CS1520 Recitation
Week 3
Javascript: Event-Driven Programming

http://cs.pitt.edu/~jlee/teaching/cs1520

Jeongmin Lee (jlee@cs.pitt.edu)
Today

- Javascript
  - Events
  - Events Listener

Many examples are based on w3schools.com and tutorialspoint.com
Events
Event

- HTML events are "things" that happen to HTML elements.
- When JavaScript is used in HTML pages, JavaScript can "react" on these events.

- Common HTML Events

<table>
<thead>
<tr>
<th>Event</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>onchange</td>
<td>An HTML element has been changed</td>
</tr>
<tr>
<td>onclick</td>
<td>The user clicks an HTML element</td>
</tr>
<tr>
<td>onmouseover</td>
<td>The user moves the mouse over an HTML element</td>
</tr>
<tr>
<td>onmouseout</td>
<td>The user moves the mouse away from an HTML element</td>
</tr>
<tr>
<td>onkeydown</td>
<td>The user pushes a keyboard key</td>
</tr>
<tr>
<td>onload</td>
<td>The browser has finished loading the page</td>
</tr>
</tbody>
</table>
Event

- Example

```html
<!DOCTYPE html>
<html>
<body>
<h1 onclick="changeText(this)">Click on this text!</h1>
<script>
function changeText(id) {
    id.innerHTML = "Ooops!";
}
</script>
</body>
</html>
```
Event

- Example

```html
<!DOCTYPE html>
<html>
<body>
<h1 onclick="changeText(this)">Click on this text!</h1>
<script>
function changeText(id) {
    id.innerHTML = "Ooops!";
}
</script>
</body>
</html>
```
The onload and onunload Events

- The **onload** and **onunload** events are triggered when the user enters or leaves the page.

- The **onload** event can be used to check the visitor's browser type and browser version, and load the proper version of the web page based on the information.

- The **onload** and **onunload** events can be used to deal with cookies.

Example:
https://www.w3schools.com/js/tryit.asp?filename=tryjs_events_onload
The onmouseover and onmouseout Events

- The **onmouseover** and **onmouseout** events can be used to trigger a function when the user mouses over, or out of, an HTML element:

```html
<!DOCTYPE html>
<html>
<body>

<div onmouseover="mOver(this)"
    onmouseout="mOut(this)"
    style="background-color:#D94A38;width:120px;height:20px;padding:40px;">
    Mouse Over Me</div>

<script>
function mOver(obj) {
    obj.innerHTML = "Thank You"
}

function mOut(obj) {
    obj.innerHTML = "Mouse Over Me"
}
</script>

</body>
</html>
```

[https://www.w3schools.com/js/tryit.asp?filename=tryjs_events_mouseover](https://www.w3schools.com/js/tryit.asp?filename=tryjs_events_mouseover)
Event Listener
EventListener

- **EventListener** is a function that attached to an **HTML element** and calls a **function** when **specified event** is triggered

```
document.getElementById("myBtn").addEventListener("click", displayDate);
```
EventListener

- You can add many event handlers to one element.

- You can add many event handlers of the same type to one element, i.e two "click" events.

- You can add event listeners to any DOM object not only HTML elements. i.e the window object.
Syntax

element.addEventListener(*event*, *function*, *useCapture*);

- The first parameter is the **type of the event** (like "click" or "mousedown").

- The second parameter is **the function we want to call** when the event occurs.

- The third parameter is a **boolean value** specifying whether to use **event bubbling** (False; inner then outer) or **event capturing** (True; outer then inner). This parameter is optional.
Examples

- **Example1:** Click event attached to a button  
  [https://www.w3schools.com/js/tryit.asp?filename=tryjs_addEventListener_add](https://www.w3schools.com/js/tryit.asp?filename=tryjs_addEventListener_add)

- **Example2:** Attach many listener to an object  
  [https://www.w3schools.com/js/tryit.asp?filename=tryjs_addEventListener_add_many](https://www.w3schools.com/js/tryit.asp?filename=tryjs_addEventListener_add_many)

- **Example3:** Attach to window object (not HTML object)  
  [https://www.w3schools.com/js/tryit.asp?filename=tryjs_addEventListener_dom](https://www.w3schools.com/js/tryit.asp?filename=tryjs_addEventListener_dom)

- **Example4:** Passing a parameter  
  [https://www.w3schools.com/js/tryit.asp?filename=tryjs_addEventListener_parameters](https://www.w3schools.com/js/tryit.asp?filename=tryjs_addEventListener_parameters)
Event Bubbling and Capturing

These are the way you order propagate multiple events in multiple HTML DOM.

E.g: <p> inside <div>
Event Bubbling and Capturing

These are the way you order propagate multiple events in multiple HTML DOM.

E.g: `<p>` inside `<div>` and attached two pop-up events on each box
Event Bubbling and Capturing

Event Bubbling
the inner most element's event is handled first and then the outer

the <p> element's click event is handled first, then the <div> element's click event.

https://www.w3schools.com/js/tryit.asp?filename=tryjs_addEventListener_usecapture
Event Bubbling and Capturing

Event Capturing

the outer most element's event is handled first and then the inner

the <div> element's click event will be handled first, then the <p> element's click event.
Questions?