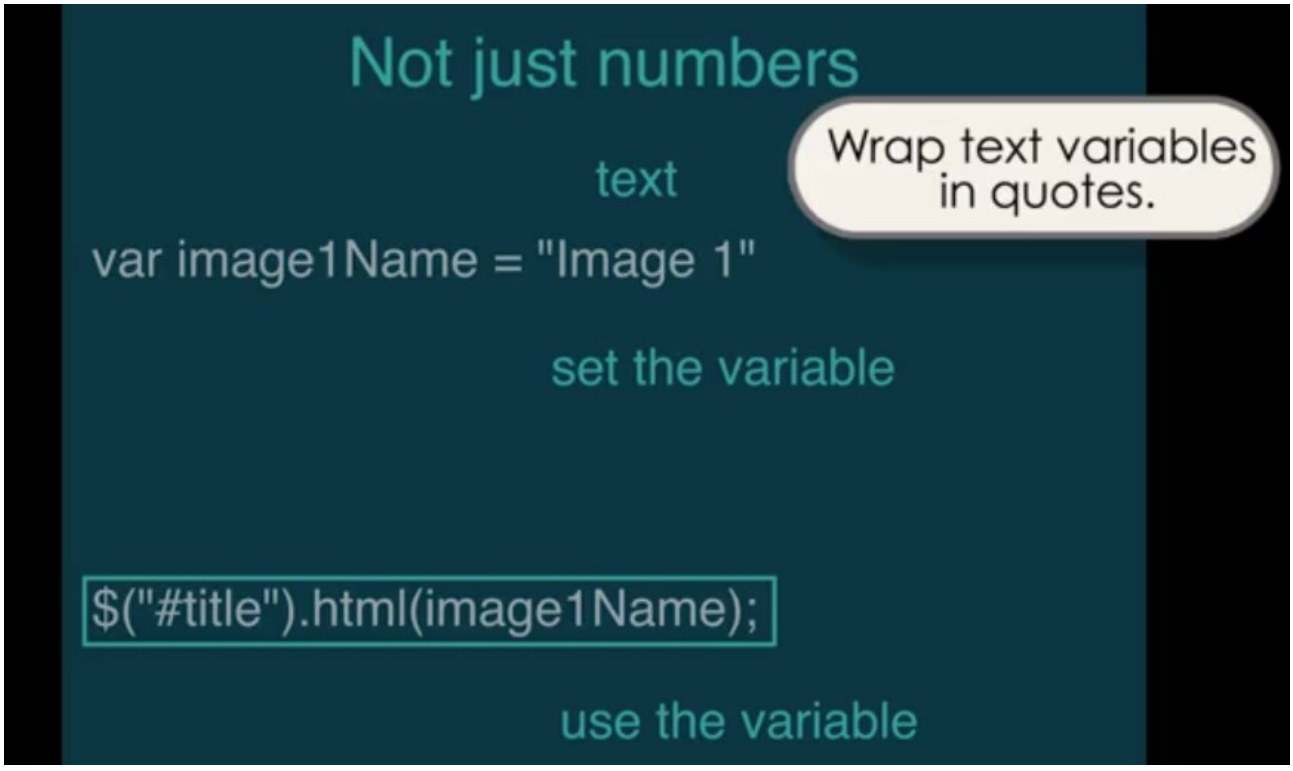
## 2.3.1.2: JavaScript objects

So we want to separate out the structural webpage from it's content. And we've seen something like this before. We've separated out the style from the structure when we separated out CSS and HTML. Now we've got the structure. That's HTML. We know what that is. But what's the content? How do we represent that?

Well, actually, you've kind of already seen it before. In the first moot, we talked about a variable, and just to remind you, a variable is a box in the computer's memory that you can put stuff into. It has a name, and you can also change the values in it. And in the first course of the specialization, we must use variable store numbers. But there's no reason why it has to just be numbers.

So for example, here is a variable that's holding some text. The text image1 and you notice that there's some quotation marks around the text. The reason for those quotation marks is to tell JavaScript that we just want this to be a bunch of text. It's not the name of a variable or a function or anything. It's just the actual text, image one, that we might want to display on a webpage. And, just as we did, in the first we can use that variable, to, put it into the, webpage and here we're using to set the HTML of the webpage to the content of that variable. So we've got some text stored in a variable and we can put it into the HTML structure using jQuery.




So we've already made the separation we want to do. We've got the content which is the variable and we've got the structure which is the HTML and we're putting the content into the structure but we're not quite there, yet, because all we've got is one bit of text and, actually, what we want is something a little bit more complex than that. So, we don't just want one title, we've got an image and that's got a bunch of different bits of data.

- It's got a title, it's got an author, it's got an image file, It's likely you have an allocation, the time it was taken. There's gonna be lots of dana that we might want to display. So we want something more complex than just a single variable.

So for example, as we've seen before, for this example, we've got the types of author, image, how we're gonna represent that all in one variable.

- Well, we're gonna use something slightly more complex called a JavaScript object.

