

JavaScript Click Event

Daniel Quinn

OK, so, all that really is needed for this is a single click event. And we're going to be handling states based on the information that we get from the users' interaction with the page.

So, we need to set a couple of variables to start. The first thing is the state of the whole quiz. And we're going to say "true" in this case represents that it's available to be accessed and can be interacted with it. So, we're just letting the quiz state variable equal true. And then, we'll also want to save the selection of the visit of the user. So, this is an array we're going to create that might have multiple values. In our purposes, it's only one value, but I just want to show you how array would work. So, this is a variable that's equal to an array, and the array is empty right now.

Then we want to target our--whenever they click on any of the list items, we want to have an event happen. So, the list item itself is called quiz choices. So, that's the ID it has. So, on click of it, we're going to say function events. And then all the stuff inside of here is going to happen.

OK, so now that we have this click event ready to go, we want to make sure that the stuff in here doesn't run if the quiz is over. So, don't run if the quiz is over. So, what we're going to do is we're gonna check that quiz state. So, if it's equal to false, we're going to set it to false later. We're going to just return false on this function, and that way, nothing happens. Otherwise, we'll get started.

So, we're going to get the element they clicked on. So, get the selected element. So, we're gonna make a new variable that exists in the scope of this function, this item, and we're going to set it to the element we clicked on. Now, we want to figure out which one did they click on, and we're going to do that based on the text inside the element. So, we're going to add the elements texts that they selected to the array.

So, if they clicked Han Solo, what we'll say is the selection variable up here that we set outside of the function selection, we're going to push into the array, the text of that--the thing they picked. So, what this means is, you know, this item is equal to the thing they clicked, and we're going to say, "Hey, reach inside of this item and get its text content," which is the text "Han Solo," and put it into that selection array.

So, now that we've added the selection Han Solo potentially to the array, we need to check if that selection is a Jedi, or if it's not, then we can end the quiz. So, we'll say, check if the selection includes a Jedi, in which case they're going to fail the test. So, we're gonna say if. And then, selection includes Darth Vader, for example, or the selection includes Princess Leia or includes Luke Skywalker. These are all conditions that will fail the quiz. Skywalker, there we go.

If that's the case, then what we want to do is visually indicate that to the user and then end the quiz. So, if it does fail, if it contains those things, then we're going to make sure that the list item they picked--we're going to look through its class list, and we're going to add the error class to it. That will highlight Han Solo red, or sorry, the Jedi that we picked red. And then, we want to display the results of the quiz itself. So, class lists add is active, which will make the result markup visible. So, insert.

The last thing we want to do is include the text content or the name of the things they chose that's wrong to show them which one was wrong. So, quiz choice, we're going to target the--that particular results markup, which is the span that contains this place where we're going to put the names. That's this. And we're going to insert the name that we've captured from the selection.

So, the inner HTML, we're going to set that to its text content. So basically, what that means is set the span's contents in between here to the text Han Solo from the item they picked. So, at this point, we're going to see how far along it sets us. We also want to make sure that the quiz is disabled when this whole thing is executed. So, that's where we set that quiz state we set up in the beginning to false. So, let's take a look. We should at least have the erroring part. So, if I pick Princess Leia, I should get a failure, right? And then I can't continue because the quiz state has been false.

So, at this point, you know, all we've done is don't run it if it's an error or if it's in ended state already. Get the element, put its text into an array, check if it contains any of the invalid choices, and then update the result markup so it becomes red like this, as well as the button itself, and then end the quiz.

So, now we need to handle the state as if they chose the correct answer, which is Han Solo. So, we're gonna say, "else." So, we're just--all you have to do is basically light it up as green. So, if it does not contain a wrong answer, we'll visually indicate that. This item class list--so this is the one they selected, just something to do with the error, is success. Success class in here. And then we're going to show the success message.

In this case, we don't need to update any markup inside of there with text because we're just saying, "Hey, you got it right." So, if we save this, we should see the success message when we pick Solo, but we didn't, so let's see what happened here.

So, we've got this item. So, I made a typo here. It should be class list. There we go. Let's take a look. We got the success. And then, of course, the quiz ends. Because once it leaves that whole if condition, in either case, we're going to turn the quiz off. So, they have to refresh the page to continue again.

So, all we've really done in this case is made some simple markup that we're manipulating. We're manipulating it on the basis of its IDs rather than classes, just to separate presentation from functionality. We've made some markup for the results, which is the error message and the success message. And then, with a simple click event based on when they're clicking on these list items, we're getting the thing they clicked on, taking its text out, checking if it meets one of these conditions, and then updating those results, true or false. And we're doing this all in pure JavaScript.

Nowadays, in modern JavaScript, there are many ways to skin a cat. You know, React is a really popular framework to work through nowadays. In the past, we used jQuery as a library to make JavaScript easier to write. So, what you want to do when you approach JavaScript is learn basic JavaScript first, and then you can move on into more advanced techniques in modern JavaScript.