

Module 3 - Lecture 8

JavaScript Event Handling



Event Programming



Types of Events

<https://developer.mozilla.org/en-US/docs/Web/Events>

Mouse Events

1. `click` - a user has clicked on the DOM element
2. `dblclick` - a user has double clicked on the DOM element
3. `mouseover` - a user has moved their mouse over the DOM element
4. `mouseout` - a user has moved their mouse out of the DOM element

Input Events

1. `keydown` - a user pressed down a key (including shift, alt, etc.) while on this DOM element
2. `keyup` - a user released a key (including shift, alt, etc.) while on this DOM element
3. `change` - a user has finished changing the value of this input element
4. `focus` - a user has selected this input element for editing
5. `blur` - a user has unselected this input element for editing



Listening for DOM Events

Requires 3 things:

1. A DOM element to listen to events on.
2. A specific event that we want to listen for.
3. A function that we wish to execute if the event occurs.
 - This is called an event handler.

NOTE:

- DOM elements can have multiple event handlers, even for the same event type.



Adding Event Handler

HTML

```
<h1 id="greeting">Hello</h1>  
<button id="change-greeting">Change Greeting</button>
```

JavaScript function to execute

```
function changeGreeting() {  
    let greetingHeader = document.getElementById('greeting');  
    greetingHeader.innerText = 'Goodbye';  
}
```

Add Event Handler

```
changeButton.addEventListener('click', (event) => {  
    changeGreeting();  
});
```



The UIEvent Object

| Property | Found In | Purpose |
|---|---------------------------|--|
| currentTarget | All events | Holds the element that the event was triggered on, ie. the button clicked or the select box that changed |
| clientX | Mouse events | The X coordinate on the screen of the click |
| clientY | Mouse events | The Y coordinate on the screen of the click |
| altKey, metaKey, ctrlKey, shiftKey | Mouse and Keyboard events | A boolean on whether the specified key was pressed down during the event |
| key | Keyboard events | The key that was pressed, taking the Shift key into account. Arrow keys show up as 'ArrowRight', 'ArrowDown', 'ArrowLeft', and 'ArrowUp' |

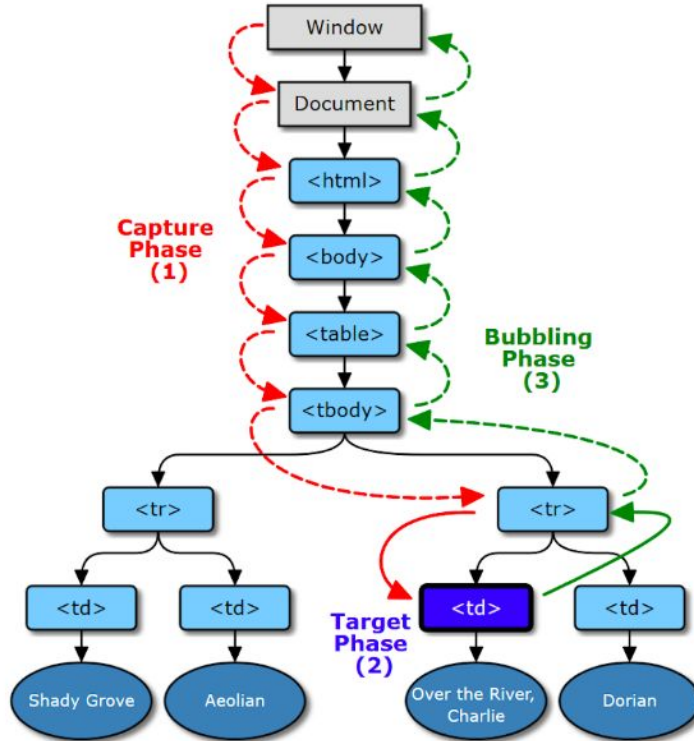


When to add events?

```
document.addEventListener("DOMContentLoaded", () => {  
    // Register all of your event listeners here  
});
```



Event Propagation



Propagation can be stopped by calling `stopPropagation()`



Default Events

- Links and Forms have default behavior that the browser will perform if not stopped.

```
aLink.addEventListener('click', (event) => {  
  // Tells the browser to not perform its normal action  
  event.preventDefault();  
  
  // Then call the event handler  
  toggleImage(event.currentTarget);  
});
```

QUESTIONS?

