DISABLING A DECREMENT BUTTON AND AN ADD TO CART BUTTON WHEN QTY IS ZERO

Let’s say you have a plus and minus button and you are in shopping cart part of the webpage…you can set the minus button with disabled by default in the html which in the javascript for the addEventListener set the property by default to true…meaning as a Boolean that it’s set to true….

<button class="quantityBtn" id="decrement" disabled>-</button>

But that is not the only fix. You have to enable the decrement button it when your quantity is greater than zero…so what do you do. So here is the next part of the fix you need to set in the javaScript addEventListener (increment button) when the plus button is pressed the minus decrement button as a Boolean becomes false (this means that it’s no longer disabled….as being disabled is true from the original default value from above) starting from a qty of zero. So the minus button will be accessible and can decrement backwards…

increment.addEventListener('click', function(){

    quantity ++

    decrement.disabled = false; //this is where you set it to false so the button will become enabled

    quantityDisplay.innerText = quantity

})

Ok now that we have the buttons set to disabled and enabling the buttons we now need to change the values if you decrement all the way down to zero now disabling the minus (decrement) button so it’s no longer enabled and will not be able to decrement to a negative value with the if statement setting if quantity is === 0 then set the decrement button disabled property to true which will disable the minus (decrement) button from working.

decrement.addEventListener('click', function(){

    quantity--

    if(quantity === 0){

        decrement.disabled = true;

    }

    quantityDisplay.innerText = quantity

})

FOR THE Color panels that we want to reverse the order:

The properties that we are going to target is the display: flex……flex-direction: row-reverse

The display:flex will be for the original container css class which takes the color panels and puts them in a vertical row going from left to right (not stacked on top of each other)

Next we create another css class by itself called .reverse

.reverse{

    flex-direction: row-reverse;

}

The css .reverse will hold the flex-direction: row-reverse; property; we will use this in the javascript for the click event when the short by shade button is clicked the colors will reverse their order.

Inside the javascript click event for the sortBtn we will access the (container) class which holds the color panels and access the **classList** property which takes control of all the classes palette-section blue-1 (etc)

    <div class="container" id="container">

        <div class="palette-section blue-1"></div>

        <div class="palette-section blue-2"></div>

        <div class="palette-section blue-3"></div>

        <div class="palette-section blue-4"></div>

        <div class="palette-section blue-5"></div>

    </div>

…now we are going to add a function to the classList called toggle()…..and inside the toggle function we will place the css class reverse which will basically put the flex-direction: row-reverse appended to container class which has the display:flex property;

const sortBtn = document.getElementById('sort-btn')

const container = document.getElementById('container')

sortBtn.addEventListener('click', function(){

    container.classList.toggle('reverse')

})

Wrap up from whats going on above….first we need to make sure that the list of items (in this case it’s the color panels) all have the same class names ie: palette-section

2. We need to make sure that they are all enclosed inside a container class or some sort of container class that they are not sharing with any othe part of the html.

3. Now we need to add the display property and sort them either flex or what ever way you want to

4. We need to create another css property that has the flex-direction of row-reverse….this way we can access this property via javascript

5. We need to setup an eventlistener for a click event when the sort button is clicked

6. Also make sure we are gaining access to the sortBtn and the container class ie:

const sortBtn = document.getElementById('sort-btn')

const container = document.getElementById('container')

7. Now we need to write code when the sortBtn has been clicked the container class will access the entire list of classes of palette-section by using the classList property and then we need to add another function called toggle()….inside the toggle propery we can add the new class of reverse that we created which will toggle the row to reverse back and forth……basically the toggle function actually adds the reverse class on and when you click the sort button again it removes the reverse class….so on and off

ARIA-LABELS…………..the aria labels are used in place of using labels and for new technology screen readers and does not interrupt your input form design. It basically is for screen readers and will be very helpful for anyone using a screen-reader

aria-label=”Full name”