Events

- Events occur when a user interacts with a web page.

Event Listeners

- They are like setting a notification to alert you when something happens. Instead of the program having to constantly check to see if an event has occurred, the event listener will let it know when the event happens, and the program can then respond appropriately.
  - JavaScript, on the other hand, uses a non-blocking approach that uses event listeners to listen out for any clicks on the page.
  - Event listeners are added to elements on the page and are part of the DOM.
  - The original way of dealing with events in the browser was to use inline attributes that were added directly into the markup.
  - Another method is to use the event handler properties that all node objects have.

The Event Object

- Whenever an event handler is triggered by an event, the callback function is called.
- The type property returns the type of event that occurred, such as *click*
- The target property returns a reference to the node that fired the event.
- There are a variety of ways to find the position of where a mouse event occurs:
  - The screenX and screenY properties show the number of pixels from the left and top of the screen respectively where the event took place.
  - The clientX and clientY properties show the number of pixels from the left and top of the client that is being used (usually the browser window).
  - The pageX and pageY properties show the number of pixels from the left and top, respectively, where the event took place in the document. This property takes account of whether the page has been scrolled.

Types of Events

- the click event that occurs when a mouse button is clicked.
- There are also the *mousedown* and *mouseup* events.
- There is also the *dblclick* event, which occurs when the user doubleclicks.
- The mouseover event occurs when the mouse pointer is placed over the element to which the event listener is attached
- the mouseout event occurs when the mouse pointer moves away from an element.
- The mousemove event occurs whenever the mouse moves.
- The keydown event occurs when a key is pressed and will continue to occur if the key is held down.

- The keypress event only occurs for keys that produce character input (plus the Delete key).
- The keyup event occurs when a key is released.
- Pressing the modifier keys such as Shift, Ctrl, Alt and meta (Cmd on Mac) will fire the keydown and keyup events, but not the keypress event as they don't produce any characters on the screen.
- The touchstart event occurs when a user initially touches the surface.
- The touchend event occurs when a user stops touching the surface.
- The touchmove event occurs after a user has touched the screen then moves around without leaving.
- The touchenter event occurs when a user has already started touching the surface, but then passes over the element to which the event listener is attached.
- The touchleave event occurs when the user is still touching the surface, but leaves the element to which the event listener is attached.
- The touchcancel event occurs when a touch event is interrupted, such as a user's finger moving outside the document window, or too many fingers being used at once. A pop-up dialog will also cancel a touch event.

- An event listener can be removed using the removeEventListener() method.
- preventDefault() is a method of the event object that can be used inside the callback function to stop the default behavior happening.
- Event propagation is the order that the events fire on each element. There are two forms of event propagation: bubbling and capturing.
- Bubbling is when the event fires on the element clicked on first, then bubbles up the document tree, firing an event on each parent element until it reaches the root node.
- Capturing starts by firing an event on the root element, then propagates downwards, firing an event on each child element until it reaches the target element that was clicked on.
- The bubble phase can be stopped from occurring by adding the event.stopPropagation()
- Event delegation can be used to attach an event listener to a parent element in order to capture events that are triggered by its child elements.
- A better way is to attach the event listener to the parent <ul> element, then use the target property to identify the element that was clicked on.
-