00:01 Okay, for this ponder, we're gonna learn about motels. And we're gonna have this gallery of pictures. It is gonna follow a mobile first with one column on a- or mobile devices, growing to two, and then three.

00:13 But if we take a look at what a modal is, let me go ahead and just demonstrate here. If you- a user clicks on an image, it will bring up this modal with a higher definition image, then our thumbnail that they clicked on.

00:25 And then they can- exit out of that modal to continue. Okay, so that's what we're going to be creating with this ponder.

00:33 under. And we're going to go ahead and download those images in the, in downloads. And I can find that zipped file, open it up, and you should see all of these images.

00:49 Notice there's two- two of the same image, but one says dash small, or SM, and one says dash full, and you'll see one is much bigger than the other.

01:00 So the one that's going to show up in the modal itself is that more high definition bigger image with less pixelation, and then the smaller kind of thumbnail pictures that they click on to begin with in the gallery will be the small versions.

01:14 So I'm going to go ahead and let's go ahead and come over here in my images right now. I just have that one from a past ponder.

01:22 So I'm going to go ahead and throw the rest of these into that images folder. So we have a whole bunch more.

01:31 Okay, now I'm going to go ahead and make another file in my ponder that's going to contain this gallery. And I'm just going to go ahead and call it modal.html, even though it's got a lot of other good responsive stuff.

01:44 We'll just call it modal. So we're going to go ahead and do the shortcut exclamation tab for that and change the title to modal.

01:54 We'll put modals. go ahead and paste this HTML. Inside of the body. And that will come in. Tab it in.

02:07 Okay, so we have a title called gallery. And then this whole section is our actual gallery of images. Notice it does have the small images.

02:15 Just to begin with, kind of like our thumbnails. Umm, figure it might be a new element that you might not have used.

02:21 This is a really good element to nest your images inside of, especially for something like a gallery. It's very, um, semantic and accessible.

02:30 And it's- It does help with ease and styling when you get to CSS. But sometimes people use it because it can give your image a nice caption or a little bit of text underneath each image with a big caption tag that goes along with the figure tag.

02:47 So lots of good reasons why we might want to use a figure tag. But just notice that our images are nested inside of a figure tag for each one of those.

02:55 Okay, we also have this new HTML called dialogue, dialogue, a dialogue element is really helpful for when you want to have something kind of pop up on the screen.

03:09 And so in this case, we're going to use that dialogue, which is already very semantic and accessible for what we're trying trying to do.

03:19 And then we'll make it even more like a modal with some of our CSS and JavaScript. Notice it is going to have an image in it.

03:26 Right now the image has no source or alt text attributes, but it will through JavaScript, depending on what they click.

03:35 And the button simply has an X in it with the class of closed viewer. Umm, so we can use that.

03:44 By default, dialogue elements are hidden. So it won't show. In fact, if we take a look at it. Right now there's no CSS.

03:54 No JavaScript. But if we look at the bottom where we expect to see that dialogue and the X, there's nothing there.

04:00 So dialogues by default. Do not have. Do not show. We have to actually have them show up, um, different ways.

04:09 Alright, so let's go ahead and copy over our JavaScript. So let's make a JavaScript file. Make sure I'm still in ponder and I'm gonna call it the same theme with .js.

04:19 And I will grab all the JavaScript here. Now it's not all included. We are going to be adding the code for this open modal function in just a bit.

04:32 The rest we will look at here. Let's see, we're going to be targeting a bunch of our elements so that we can be looking at the gallery itself.

04:45 We're going to be looking at the dialogue element, the image element, and that little x that, dot close viewer that we just saw in the HTML.

04:55 Okay, we're going to be listening for them to click inside of the gallery. Okay, and that's when OpenModel will get, uh, OpenModel will get called.

05:08 These other two I've given you, um, this one is listening for that X, that little close button, that we have here, that dot close viewer class that we had, and it's going to go ahead and close our modal if they click the X.

05:22 It will also close the modal if they click in the, in the dialogue element itself, and not the image necessarily, but anywhere outside of the image inside that modal, we'll go ahead and click it as well, or it will go ahead and close it as well.

05:40 Okay, but to actually get the modal to show up, that's what we're going to be coding together. we'll get back to that.

05:46 Let's go ahead and get our CSS done. And copy that. And paste that in. And before I forget, I'm going to go ahead and link those two up, so we can have the CSS.

