EVENTS ON THE addEventListener….

//setup event listener for click events for container whenever it gets clicked....but we actually want to target button in order to get to the id

//note that we have the e or event as our parameter..it comes default with the addEventListener so we don't need to do anything like declare it except use the e event

container.addEventListener("click", function (e) {

  console.log(e.target.id);

  console.log(e);

});

Events are any interaction in the browser..like a click, double click, scrolling…etc

So when we log out the event or e and click somewhere in the page we get:

**index.js:42 PointerEvent {isTrusted: true, pointerId: 1, width: 1, height: 1, pressure: 0, …}**

What is this….this is actually an object and when you open it up there will be a lot of stuff going on…but what we actually want is is the **target** section and focus on the **id**  and we will get

**ostrich-pillow………..** this is the id of the first button in the store (if that is where you clicked on)

ALL this is for is to help us eliminate the use for multiple document.getElementById(“name-of-id”)

We can actually just bubble down the dom with the **event** and find our id that way

**NOW WE WANT TARGET THE parentElement**… for the id to use in order to change the css

Ie: you want to change the background of a card you clicked to a different color

To do this we still must use the event type and append to it .parentElement

container.addEventListener("click", function (e) {

  console.log(document.getElementById(e.target.id).parentElement);

});

Here we have the (document.getElementById(e.target.id).parentElement)

This is saying when the event of click happens get the target then to the id then find the parentElement of what the user is clicking…in this case it’s “product”

LOOK BELOW AT THE SCREENSHOT. The div=class”product”…so we are accessing the parent of the button (that is what we actually clicked on) which is a class even though it’s not an id

for (let product of products) {

  productsHtml += `

    <div class="product">

        <h3>${product.name}</h3>

         <h4> £${product.price}</h4>

        <img src="${product.image}">

        <button id="${product.id}">Buy Now</button>

    </div>

    `;

Now we are going to use the parentElement to add on and remove classed to the children

document.addEventListener("click", function (e) {

  console.log(e.target.id);

  //   document.getElementById(e.target.id).parentElement;

  document.getElementById(e.target.id).parentElement.classList.add("read");

  document.getElementById(e.target.id).parentElement.classList.remove("unread");

});

//note that we are targeting the event which we will access the target to the id. Next we will use the parentElement to target the classList and we now can add or remove class names which we can manipulate the css...here we are add to the class list a class of "read"...and inside the css file we have a class of .read with css rules to change the background...note that if you don't use the .parentElement you only will change part of the row ie: if you click on the name Stevie Wonder only that will change to background of white...not the entire row...so you need to add in the .parenteElement in order to target the entire list of child elements in that row....so the parent will actually be targeted not the child...meaning the whole row will be affected.  So note that the classList propery is being used here...below is part of the html..note the list of classe inside the div with "message" and "unread" that is the parent now the classes: "from" "message-text" and "date" are the children...we don't want to only target them individually..that is why we added .parentElement which targets "message unread" in the parent div which when you click any part of the row the whole row gets changed not only the name or the message-text by itself

//  <div class="message unread">

//     <div class="from" id="message-from-1">

//       Stevie Wonder

//     </div>

//     <div class="message-text" id="message-text-1">

//       I just wrote to say I love you❤️

//     </div>

//     <div class="date" id="message-date-1">

//       28 Aug

//     </div>

//   </div>;