ACIT 1620 - FUNDAMENTAL WEB TECHNOLOGIES

WEEK 12

NEDA CHANGIZI

Today's Learning Outcomes

- Explain what events are in JavaScript
- Add event listeners to listen and respond to events
- Use the concept of event delegation and event object to listen and respond to events more efficiently

JS and DOM Events

- What are some examples of events that can happen in the browser?

- Activity – Have a button in your HTML code with text "Click Me!". It should display an alert to the user when it is clicked.
  1. Which element is responsible for a reaction?
     - Get access to the element (in our example, the button) that is supposed to react to an event listen for events.
  2. How is it going to react to the event?
     - Write a function, to be executed and handle the events when the event happens (in our example, displaying a message)

Event Handling – 3 ways

- HTML Attribute (inline event handler)

JS code

```
<button onclick="alert('Button pressed')">Press me</button>
```

- DOM Element Property

```
let btn = document.querySelector("button");
function buttonPressed ()
{
    alert("Button Pressed")
}
btn.onclick = buttonPressed;
```

Note:
() not used

- addEventListener()

element.addEventListener(*event*, *function*);

```
let btn = document.querySelector("button");
btn.addEventListener('click', buttonPressed);
function buttonPressed ()
{
    alert("Button Pressed")
}
```

Event Handling – Which method to use?

- addEventListener() has some advantages:
  - There is a counterpart function, removeEventListener()
  - Multiple handlers can be registered for one event:

```
myElement.addEventListener('click', functionA);
myElement.addEventListener('click', functionB);
```

  - Whereas using "onclick" property overwrites any existing handler:

```
myElement.onclick = functionA;
myElement.onclick = functionB;
```

Only functionB is be executed when the element is clicked

- However addEventListener() is not supported in some older browsers (older than IE 8).

Activity – remove event handler

- Update the previous activity to display the alert only the first time that the button is clicked.

Activity – multiple event handlers

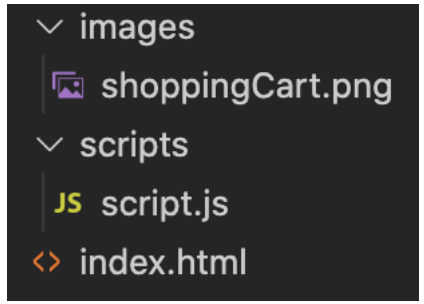
- Add two more event listeners to the button from the previous activity:
  1. Change the background of the page to pink when the button clicked
  2. Change the button's text to say "clicked!" and then change the text on the button back to "Click Me!" when the button is pressed again.

Activity – Updating DOM img attributes

- Update your HTML code:

```
<body>
    <img id="shoppingCart">
    <button>Click Me!</button>
</body>
```

- Add another event listener to the button from the previous activity to find the img element with id="shoppingCart" and update its src, alt, width, and height attribute. Save the image (https://cdn-icons-png.flaticon.com/512/263/263142.png) in the following folder structure



Writing event handlers more compactly with anonymous functions

```
let btn = document.querySelector("button");
function buttonPressed ()
{
    alert("Button Pressed")
}
btn.onclick = buttonPressed;
```

```
let btn = document.querySelector("button");
btn.onclick = function ()
{
    alert("Button Pressed")
};
```

```
let btn = document.querySelector("button");
btn.addEventListener('click', buttonPressed);
function buttonPressed ()
{
    alert("Button Pressed")
}
```

```
let btn = document.querySelector("button");
btn.addEventListener('click', function ()
{
    alert("Button Pressed")
});
```

How do you remove the event listener?

