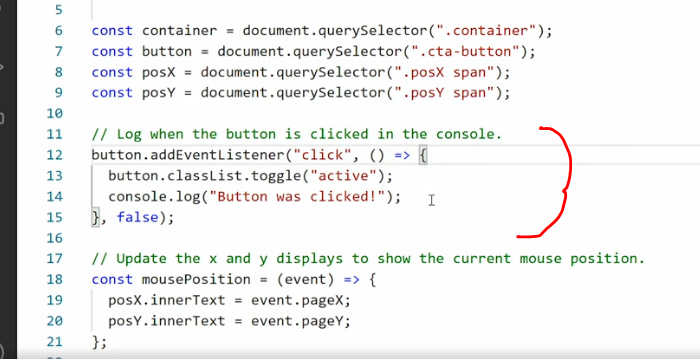
* - [Narrator] **Everything that happens in the browser is an event**.
* *Opening the browser is an event, visiting a URL is an event, moving your mouse or clicking it or touching your screen is an event, scrolling is an event, clicking the back and forth buttons in the browser or reloading the document or hitting a key on your keyboard, anything and everything that happens in the browser is an event.*
* And with JavaScript, we can listen for those events then capture them when they happen and do stuff when they are triggered.
* You can get a full breakdown of all available events in the MDN Event reference.

<https://developer.mozilla.org/en-US/docs/Web/Events>

* Scrolling down this page you'll realize there are a lot of events you can monitor and any of these events can be monitored and when they are triggered you can do whatever you want as a result.
* Let me give you a basic example of what I'm talking about.
* In exercise files for this movie I've set up this very well designed web page that has very functional functions like when you move your mouse around you can see the X and Y coordinates of the mouse in relation to the window, when you harbor your mouse inside the rounded corner box here that changes color and if you go to the button and click on it it also changes color.
* And if you go and look at the console, you can also see when you click the button it tells you how many times the button has been clicked.
* Very cool.
* Now, **all of these is powered by event listeners.**
* In the code for this example you'll see I'm using several different event listeners to listen for different events.
* And this here at the top is a good example of what an event listener looks like.
* First, a) we find an element that we want to listen to, an event for, in this case the button.
* And b) the button is the button you see in the middle, it's the elements that has the class CLA button.
* So, c) we grab the button element and then we say, add an event listener to this element d) then we have to specify what event we are listening for, in this case click and e) when that event occurs we can call a callback function.
* F) That function sits inside the event listener at this time as an arrow function.



* F1) And this arrow function does two things, first, it grabs the button element, the classList property and toggles the active class on or off F2) then it console logs out the text the button was clicked.
* That's why when I click this button here we are toggling a class on and off and that class changes the color of the button.
* **An event listener can be appended to any element within the window or the Document Object Model.**
* So, down here I have appended an event listener to the window object itself, that means everything that is happening inside the browser window and then I say, "For this entire window, add an event listener and trigger it anytime the mouse moves.

Graphical user interface, text, application

Description automatically generated

* " When the mouse moves, trigger the function mouse position, the function mouse position sits up here and it captures the entire event then it grabs just the event pageX property and pageY properties and put their values of those into the inner text of posX and posY.
* That's what you are seeing down here, that's why we can see this value update when I move my mouse around.

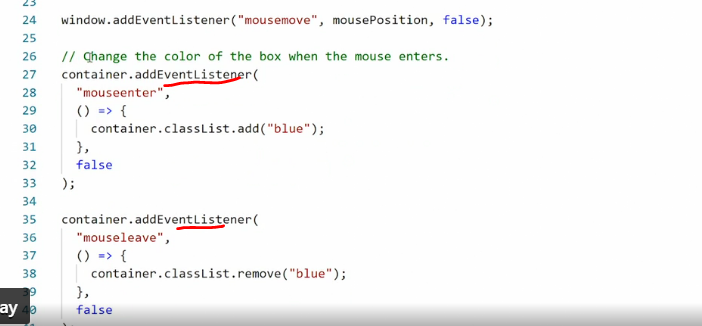
Graphical user interface, diagram, application

Description automatically generated

* Now, if I open my console again and we can go in and just dump out the entire events, so I'll just console logout event, this event argument is automatically passed into the callback function by an event listener.
* So, when I save this and output it you'll see when I move my mouse around, whoa, a lot of stuff happens here.

Graphical user interface, text, application

Description automatically generated

* So, let's open one of this event objects.
* So, these event object gives me all the information I need about the current event that was triggered.
* So, here you can see that we have, let's see, clientX and clientY and we also have layerX and layerY.
* So, there are many different things, pageX and pageY.
* I think I'm tracking PageX and PageY, yeah.
* That's because those are the coordinates for where the mouse pointer is inside the event window, but you can also see there is a tone of other information here we could pull out if we wanted to.
* So, this event object contains absolutely all available information from the browser about the event you are tracking and then you can take anything from that object and use it if you want to.
* Scrolling further down we have two more event listeners.
* 
* These are monitoring the container and the event lister is looking for the mouseenter and mouseleave events.
* This are pinned to the big blue container here.
* So, when I enter the container with my mouse the first event listener is triggered, the one that tracks mouseenter and when I leave the container, mouseleave is triggered.
* Now, if you look closely, you'll see this actually depends to the event itself.
* So, I have rounded corners here, I'm outside the rounded corner, only when I go inside this will get triggered.
* So, this is a detailed event listener that really does listen to whether I'm inside or outside the current event.
* What I want you to take away right now is the following:
* a) everything that happens in the browser is an event and
* b) you can monitor any of those events by adding an event listener and listening for them.
* c) Event listeners can be appended to any element inside the window and inside the DOM and
* d) you can trigger whatever function you want either using an anonymous function inside the event listener or by using a callback.
* E) And if you use a callback or an anonymous function, you can grab the event object and do something with that event object if that is meaningful in the current context.