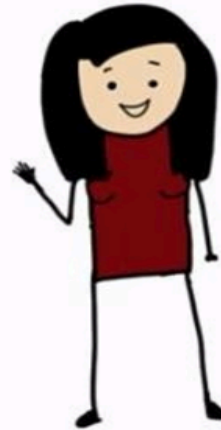


# CSC 337

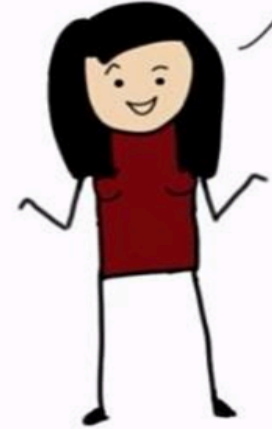
Event Driven Programming  
Rick Mercer

Event-driven programming be like...

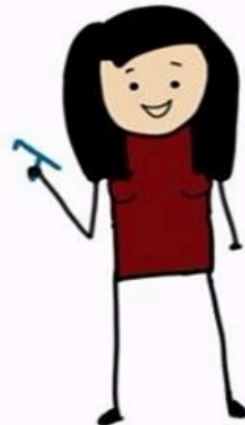
Hey, I just met you



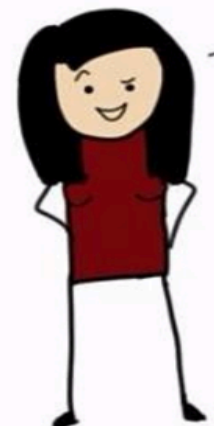
And this is crazy ...



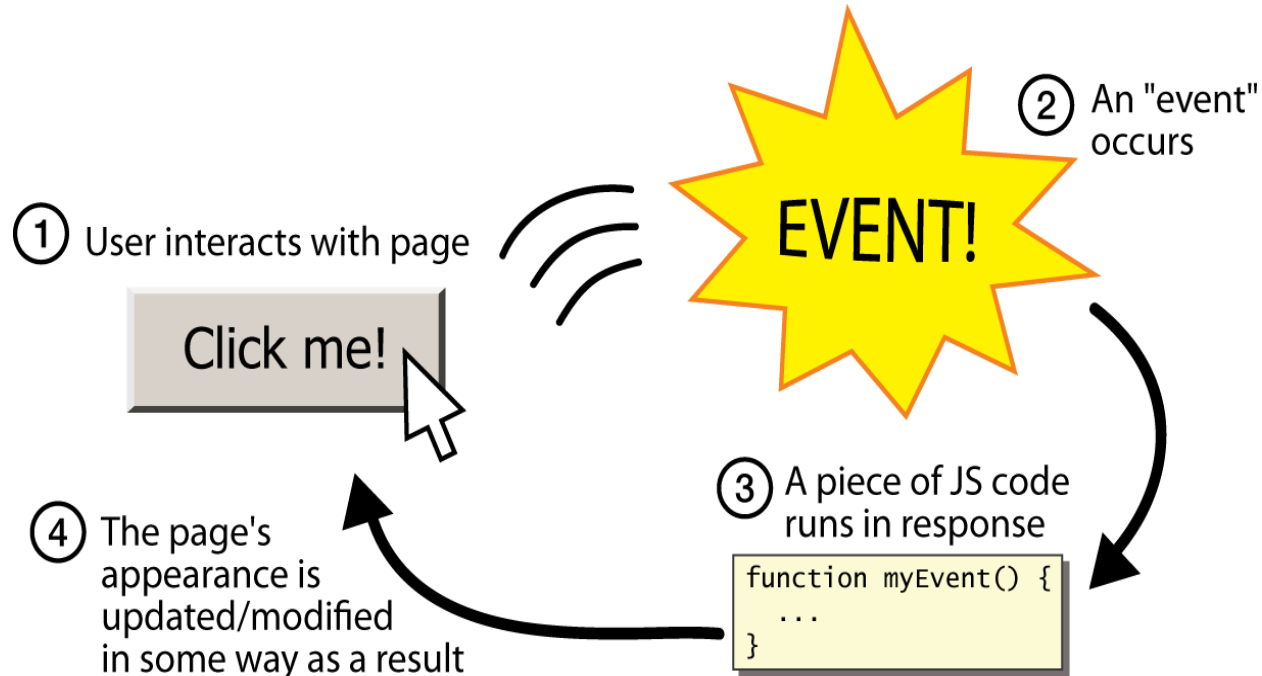
But here's my  
function ...



So callback may be ...