And this is what it looks like.

```
var image1Data = {
    title: "Image 1",
    author : "Tuxyso"
};
```

It's as we've seen before, it starts and ends with curly brackets just as the functions do and if statements do. And we're assigning it to the variable Image 1 data, so this is just a variable. And inside the body, what we've got is a bunch of these pairs of a name and a value.
And each of these variables in themselves, they're kind of mini variables within the big variable. And I wanna make a note here of some of the punctuation, the syntax. That these are all the name and the values separated by a caylon and sometimes, it comes more naturally, you put an equals sign cuz that's what you do when you create a variable. In this context, you need to use a caylon, you just need to remember that punctuation and they're all separated by commas, not semi-colons.

- This is how you might use an object. So you've got the name of the object, if you want to get hold of the title, you just use the name of that individual sub-variable, the member variable and you use a dot to get a hold of it: **image1Data.title**

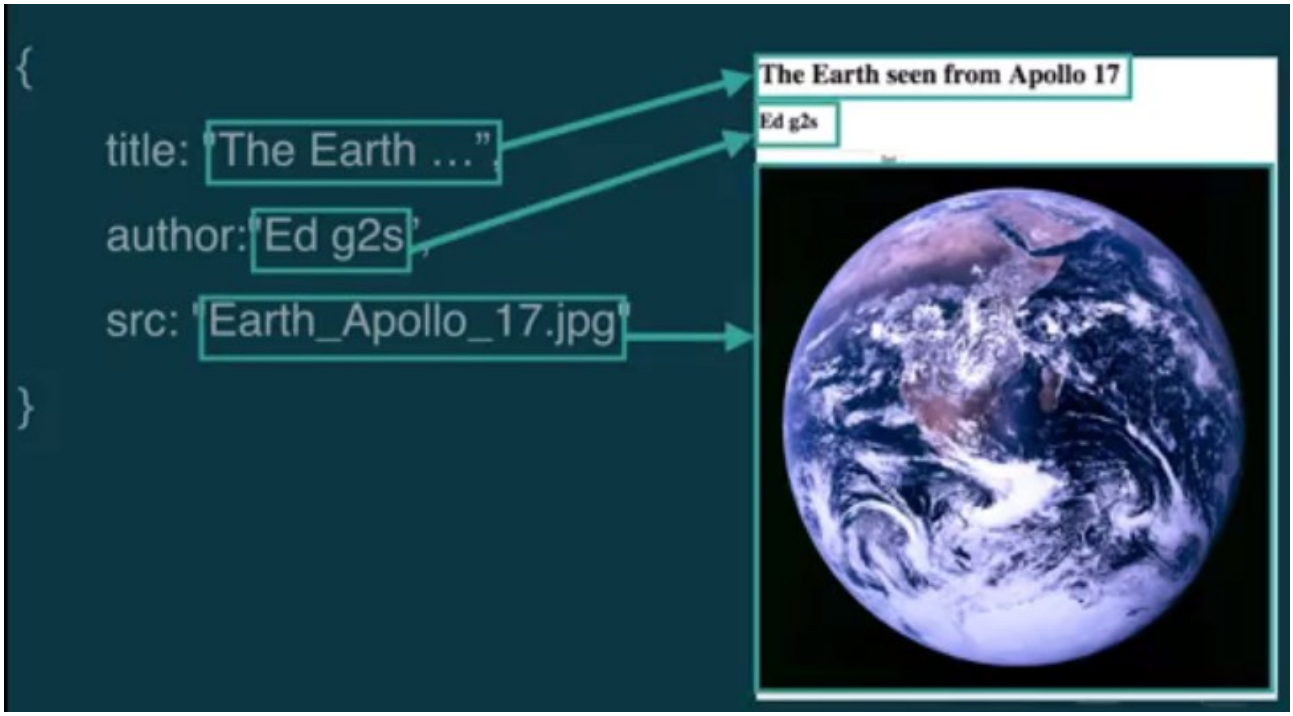
So just to recap what this is. If we think of one variable as a box, we can think of an object as a collection of boxes, a little stack of boxes. One for the title, one for the author, and one for the source of that object.

Now I should say at this point that if you've done some programming before in a language like Java you might be feeling a little bit confused, cause this sounds quite a lot different from what you would do in Java or C++ or even Python. I'm talking about objects, but not once have I mentioned a class, which are fundamental to programming in most languages.

- JavaScript does not have classes It just has objects. So, you don't need to create a class in order to create an object. All you have to do is do exactly what I've shown in the last slide, here, is just type out this curly brackets and list.

The men variables of the object and list their values and you've created an object. No need to find a class before hand. You can just create an object just like this and each object in that way is unique. And it might take a bit getting used to if you're used to object going in another language. But if you're starting off actually, it's really quite a lot easier and once you get used to it, it's quite a nice way of doing things.

- So, just to recap, slightly, about what an object is, it's a complex type of variable which can contain sub-variables. And that allows us to do exactly what we want to do. It allows us to express the content of a complex webpage like this, because it has multiple elements.




It's got a title which can go into the title HTML. And an author field, that goes into the author HTML. And a source which supplies the image for. The fill for the image tag. So, now we've got everything we need to represent the contents of our webpage and the next step we need to do is take that content and put it into the structure of the HTML which is defined by the HTML we've already looked at.