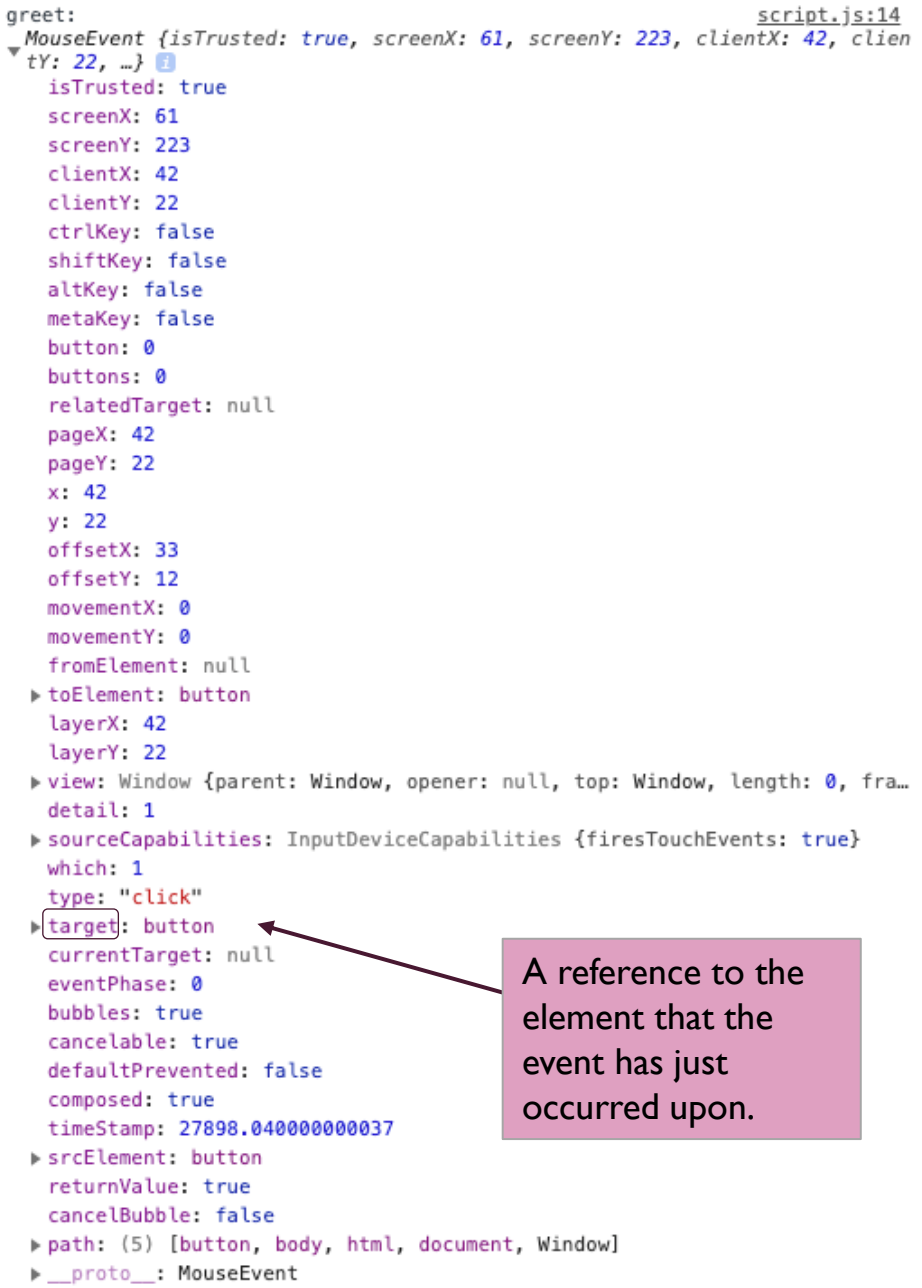
Event Object

- Automatically passed as parameter to event handlers to provide extra features and information about the event.

```
let btn = document.querySelector("button");

function greet(event){
    console.log('greet:', event)
}
btn.addEventListener('click', greet)
```

```
<body>
    <button>Click!</button>
    <script src="scripts/script.js"></script>
</body>
```