# Event-driven programming



- JS programs have no `main` method to "run"
  - they respond to user actions called **events**
- **Event-driven programming**
  - Writing programs driven by user events

# Types of events

- A few events the occur when people use a website:
  - Click the mouse on a certain element (text or button)
  - Hovering the cursor over a certain element
  - Pressing a key on the keyboard
  - Resizing or close the browser window
  - Web page finishes loading
  - A form gets submitted, which you have already done
  - A video being plays, pauses, or finishes play
  - Input is entered
- Many other events can be ‘captured’ when the user interacts with the web page and a JS function gets called

# HTML Event attributes

- Add an event attribute to an HTML element

- General Form:

- ```
<element attribute= "functionName ( argsoptional ) ">label</element>
```

- Example in an HTML file

- ```
<div onclick="doIt('prompt')">Click this text</div>
```

- There are many attributes

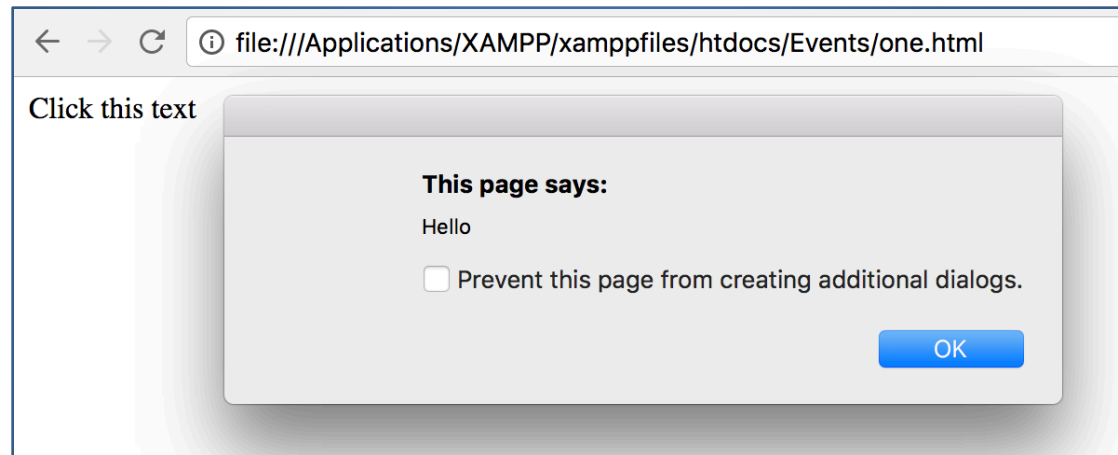
- onerror onload onresize
    - onblur onfocus onreset onselect onchange
    - onkeydown onkeypress onkeyup
    - onclick onmouseover onmouseup onmousedown onmousemove
    - ondrag onscroll
    - oncut onpaste
    - onplay onpause onvolumechange

- [https://www.w3schools.com/tags/ref\\_eventattributes.asp](https://www.w3schools.com/tags/ref_eventattributes.asp)

# Event Handling with HTML and JavaScript

- Example above calls the function named `doIt` by clicking the text `Click this text`
- JavaScript functions can be set as event handlers
  - when you interact with the element, the function executes
- Example in an HTML file

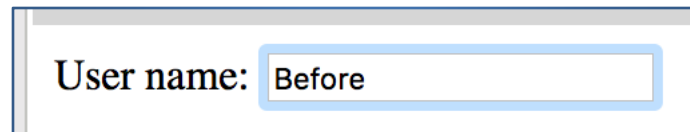
```
<body>
<div onclick="doIt('Hello')">Click this text</div>
<script>
function doIt(str) {
    alert(str);
}
</script>
</body>
```



# An Event Handler with <input>

- Another common event: User presses enter in  
New thing .value  
New thing `<input type="text">`
- This uses a new HTML tag `input` and `type` attribute

`<body>`



User name:

User name: `<input id="findMe" onchange="f3()">`

`<script>`

```
function f3() {  
    element = document.getElementById("findMe");  
    // .value is the entered text  
    document.write(element.value);  
}  
</script>
```

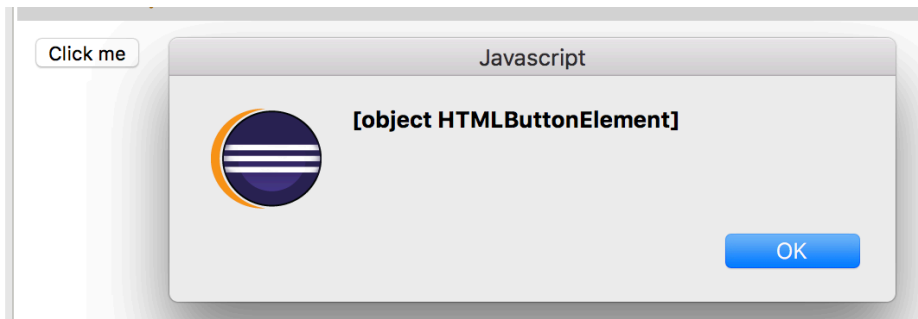


Before

# An Event Handler with <button>

- New HTML tag `<button>` `</button>`
- Button's text 'Click me' appears inside tag
- To make a responsive button or other UI control:
  - Choose the control (e.g. button or text) and event attribute (e.g. onclick or onchange) of interest
  - Write a JavaScript function to run when the event occurs
  - Attach the function to the event as an HTML attribute

```
<button onclick="f7()" id="findMe">Click me </button>
<script>
function f7() {
    button = document.getElementById("findMe");
    alert(button);
}
</script>
```



