



RICE[®]

Web Development

COMP 431 / COMP 531

Events and DHTML

Scott E Pollack, PhD

January 26, 2016

Recap

- HTML
- JavaScript
- Forms
- CSS
- Homework Assignment 2 (Dynamic Page)
 - Due **Thursday** 1/28

~~9/3 CSS~~
9/8 Events
9/10 HTML5
9/15 Scope

Homework Assignment 3
(Draft Front-End)
Due Tuesday 2/9

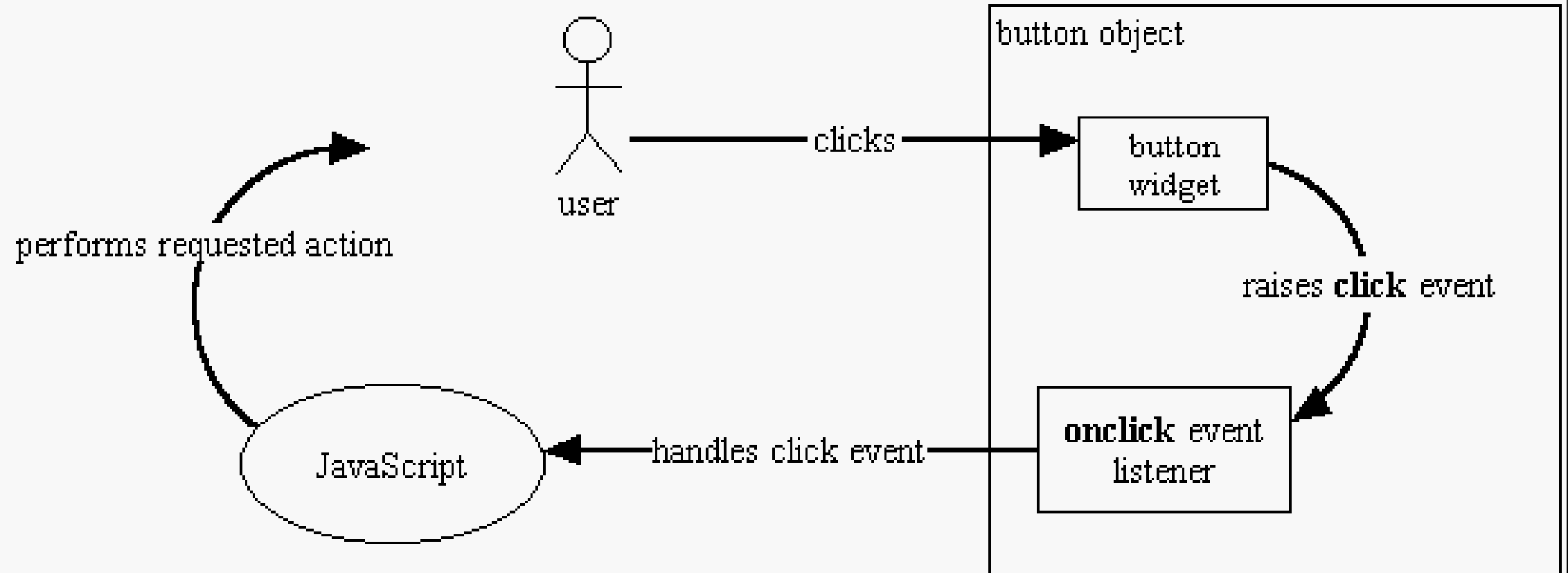
Homework Assignment 4
(JavaScript Game)
Due Thursday 2/18

Simple and Dynamic Page Thoughts

- What's the purpose?
 - Experience in web dev
 - Trial and error
 - Testing and debugging skills
- Computer Programming
 - Variable names
 - Variable scope
 - Indentation
 - White-space (watch inlining!)
 - DRY code
- Comments: why not what
- Use braces for blocks if { } else { }
- Use variables for repeated values
- Do not assume anything
 - Query parameter order
- Write defensive code
- Learn Regular Expressions
- Page Layout – good looking forms
- Fix your images sizes
- No non-CSS styling
- More than the bare minimum

Events

```
<button value="Click Me" onclick="alert('Thank you')" />
```



A Brief History of Events

- 1995 – DOM Level 0 Events [Netscape]
- 1997 – Microsoft Event Model [IE4 – IE8]
 - Coined the term Dynamic HTML (DHTML)
- 2000 – DOM Level 2 Events [W3C]
- Sep 2010 – DOM Level 3 Events [W3C, IE9]
- Dec 2013 – MS Event Model dies [IE11]

2015	<u>Chrome</u>	<u>IE</u>	<u>Firefox</u>	<u>Safari</u>	<u>Opera</u>
July	63.3 %	6.5 %	21.6 %	4.9 %	2.5 %

	Total	IE 11	IE 10	IE 9	IE 8	IE 7
July	6.5 %	3.4 %	0.8 %	1.0 %	0.9 %	0.2 %

DOM Level 0 Event Plumbing

onclick truthy says to execute default action or not
preventDefault()



- Inline (obtrusive)

```
<a href="?q=true" onclick="return clickMe()">
```

- Traditional (unobtrusive)

```
window.onload = function() {  
    var q = location.search.replace("?q=", "");  
    if (q) {  
        changeResponse("You clicked the link with the box checked");  
    }  
  
    // add event handler for pressMe button  
    document.getElementById("pressMe").onclick = function() {  
        changeResponse("You pressed the button!")  
    }  
}
```

