**DwBP - 7 - Displaying Video Information From the mediainfo Property**

In the last video, we retrieved the video object from the catalog and used it to replace the video that was in the player with the new one that we retrieved. In this video, what we're going to do is retrieve metadata about the video that is currently in the player and then use that somewhere in the HTML page. In this case, we're going to dynamically build some HTML and injected into and HTML div element.

You can use this data in other places though. You can use it, for instance, in a custom in screen. You can use it in an overlay. You can use it in a call to action. And so although we're just going to be displaying it on the page, the idea of retrieving the meta data and using it in your current page, you can do lots of things with that data.

In the first couple of coded exercises, where we've been taking care of use cases we've developed, I've gone through and actually entered all the code. You've watched me do that. I think now we have a good enough foundation, where rather than me typing all the code, we're just going to go through and look at the code, and we can get the concepts without you actually watching me type them. And going to this time is task three solution. Remember, we have a starting file, the steps, and a solution for each one of our tasks. In this case, this is the solution.

So let's look what's here. This we've done before. That's a style that just re-sizes the video. Here's a video tag. This is a BR, just to give you some space between the actual player and where the target is, where we're going to inject some HTML. Of course, here's the script that is associated with our player, and here's the information we really want to look at. And this is our JavaScript.

The first thing to notice that's a little different is we have it uses code before-- we've talked about it-- where we're using the ready method. And basically what we're doing is saying, take the video tag that we've identified with the ID My Player ID, convert it to a video JS, a Brightcove player. And then, when it's ready, use this anonymous function.

The next couple of lines we create some variables. And here, something once again we're familiar with, we are using an event handler. Here using the on method to create an event handler for the loadstart event. And what this means is this is when the video begins to load into the player. And we also know that when that happens, the media info property is populated, and we can then get to the data. Again, we couldn't do this just on ready. We have to wait for the loadstart event to be dispatched.

Here we log to the console what this media info object is. Let's go look at that. So we have an idea of the kind of information we can use. I'm going to go out to the s Notice, I have the developer tools turned on, and I'm going to collect task three solution. And it starts to play, and some things happen which aren't really of our concern just yet.

What's interesting to us-- here's the media info, and here's the object. I'm going to click here so we can see all the pieces of the object. And, in fact, here's all the information you have access to in the media info property. Some of them aren't a surprise at all, like the description, the long description, the name, the poster that's used before the video starts to play. We have the duration. All that information is stored here.

Notice, we also have, if you have any tags, any meta data there associated with it. We also have the sources, in this case. When I uploaded the video, I was using her rendition profile that created 14 different renditions. Here you see one of them. So this is all of the data we can access when we use the media info property.

Let's go back to our code and look what we did in this case. And you have a hint of what you see right here under the video. What we have is the dynamic HTML variable, and we're going to populate that with a combination of HTML and also some variables from the media info property. So here you see we have a video title, and then we're going to make it strong, and I use myplayer.mediinfo.name, which we saw was out there. And then we add to that using this plus equals. We add to that the description myplayer.mediainfo.description.

We then want to get a reference to this div. And what the div here is we have an ID. It just gives us a location in which to put the dynamically built HTML. So we get it by ID. The inner HTML us what's inside of that div. We assign the dynamically created HTML, and then we play the video. And we can see that's exactly what it's doing. I'll reload the page. The video's playing. There's the video title. And there's the actual title. And here's a description, and there's the actual description.

By the way, also associated with that is on-demand training course, I've built codePens where you can go in and look at the different exercises. With a codePen here's the HTML which we're familiar with. Here's a simple CS with which we're familiar. And then here is the JavaScript.

What's nice about this is you can go and play with the code here and experiment a little bit. So let's say you weren't quite sure what this plus equal did. So I'm going to take that off. And when you wait just a moment, the page reloads, and notice you have just the description. The title is no longer there. Also notice that I commented out the place. I'm going to un-comment that. It will reload again. And now it's playing. I'm going to put the plus equals back in. It'll reload again, and there's both the title and the description.

So you have access to these codePens You can go in, click on it. It will open your browser, and you can modify. Of course, we don't let you modify in the original. It'll automatically prevent you from doing that. And so this is another way you can use the code.

Let's do a quick review now of the coding concepts we just used. Here we have the media info property. And as it says, it contains information on the media currently in the player. The property is created and populate after loadstart is dispatched. So you can't do it right on ready. You have to wait for the loadstart event. And you can see the last bullet, it's a convenient place to go in and use the data.

Here's a screenshot of the data that comes back with the media info. We actually looked at it in real time with our console. And here's just highlighted in red where we use the information from the media info and then the dynamic HTML, the HTML that we use in blue there. And you can see how we can mix the static HTML with the variables from the media info property to build some dynamic HTML, which we then inject into the page.

That completes task three, where we use the media info property to inject information onto our HTML page. One thing I wanted to point out-- it uses the ready method. We're going to be using that from now on. Also, if you want to experiment, there's the URL to the CodePen.

In the next video, we're actually going to use a different implementation of the player. So far what we've done is we've copied the actual video tag the script tag and put it onto the page. Now, in the next video, we're going to use the I-frame implementation, and we'll see both good things and bad things about using that. Hope to see you there. Thank you.