Events

Adding the Interactivity!!

- It has been up to us to decide when the functions should execute
- It would be better if the functions were called based on special "events"
- The JavaScript API lets us add dynamic function calls!!

Events

- onclick
  - User clicks on an HTML element
- onmouseover
  - User moves the mouse over an HTML element
- onresize
  - browser window is resized
- onload
  - browser finishes loading the page

How it works

- Any element can react to an event.
- You need to add the event to the tag and include what you want to happen

```
<div onclick = "message()"> Clicking on this Div
will invoke a JavaScript function</div>
```

Using Quotes

- You can use single quotes or double quotes for the event result
- Double quotes make it easier if you want to pass String parameters
- Be careful of copying and pasting quotes!

```
<div onclick = "message('Hi')">
```

Example

- Events – Basic Example
- Events – Basic Date Example

Events Change the Program Flow

- Some programs ran in a linear order (step-by-step)
- Events cause the program to "run continuously" since the DOM is always listening for events

More Events

- **Mouse Events**
  - onclick, ondblclick, onmousedown, onmouseenter, onmouseleave, on mousemove, onmouseout,….
- **Keyboard Events**
  - onkeydown, onkeypress, onkeyup
- **Frame Events**
  - onload, onresize, onscroll, onerror,…
- **Comprehensive list:**
  - https://developer.mozilla.org/en-US/docs/Web/Events

Review

- Without the events, JavaScript would be limited in ability to interact with the DOM
- Events are cool….they are also annoying
- Don't worry about memorizing the different events. As the need arises, look them up

University of Michigan
School of Information