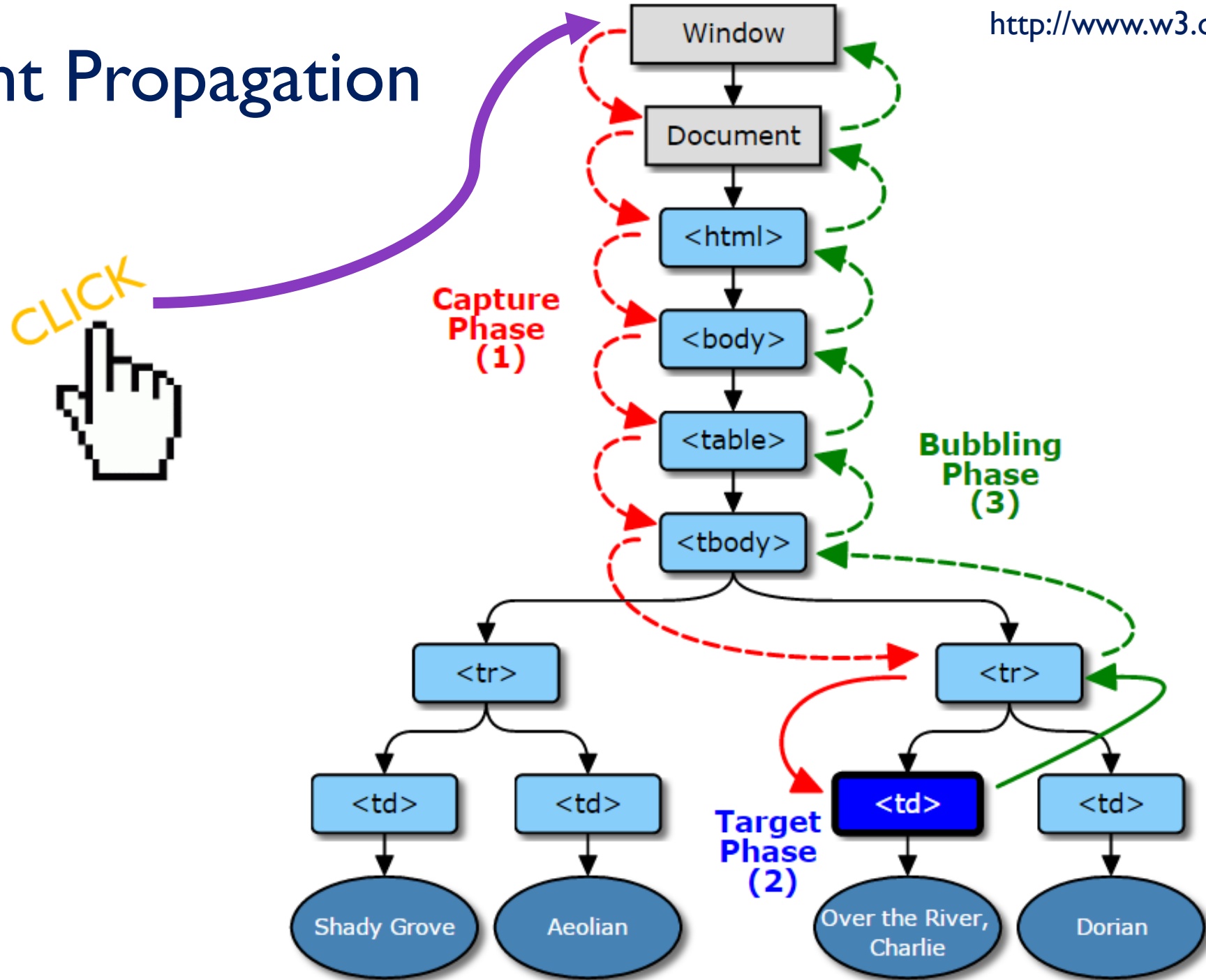
DOM Level 2 Event Plumbing

- Allows for multiple registration of handlers
 - level 0 only has one handler – multiplex?
- Allows for removal

```
// add level 2 handlers
var l2p = document.getElementById("level2p")

l2p.addEventListener("click", l2pResponse, false);
l2p.addEventListener("click", l2pColorFn, false);
l2p.addEventListener("click", l2pBgColorFn, false);
l2p.removeEventListener('click', l2pBgColorFn)
```

Event Propagation



Event Types: on<type>

- blur / focus (element loses / receives focus)
- change (form field value changes)
- click (mouse down and up on one element)
- contextmenu (right-click)
- keydown / keyup (depress / release a key, repeat)
- keypress (character key id depressed, repeats)
- mousedown / mouseup (depress / release mouse button)
- mousemove (mouse in motion)
- mouseover / out (mouse enters / leaves an element)
- mousewheel (wheel motion)
- reset / resize / scroll / copy / paste / submit ...

Demo: Events in Action

<https://www.clear.rice.edu/comp431/sample/hello-events.html>

Demo: Image Object Caching

<https://www.clear.rice.edu/comp431/sample/imageCaching.html>

In-Class Exercise:

Events: “Avoid the Mouse” Game

Make a page that has a button on it.

1. The button says “Click Me”
2. When the mouse approaches the button, the button moves out of the way so it can't be clicked
3. Holding down CTRL prevents the button from moving out of the way
4. When the button is clicked, a `<div>` appears that looks similar to this:
5. The button text changes to “Play Again” and does not move when the user attempts to click it.
6. When clicked, the button text changes to “Click Me” – go to 1.

Congratulations!
You won!

Turnin avoid.html and avoid.js to COMP431-S16:inclass-5