Activity – Event Object .target

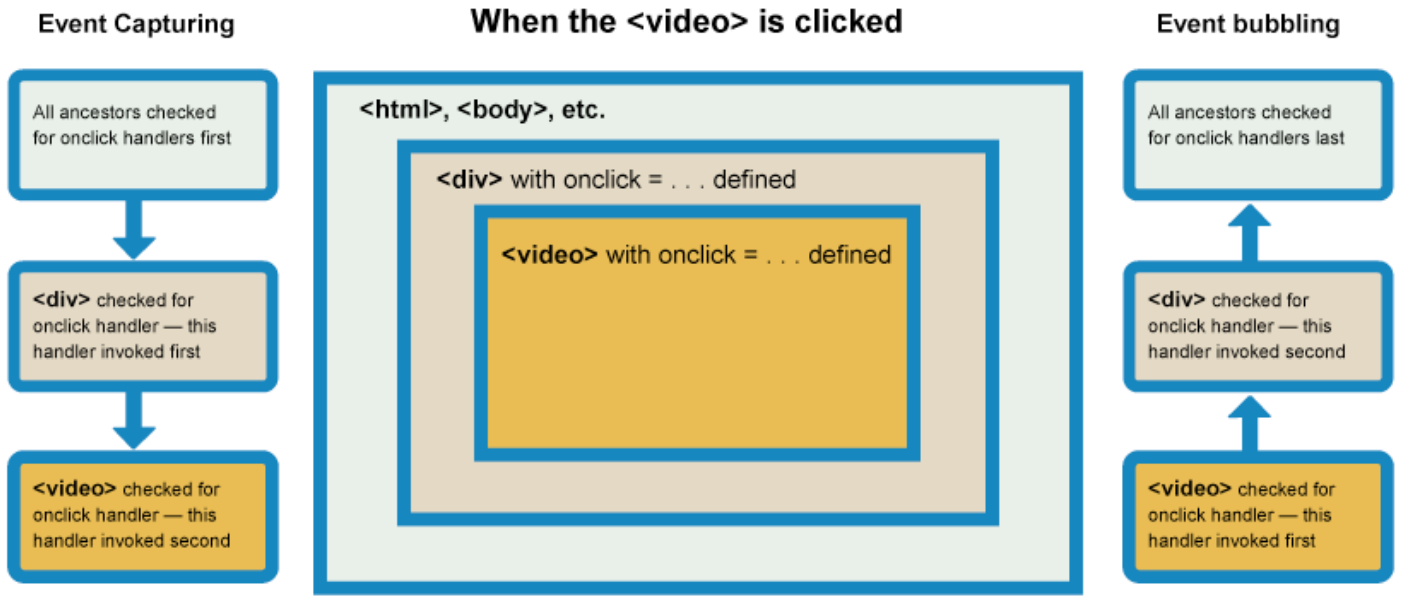
- Add another event handler to the previous activity to create and add a button (with title "purple") and a paragraph (with some text) to the HTML document when the button is pressed.
- Add event handlers to the paragraph and the button to use one handling function to change the element's background color to green when the cursor is moved onto it.

Activity – Event Object .target - cnt'd

- Update the previous activity to have a loop to add 4 buttons with texts: "purple", "red", "blue", and "pink" to the HTML page (instead of only the "purple" button) and attach an event listener to each of them.
- Do we need to update the event listener function to change background color of each button to green when the cursor is moved onto them?

Event Bubbling and Capturing

- In modern browsers, by default, all event handlers are registered for the bubbling phase.




```
element.addEventListener(event, function, useCapture)
```

Optional (Default = false).
A Boolean value that specifies whether the event should be executed in the capturing or in the bubbling phase.

Activity – Bubbling phase

- Update the previous example to have a and in the loop nest the 4 buttons under the . Now make the listen for two kinds of events:
  - The same mouse over event from before to turn the background of the target button to green
  - “click” events to change the text color of the paragraph based on the button that was clicked.