`id="anID"      getElementById      .value      .innerHTML`

- Finally... let us change only part of a page ... a `<p>` here
- New thing `<input type="password"`

`<h4>Change HTML</h4>`

Password: `<input type="password" id="pswd" size=9>`

`<button onclick="showPassword()">Show password</button>   <br>`

`<p id="toChange">   <!-- The innerHTML -->`

This element will be changed  
to the password entered

`</p>`

`<script>`

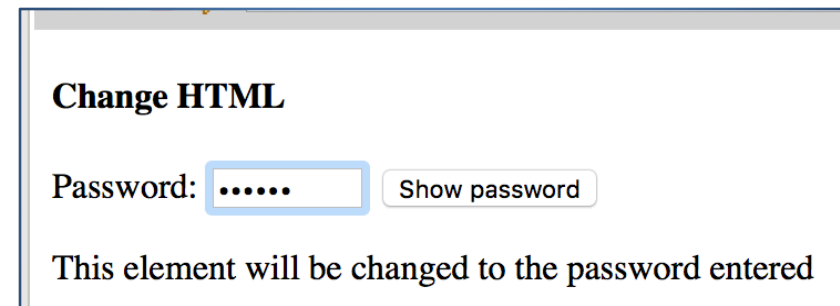
`function showPassword() {`

`var element = document.getElementById('toChange');`

`element.innerHTML = document.getElementById('pswd').value;`

`}`

`</script>`





## onmousemove and passing event as an argument

- Use these things again: `id='anID'` .innerHTML and what we will use a lot:

```
document.getElementById('anID');
```

```
<body>
```

```
<div onmousemove="showWhereTheMouseIs(event)">
```

```
    Move the mouse over me
```

```
</div>
```

```
<div id='point'> </div>
```

```
<script>
```

```
function showWhereTheMouseIs(e) {
```

```
    var element = document.getElementById('point');
```

```
    element.innerHTML='Mouse is at ('+e.pageX+', ' + e.pageY + ")";
```

```
}
```

```
</script>
```

```
</body>
```

Move the mouse over me  
Mouse is at (33, 13)

# Changing styles in HTML elements

- We can change the appearance of an HTML element

```
var element = document.getElementById("findMe");  
element.style.color = "pink";
```

- Same properties as CSS, but with camelCasedName

```
function f6() {  
    var block = document.getElementById("findMe");  
    block.style.color = "pink";  
    block.style.fontFamily = "Courier";  
    block.style.fontSize = "18pt";  
    block.style.backgroundColor = "black";  
    block.style.border = "thick solid #0000FF";  
    block.style.border.color = "blue";  
}
```

**Text the can be found with getElementById("findMe")**

# DOM element objects

- Access/modify the attributes of a DOM object:
  - *objectName.attributeName*
- Most DOM object attributes have the same names as the corresponding HTML attribute
  - `img` tag's `src` property or `a` tag's `href` property

# DOM element objects

```
<button id='button' onclick="f7()">See Attributes</button>
<img id="favImg" src='fav.jpg' alt="my fav">
<script>
function f7() {
    var block = document.getElementById("favImg");
    document.write('block.src: ' + block.src + '<br>');
    document.write('block.alt: ' + block.alt + '<br>');
    document.write(' block.id: ' + block.id + "<br>");
}
</script>
```

See Attributes



block.src: file:///Applications/XAMPP/xamppfiles/htdocs/Events/fav.jpg  
block.alt: my fav  
block.id: favImg

# iframe (needed on Project 4)

- You can load an HTML file into an iframe

```
<iframe id="section" src="intro.html" width="600px" height="400px"> </iframe>
```

```
<!DOCTYPE html>
<html> <!-- file name: int
<head>
<title>Intro.html</title>
</head>
<body>
<h4>This is the HTML page first set as the src
  in an iframe</h4>
</body>
</html>
```

**This is the HTML page first set as the src in an iframe**

# iframe again

```
<button type="button" onclick="setFrameTo(0)">
```

```
0) Calculator
```

```
</button>
```

Click button to change iframe.src <br>

```
<iframe id="section" src="intro.html" width="200px"
height="100px"> </iframe>
```

```
<script>
```

```
var x = [];
```

```
x[0] = "0Calculator.html";
```

```
function setFrameTo(number) {
```

```
    var frame = document.getElementById("section");
```

```
    frame.src = x[number];
```

```
}
```

```
</script>
```

0) Calculator Click button to change iframe.src

TODO: Add JS to create the results of a calculator

# ICA: Remembering things

- Code demo: maintain a list of passwords and show the complete list until a refresh

Password:

1: abc

2: 12345

3: uofa