06:15 And the JavaScript. And we do need to defer here, because we're in the head section. Okay. And let's go look at the CSS really fast, so there might be a few things that might look a little more, uhm, a little unfamiliar to you.

06:35 Okay, so here we're taking maybe some images that aren't a true square, and if you looked at, how they looked here, they look more like a true square.

06:45 And maybe they were more rectangle. If I go ahead and just give them a width and hype that are the same to make them a square, it might mess up the aspect ratio and squish, you know, their faces and make them look a little odd.

06:58 So, So we use this object fit cover, and then it's going to be able to leave their aspect ratios alone so it doesn't look squished.

07:07 It's going to fill up the given dimensions of this 250 by 250 to make a square, but it's just going to clip the images.

07:14 It's going to kind of crop the images at the edges in order to do that. So you might miss a little bit of the picture, but it won't mess up that aspect ratio.

07:22 So this is kind of nice. Um, so that might be a little new to you. The rest, you should kind of, again, we're using the VA.

07:30 Each unit of measurement, which means the viewport height. So when we're looking at maybe the dialogue image, I wanted to fit so much of the viewport height.

07:40 So that might be a little new to you. This position Absolute. It might be a little bit different. It's going to say, I don't want this to be in the natural flow of the HTML.

07:54 So we're actually looking at that X in this case. Okay, so it's saying that I don't want this X to be in the flow of HTML.

08:03 I don't want it to come down below my image. I want it to be outside of the flow. And so I'm going to just say position absolute, which means whatever its parent position, if it's got a position parent, I guess, it would be adjusted to that parent.

08:20 But if it doesn't have a positioned, um, ancestor or parent, it's going to use the, body. So with position absolute, because it's not really in the flow of HTML, you need to tell it where it's going to start, where it's going to be placed.

08:33 And so you have to say, I want it to be ten, um, viewport height from the top, and two viewport height from the right.

08:41 Okay, so that it has that position where you want it to be. We'll see that as we run it. Okay, and then again, we do have that responsiveness that we saw with our scripture activity, where we're going to go from one to two, when the screen gets a little bigger, and then to three columns when it gets

08:56 even Thank much. Um, and then we've just adjusted kind of how the dialogue or the modal is going to look at each of those views here.

09:07 Alright, so let me save that, and we'll go ahead and take a look. And you'll see all that CSS going on with the one to two to three columns, so that should all be working.

09:19 And right now when we click on the image, nothing's happening because we haven't, created or finished that function yet. Alright, so a few different things here.

09:29 When they click an image, how do we know which image they clicked? Because we really do want the model to, show the image that they clicked.

09:39 We're going to be using this e, or event, parameter. And you notice, when we called this function open modal, it didn't really send in a parameter at all.

09:51 So, what's going on here is, This is called the event, and it goes along with the object. In fact, if you look, it explains it really well in your introduction.

10:01 It says, um, it always is going every time you have an ad event listener, every time you trigger an event, and you have this special parameter that you can use if you want to.

10:16 A lot of the time. Sometimes coders will call it event, or e, for short, and it's going to have this object, this event object, with all this different properties and information and methods that relate to that event.

10:30 Where was it clicked? You know, where were- they add on the screen and all this kind of things that it will keep track of.

10:37 So it's going to give you some control over how that event behaves. So, if we want to kind of see what that is, let's go ahead and console log that e.

10:49 And it's only going to do, you're only going to be able to see this console log. We'll only click on one of our pictures.

10:55 So let's go ahead and open our console. And if I click there, notice we get that event to come up.

11:05 It's going to know all this information about it, where the click was exactly x and y on the page, all this stuff.

11:13 The part we're going to use is this target. It knows that I clicked on an image, and it knows exactly which image that was, which is really cool.

11:25 So, if I use that property that I just saw the target, and do it again, it actually grabs person sitting on a cap.

11:34 If I click on this one, it'll grab a work desk if I click on this one. It gets the exact image that was clicked on.

11:41 Which is pretty cool because we're kind of just listening for the entire gallery to get clicked on. So this is going to tell us which picture in that gallery the user clicked on.

11:52 Which is really nice. We- haven't gotten to looping through an array with for each. That will be another way that we'll be able to do that.

12:01 We'll be able to listen for all the images to be clicked. But since we haven't learned that yet, we're going to go ahead and use this event.target property to- do that.

12:11 Alright, so, let's go ahead and do this, this function. Okay, first of all, we're going to store that target in a- a ver- or a constant called IMG.

