

MODULE 4

Event Handling



Event Handling

Mouse events [🔗](#)

Event Name
<code>auxclick</code>
<code>click</code>
<code>contextmenu</code>
<code>dblclick</code>
<code>mousedown</code>
<code>mouseenter</code>
<code>mouseleave</code>
<code>mousemove</code>
<code>mouseover</code>
<code>mouseout</code>
<code>mouseup</code>
<code>pointerlockchange</code>
<code>pointerlockerror</code>
<code>select</code>
<code>wheel</code>

Keyboard events [🔗](#)

Event Name
<code>keydown</code>
<code>keypress</code>
<code>keyup</code>

Focus events [🔗](#)

Event Name
<code>focus</code>
<code>blur</code>

Form events [🔗](#)

Event Name
<code>reset</code>
<code>submit</code>

Event handlers allow our code the ability to react when an event occurs.

Three Pieces to the Puzzle

1. A DOM element where we want to listen for events
2. A specific event that we want to listen for
3. A function that holds the logic that we want to execute

Event Design Pattern (Named)

```
let changeButton = document.getElementById('change-greeting');
```

1. A DOM element where we want to listen for events

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

2. A specific event that we want to listen to

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

3. A function that holds the logic that we want to execute

Event Design Pattern (Anonymous)

- the name of the event
- the type of data structure used to represent key properties of the event
- the object that will 'emit' or 'publish' the event

1. A DOM element where we want to listen for events

```
let p = document.getElementsByTagName('p')[0];
```

```
p.addEventListener("click", function(event) {  
    this.innerHTML = "Paragraph Clicked!";  
});
```

2. A specific event that we want to listen to

3. A function that holds the logic that we want to execute

Event 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'

Event Propagation

```
<div id="foo">  
  <p>Inner mind</p>  
</div>
```

```
let p = document.getElementsByTagName('p')[0];  
p.addEventListener("click", function(event) {  
  this.innerHTML = "Paragraph Clicked!";  
});  
let d = document.getElementsByTagName('div')[0];  
d.addEventListener("click", function(event) {  
  this.innerHTML = "Div Clicked!";  
});
```

Event Propagation

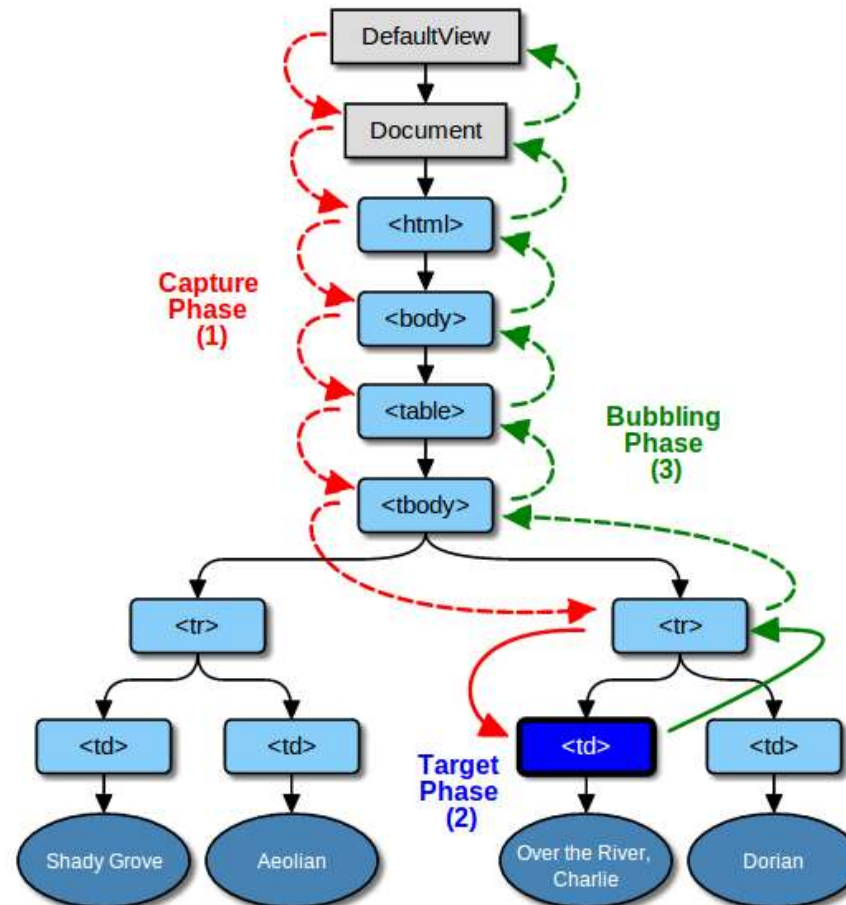


Fig. 1 Graphical representation of an event dispatched in a DOM tree using the DOM event flow

When to add listeners

- Need to wait until the DOM is fully loaded

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

LET'S CODE!



ELEVATE  YOURSELF

WHAT QUESTIONS DO
YOU HAVE?



Reading for tonight: **API Web Services**

