

JavaScript Events

and the Browser

Event Model Challenges

- There are two event models:
 - The one used by IE prior to v9
 - The one defined in the standard and used by everyone else
- There's also a DOM Level 0 model which is what we've seen so far in examples.
 - onclick
 - onsubmit
 - Many others

Event Capture or Bubble

- Event handling differs fundamentally on whether events are captured or bubbled.
- Event capture means that when a link is clicked, the `<body>` element gets the first opportunity to handle the event, and then continues to pass the processing of the event down through the DOM tree eventually getting to that link.
- Event bubbling means that handling begins at the element and bubbles up through the DOM tree, ending at the document (body) level.

W3C Model & Old IE

- The W3C model supports both event capture and bubble, you choose the model when you execute the function.
- The old IE model supports only bubbling.
- In this class, we will focus primarily on the W3C model and largely ignore the old IE model. (You should still understand the difference though!)
- There are other ways to make things work in older browsers.

addEventListener()

- The `addEventListener()` method of the window object is used to indicate that a given HTML element should "listen" or be ready to handle, an event.
- Basic syntax:
- `addEventListener(event,function,capture/bubble);`
- The final argument is optional, so you can see this:
- `addEventListener(event,function);`

Event Handling

- Here is an example of adding an event listener for a submit button called "myBtn":

```
myBtn.addEventListener("submit",myFunction(),false);
```

- This tells the button to listen for the submit event, when it receives the submit event, the function "myFunction()" should be executed.
- The "false" indicates that this event listener will use bubbling.
- Change "false" to "true" to indicate event capture instead.

Removing an Event Listener

- Just as event listeners can be added, so too can they be removed.
- Same essential syntax as adding:
 - `removeEventListener(event, function, capture/bubble)`

Interesting Events

- There are numerous events that can be handled by JavaScript.
- Body/Window related:
 - load & unload
- Mouse/Tap:
 - click, dblclick
 - mouseover, mouseout, mousemove, mousedown, mouseup
- Typically form related:
 - blur & focus
 - change
 - keydown, keyup, keypress
 - submit
- Many more: <https://developer.mozilla.org/en-US/docs/Web/Events>

Event Demo

- Code adds event listeners to various HTML elements, calling a function appropriate for each different event/target.
- Note the "event" that triggered the function call is passed as an argument to the function automatically.