12:32 So we know which image got clicked, and it's going to be stored in IMG. Then we're going to go ahead, and because we know which one got clicked, we can store- the different parts of that.

12:43 We can store the SRC attribute, just this- this string right here. We can also store the alt text, which is the blue part here.

12:53 So that- when we bring up the bigger picture, we have that same information that we need. So we're going to go ahead and say, hey, I want to assign the constant SRC, the actual- So we'll add.

13:06 Attribute SRC. So we'll say, take that image that got clicked on, we just set up in it as a constant, and use the get attribute to grab the SRC.

13:19 So what's going to get assigned to this SRC call? A constant is what's inside of these quotes right here, images slash catch dot dash SM dot JPEG.

13:28 peck. Okay, we can do the same thing for the alternative text. We'll call that ALT. And again, that same image, get attribute.

13:39 This time we want the ALT attribute. So what's gonna get stored in this ALT? Is everything in the quotes here, person sitting on a couch with a phone.

13:51 Alright. If that's what they could. So that's it. And then last one, we're going to go ahead and change the small picture to the big picture.

14:03 And we're going to store that in full. And we're going to do that by giving a string. So we're going to say, hey, take that SRC that we just got and replace anywhere it says SM with full.

14:20 And why are we doing that? Because, remember, we have the exact same named image, except it's going to be dash.

14:26 Full, instead of dash SM. And we want the higher definition one to go into the modal. So we're getting ready to do that.

14:34 We're going to be having full be the SRC for our new image that we're going to be putting on here in just a second.

14:41 And remember. It does explain what replaces here. It's going to be searching for a string value. Thank you. And it's going to replace something that it binds with something else.

14:53 So we're replacing SM with FUL. So it'll have the same full, you know, book dash dot jpeg, but instead of SM, it's going to have FUL.

15:04 Alright, now we can go ahead. Head and work with that modal. can target this modal image that we had up here.

15:13 And we're going to go ahead and say modal. Image. And assign the SRC to be whatever was in that fold.

15:26 So it's going to bring up the same picture, but with the high definition one. And then again, we want to target that.

15:33 Same modal image. And change its alt. And the reason we're doing all this, just to leave the same alt that the little one had.

15:41 We're going to have the same alt for the big one. And so we're just kind of transferring that one over.

15:45 We didn't have to replace anything with that one. Then, once the modal image, Has the proper, Remember, our modal image had nothing in it.

15:54 So now we just decided an SRC and an alt here. Now we need to go ahead and open that modal.

16:02 And that's with the show modal. Oops. Method. Okay. So now let's go ahead and take a look. And when we click it- we should get that modal to pop up.

16:16 There's that close button that we were talking about earlier. How we had it come down from the top and right with the absolute positioning.

16:24 And then whichever way. One we click on, it's going to bring open that high definition of the same image. edge.

16:31 Okay, so it brought in small, but if we look at our image in the modal right now, it's got the full.

16:42 Do you see that? Which is really cool. So, I what knows which image we clicked on with that event object.

16:49 We're using the dot target property to know which one they clicked on. We're taking the small thumbnails SRC and all the attribute.

16:56 Buryp, Placing dash or SM with FUL. And so when this gets clicked, it will go ahead and open up that more high-definition picture with you.

17:06 Which is pretty cool. The way we did the show modal here is we've got- it so that you really can interact with the rest of the page.

17:17 There is a dot show that would let you do that, but we wanted it to be a true modal, which you really don't- you know, mess with the rest of the page until you close out.

17:25 We can click on the x to close out. We can click outside of the image, but still in the dialogue to close out.

17:31 We can also, because this is by default hit the escape key on our keyboard, and it will also escape out, which is great for accessibility.

17:41 So, lots of different ways for the user. Even though they can't interact with the rest of the page once this is open, it's easy for them to close that modal out.

17:50 Alright, so I see- I think that is pretty much all the new information that you're going to need. Your homework will be a little bit similar to this.

17:59 You will have a responsive menu as well as a gallery. You will have- Have a good all the same image in your homework, but realize that it will work even if you had different pictures.

18:10 So feel free to put different pictures in there if you'd like. But this is kind of where the magic is happening.

18:16 Um, it's going to go ahead. You don't have to have the console here, but you- You will have- Yes. Something similar with the rest of this for your assignment.

18:23 Alright. That we have our responsive gallery with- motos. So,