

AQUENT

GYMNASIUM

JQUERY: BUILDING BLOCKS

Lesson 5

MATERIALS & ASSIGNMENTS

THIS LESSON'S CORE CONCEPTS

In this lesson you will become fluent in event handling, and apply it to a form validation scenario. jQuery's special handling of the ready event will be emphasized and elucidated.

1. Events are how the browser lets us know that something has happened. We react by setting up event handlers or event listeners. We've already done this with the jQuery "click" method: `$(selector).click(function() {...})`.

2. The click method is a convenience method, meaning it's a simple, specific method that wraps a call to a more complicated, more generic method. In this case, "click" is a convenience method for `$(selector).on('click', function() {...})`, which you can use to set up handlers for any event.

3. There are events for any little thing a user does: moving the mouse, clicking the buttons, typing; there are specialized events for forms; and there are non-user events like when network events complete.

4. Compound events fire alongside their atomic components. For example, a click event is made up of a mousedown and a mouseup. Dragging is made of a mousedown, some mousemoves, and a mouseup.

5. Long-running interactions like mouse movements or click-and-drags are handled as a very fast series of individual events, like frames in a movie.

6. jQuery UI includes interaction helpers like `$.draggable()`, which makes elements draggable and sets up its own custom events. jQuery UI also has an excellent download builder for customizing your download. (Note that this can impact page performance by reducing cacheability, and should only be used when file size can be significantly impacted, as with here.)

7. When the user types in a textarea or input element, the browser fires a series of "propertychange" events. (Older versions of IE use the "input" event instead.) The intuitive-sounding but wrong "change" event only fires after the user has stopped typing and left the box. You can handle these events to get notified as the user types by setting up the appropriate handlers: `$(selector).on('propertychange input', function() {...})`;

8. Adding dynamic behavior to pages often requires duplicating existing elements. There are many approaches to this; for simplicity, we've covered the clone method: Select a representative element, clone it (with or without data and event handlers), tweak it, and append it to the element that should become its parent element. Syntax:

```
var $newElement = $(selector).clone(withDataAndHandlers);  
Make changes.  
$newElement.appendTo(selectorForParentElement);
```

9. Event handler functions take a single argument: the event object. This object includes a great deal of information about the event, like coordinates of a click, or which element was the original target of the event.

10. If an event's targeted element – like the element that you've clicked – does not handle an event, the browser asks its parent element if it would like to handle the event, and so forth. This allows a complex element with many children to handle events in a unified way, and is called “event bubbling.” If this is not the desired behavior, as with this lesson's background element, you should compare the current element to the event's target element:

```
$(selector).click(function(evt) {  
    if (evt.target !== this) return;  
});
```

ASSIGNMENTS

1. Quiz
2. Clear each caption box (both input area and poster caption) when the pane is closed. Note that there are many different approaches to this; as usual, your goals are 1.) To simply accomplish the goal and 2.) To do so with as little code as possible.
3. Add a black/white toggle or radio buttons to each caption. While no extra points will be awarded for creativity, any effective yet interesting interfaces here will merit a smile and a firm handshake.

READING

- jQuery's mouse events documentation
<http://api.jquery.com/category/events/mouse-events/>
- jQuery UI's draggable demo (<http://jqueryui.com/draggable/>) and docs (<http://api.jqueryui.com/draggable/>)
- The Event Delegation best practice
<http://www.sitepoint.com/event-delegation-with-jquery/>
- jQuery Data Attributes
<http://api.jquery.com/data/>
- Data background
<http://www.broken-links.com/2010/11/18/data-attributes-in-html-and-jquery/>