

DOM events



Objectives :

- Attach event handlers to DOM elements
- Modify the DOM in response to an event
- Use callbacks in methods like `addEventListener`
- Explain the difference between `this` and `event.target` in event listeners
- Respond to the event `DOMContentLoaded` event

Review: making a directory and files from the command line

(you tell me)

How can I efficiently create a new directory that includes and **index.html**, **app.js**, and a **style.css** file?

WIRE IT UP

EVENTS?

In your console on any webpage...

```
monitorEvents(window)
```

What do you see?

Review: how do you view & access properties on the #document object?

- what events do you see?

Targeting elements in the DOM

(jQuery will make quick work of this, but how do you do it from scratch?)

...so many ways!

```
<button id='myButton' class='buttonUp'>Click Me!</button>
```

```
var btn = document.querySelector('button');  
// OR  
var btn = document.getElementsByClassName('buttonUp')[0];  
// what does this return??  
// OR  
var btn = document.getElementById('myButton');  
// OR  
var btn = document.querySelectorAll('button');  
// OR  
var btn = document.getElementsByTagName('button')[0];  
// what does this return?
```


Now, we've targeted a node, so let's attach an event listener!

addEventListener(s)

- addEventListener
- removeEventListener
- **these require a callback function**

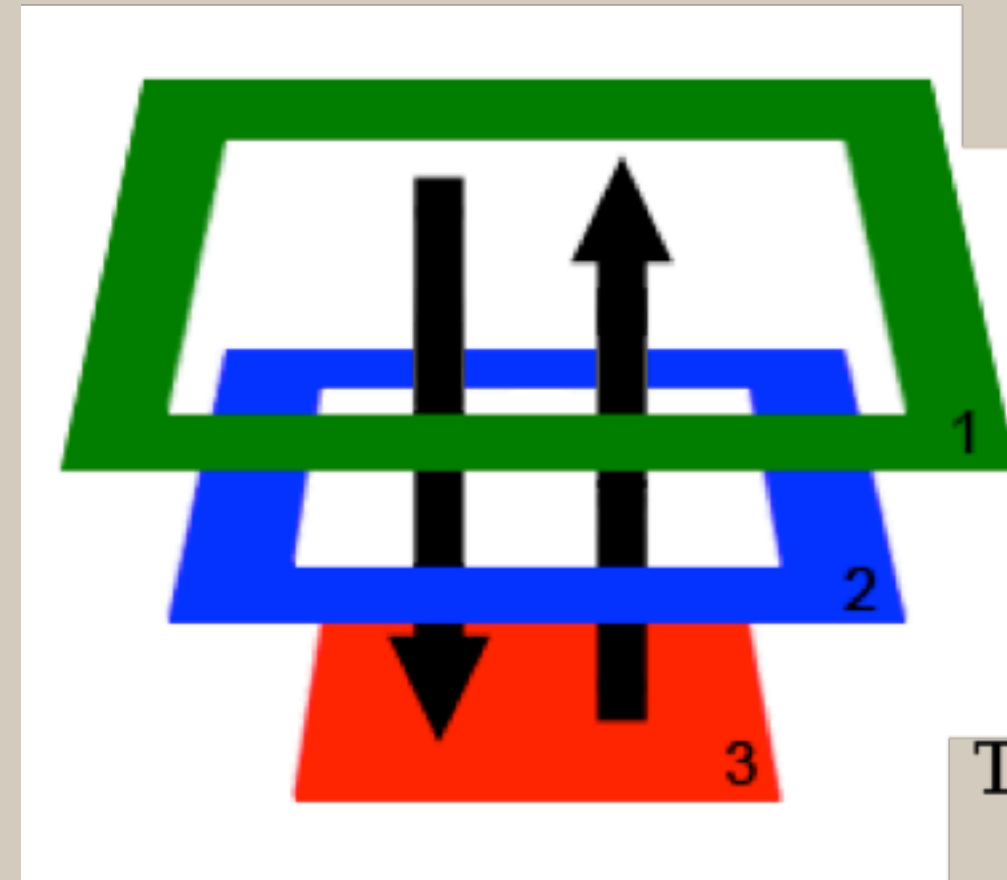
eventListener Anatomy

```
// selector
var button = document.getElementById( 'myButton' );

// example with anonymous callback function
button.addEventListener( 'click', function() {
    alert( 'someone clicked something somewhere' );
})

// OR define a REUSABLE function
var alertMe = function() {
    alert( 'someone clicked something somewhere' );
}
button.addEventListener( 'click', alertMe)
```

Event Propagation



The topmost element

The innermost element

another view...

Meet the Phases



THIS vs.
event.target ???

DOMContentLoaded?

fin

