CS7026

jQuery Events
What are Events?

- jQuery is tailor-made to respond to events in an HTML page.

- Events are actions that can be detected by your Web Application. E.g.
  - A mouse click
  - A web page loading
  - A mouse rolling over an element
  - Submitting an HTML form
  - A keystroke on your keyboard

- When these events are fired you can then respond to them with a custom function to do pretty much whatever you want.

- The term "fires" is often used with events. E.g.: "The keypress event fires the moment you press a key".
Events

- For every `onxxx` JavaScript event available, like `onclick`, `onchange`, `onsubmit`, there is a jQuery equivalent.

- Some other events, like `ready` and `hover`, are provided as convenient methods for certain tasks.

- You can find a complete list of all events in the jQuery Events Documentation.
Common DOM Events

- **Mouse Events:**
  - click
  - dblclick
  - mouseenter
  - mouseleave
  - hover

- **Keyboard Events**
  - keypress
  - keydown
  - keyup
Common DOM Events

- **Form Events**
  - Submit
  - Change
  - Focus
  - Blur

- **Document.Window Events**
  - load
  - resize
  - scroll
  - unload
jQuery Syntax For Event Methods

- In jQuery, most DOM events have an equivalent jQuery method.

- To assign a click event to all paragraphs on a page, you can do this:
  ```javascript
  $\("p\"\).click();
  ```

- The next step is to define what should happen when the event fires. You must pass a function to the event:
  ```javascript
  $\("p\"\).click(function(){
    // action goes here!!
  });
  ```
Some Common Event Methods...

- $(document).ready()
  - The $(document).ready() method allows us to execute a function when the document is fully loaded.

- click()
  - The click() method attaches an event handler function to an HTML element.
  - The function is executed when the user clicks on the HTML element.
  - The following example says: When a click event fires on a `<p>` element; hide the current `<p>` element:
    ```javascript
    $("p").click(function(){
      $(this).hide();
    });
    ```
Some Common Event Methods…

- **dblclick()**
  - Executed when the user double-clicks on the HTML element:

- **mouseenter()**
  - Executed when the mouse pointer enters the HTML element:
    ```javascript
    $('#p1').mouseenter(function()
      alert("You entered p1!");
    });
    ```

- **mouseleave()**
  - Executed when the mouse pointer leaves the HTML element:
Some Common Event Methods...

- `hover()`
  - The `hover()` method takes two functions and is a combination of the `mouseenter()` and `mouseleave()` methods.
  - The first function is executed when the mouse enters the HTML element, and the second function is executed when the mouse leaves the HTML element:
    ```javascript
    $('#p1').hover(function(){
        alert('You entered p1!');
    },
    function(){
        alert('Bye! You are now leaving p1!');
    });
    ```
Some Common Event Methods...

- **focus()**
  - Executed when the form field gets focus:
  - `$("input").focus(function() {
        $(this).css("background-color","#cccccc");
    });`

- **blur()**
  - Executed when the form field loses focus
Some Common Event Methods...

- Keyboard Events:
  - The order of events related to the keyboard events:
    1. `keydown` - The key is on its way down
    2. `keypress` - The key is pressed down
    3. `keyup` - The key is released

- Use the `event.which` property to return which keyboard key was pressed.
Some Common Event Methods...

- **keydown()**
  - Trigger the keydown event for the selected elements:
    ```javascript
    $(selector).keydown()
    ```
  - Attach a function to the keydown event:
    ```javascript
    $(selector).keydown(function)
    ```

```javascript
$("input").keydown(function(){
    $("input").css("background-color","yellow");
});
```
Some Common Event Methods...

*keypress()*

- The *keypress()* method triggers the keypress event, or attaches a function to run when a keypress event occurs.
- The keypress event is similar to the keydown event. The event occurs when a button is pressed down.
- However, the keypress event is not fired for all keys (e.g. ALT, CTRL, SHIFT, ESC). Use the *keydown()* method to also check these keys.

- Count the number of key presses in an `<input>` field:

```javascript
i=0
$"input".keypress(function(){
    $"span".text(i+=1);
});
```
Some Common Event Methods...

- **keyup()**
  - The keyup event occurs when a keyboard key is released.
  - The **keyup()** method triggers the keyup event, or attaches a function to run when a keyup event occurs.
  - Set the background colour of an `<input>` field when a keyboard key is released:
    ```javascript
    $("input").keyup(function()
    {
        $("input").css("background-color","pink");
    });
    ```
Time to practice:

Download the lab files or create your own html page and try to implement the following:

- User double clicks on a button and gets an error message.
- Mouse enters an image and it gets bigger.
- Mouse enters a menu item and the background changes; it leaves the menu item and the background returns to the original colour.
- User selects the submit button for a form and gets a alert acknowledging that they have submitted the form.