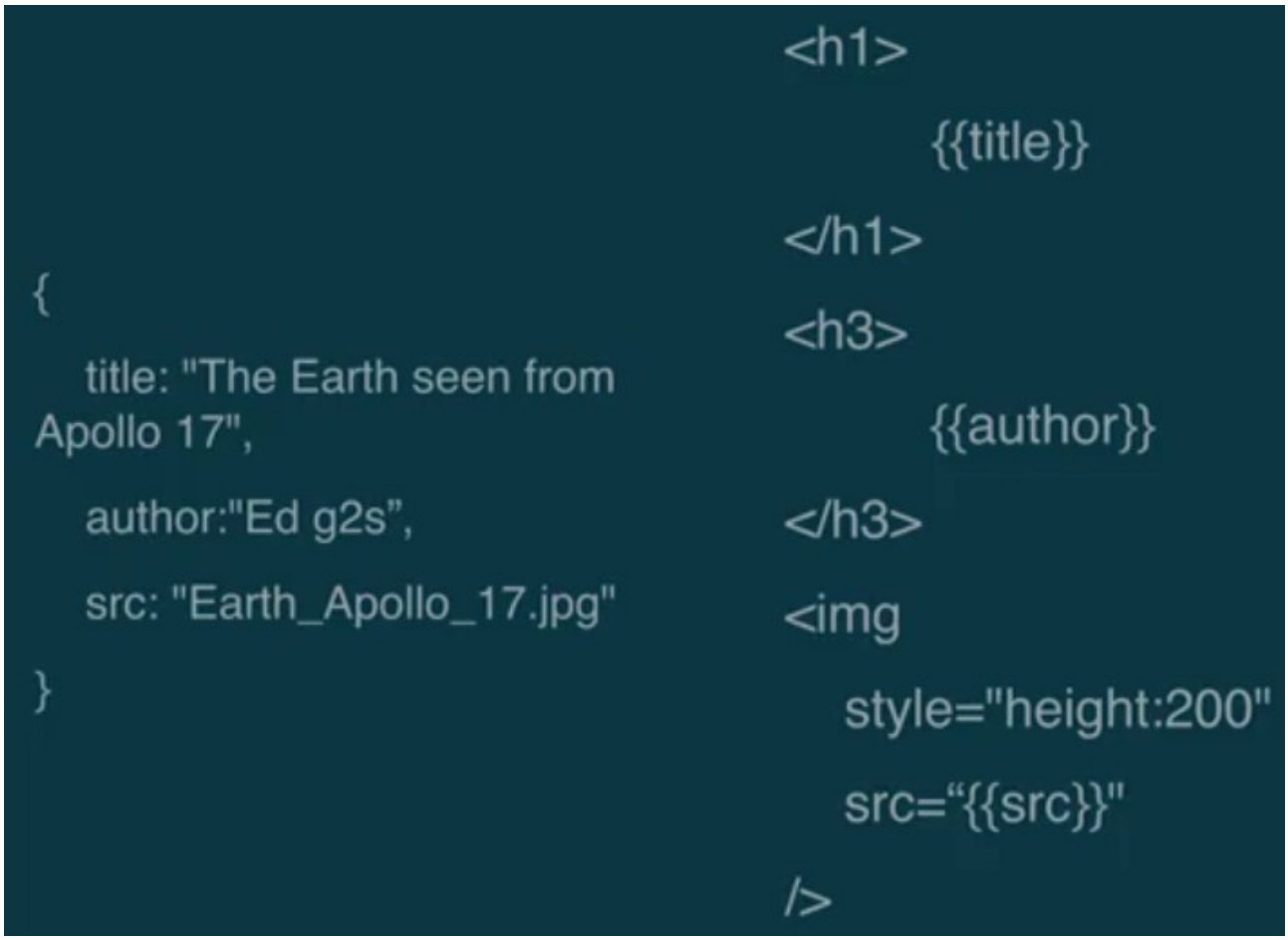
You can download it from Handlebars website, and there's lots of documentation on there to help you get started with your templates and using the Handlebars library. And, more importantly, it will use a very similar system. It's the basis of the Meteor System that we'll be talking about in the next MOOC of this specialization, when we move on to server-side programming.
So, it's gonna guide everything together. But we're not quite using Meteor yet. But we can use Handlebars, which is a building block of it and how they do templates. So, lets look at an example of a template using Handlebars. So, this is a template. And it looks a lot like HTML, but let me point out a few things.

- First off, it exists inside a script tag, just like our JavaScript. But, it is not a text slash JavaScript. We looked at the type of a script tag. This is text/x-handlebars-template. This is saying this is a template, it's not more JavaScript, you need to do something different with it.

- Then, these are the really important things of the template, they're called template expressions. And these are the holes I was talking about. These are the things that, the placeholders in the template code where you're gonna put stuff. So, you need to know where stuff is gonna go. There's a hole.

There's nothing there yet, but once we start filling in from the data from in our object, we know where to put it. And as you can see, there is one for the title, one for the author, and one for the source. And each of these will be filled separately from different member variables of our objects. Each one of these starts with two curly brackets, and ends with two curly brackets, just to show this is the template expression, this is where we are. If you're wondering why the library is called Handlebars, it's based on another library called Moustache. Handlebar, moustache. And somebody, somewhere along the lines thought that these curly brackets looked a lot like moustaches, so, called the library that. You may or may not think that. Okay.

So, let's look at how this actually works in practice. So, on the left is our object, on the right is a template. And, basically, each of these templates lines up with one of the member variables of their objects. You can see they've got the same names. That's the simple way the Handlebars know which lines are with which. So, the title template expression has to be filled with a member variable of our object called title. The author template expression is looking for one called author. The source template expressions look them up from source. So, the template simply tells you the names of the member variables of the object, pulls those values out, and puts them into the HTML.



That's all it does, it's very simple. But, as we can see, it's a very powerful piece of technology for separating content and structure. Let's look in a bit more dana on how to use a template, and how data from an object can go inside a template.

- So, on the left here we have an object with a bunch of member variables, title, author, source. And on the right, we've got a template with a bunch of template expressions, title, author, source. They've got the same names.

And what simply happens is that the template Handlebars takes the value of the member variable called title and copies it into the template expression called title. So, it replaces that template expression in the HTML with whatever is inside that variable. And it does the same thing for author, and the same thing for source. And the way it knows how to copy what, is because the expression the same name as the variable. So, it's actually a very simple thing it's doing. It's taking, it's looking for name of an object variable, taking that and copying the value into the HTML and creating a new HTML web page based on that. So, very simple and really powerful technique that allows us to separate the structure in the HTML from the content in the object