Week 10

JAVASCRIPT EVENTS

Listening for an event

```
function doSomething (event) {
    // handle the event
}

// BAD
<body onclick="doSomething()">

// BAD
document.body.onclick = doSomething

// GOOD
document.body.addEventListener('click', doSomething)
```

Common types of events

http://en.wikipedia.org/wiki/DOM_events

Mouse events:

click, dbclick, mousedown, mousemove, mouseup ...

Keyboard events:

keydown, keyup, keypress ...

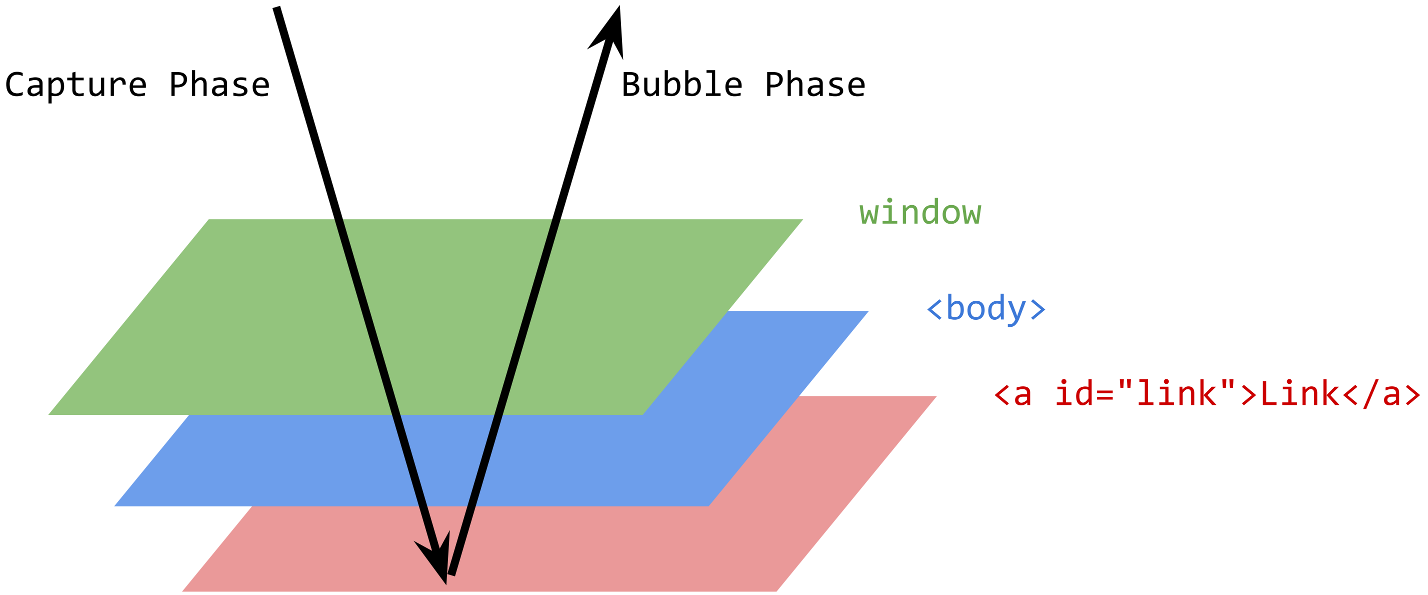
Window events:

load, resize, scroll ...

Form Input events:

change, focus, blur ...

How events actually work



It bubbles up. Which means...

If you click on `#link` , `window` will get notified as well. And you can check if the click event was triggered on `#link` or not by checking the event object's target property.

```
window.addEventListener('click', function (event) {
    if (event.target.id === 'link') {
        console.log('Clicked on Link and bubbled to the window!')
    }
})
```

What if I don't want it to bubble?

You can stop an event from bubbling up by using `stopPropagation()`

```
var link = document.getElementById('link')

link.addEventListener('click', function (event) {
    event.stopPropagation()
})

window.addEventListener('click', function (event) {
    if (event.target.id === 'link') {
        // You will no longer get here
    }
})
```

You can also disable things…

You can stop the default behavior of an event with `preventDefault()`

```
var input = document.getElementById('input')

input.addEventListener('keydown', function (event) {
    event.preventDefault()
    // Nothing can be typed into this input field now
})
```

Event Delegation

It's tedious to add listeners to a lot of elements when they all behave the same way. Also, what if some elements are created AFTER we registered the listeners?

Luckily, since events bubble all the way up, we can simply attach one single listener to the parent element, and check `event.target`.

```
parent.addEventListener('click', function (event) {

    if (event.target.classList.contains('button')) {
        console.log('You clicked on a button!')
    }

})
```

Of course, jQuery makes it even easier…

```
$('#parent').on('click', '.button', function () {

    // Only clicks on elements with class="button"
    // will trigger this function

})
```