Web Front-End Development
Week 4: jQuery Events

Before browsers settled on using the modern W3C standard event API, events were hard to work with because different browsers used different event APIs. Today's major modern browsers have standardized around the new event APIs but using them is still harder than it needs to be and if you have to support older browsers, then you're still stuck with the same compatibility problem. jQuery provides a mechanism for working with events that's simpler than relying on the document object model and the W3C APIs.
jQuery also works with sets of elements by default so it's very easy to write code that assigns event handlers to groups of objects just by using the results of the jQuery selectors and filters.

Events
http://api.jquery.com/category/events/
jQuery functions handle common events

Document Loading
ready() is called when the DOM is ready to be manipulated
This differs from window.load() which is fired after the web page and all of its contents, including all assets, have been loaded.

Event Handler Attachment http://api.jquery.com/category/events/event-handler-attachment/
There are different ways to attach event handlers to events.
on() attaches an event handler to one or more events for the selected element
off() removes an event handler to one or more events for the selected element

https://repl.it/@aileenjp/jQuery-intro

Example:

```
function imageSwap() {
  $("#logo").attr("src", "images/starwarsgalaxy.jpg");
}

$("document").ready(function () {
      $("#logo").on("click", imageSwap);
});
```

- on() attaches the click event and the handler(imageSwap) to the id logo
- attr() takes the attribute name and value to set it to

Or you can use the click event.

```
$("#logo").click(imageSwap);
```

What if you want the click event on the image to change the image back and forth?

```
function imageSwap2() {
```

```
  if ($("#logo").attr("src") == "images/starwarslogo400.png") {
    $("#logo").attr("src", "images/starwarsgalaxy.jpg");
  } else if ($("#logo").attr("src") == "images/starwarsgalaxy.jpg") {
    $("#logo").attr("src", "images/starwarslogo400.png");
  }
}

$("document").ready(function () {
      $("#logo").click(imageSwap2);
});
```

Keyboard Events http://api.jquery.com/category/events/keyboard-events/

Mouse Events http://api.jquery.com/category/events/mouse-events/
.click() triggered when the user clicks on the selected element
.mouseover() triggered when the mouse enters the selected element or its decedents
.mouseenter() triggered when the mouse enters the selected element
.mouseenter() is only triggered when you're over the selected element, mouseover() is triggered when you're over children elements as well.

.mouseout()triggered when the mouse leaves the selected element or its decedents
.mouseleave() triggered when the mouse leaves the selected element
Same difference between mouseleave() and mouseout() - mouseleave() is only only triggered when you're over the selected element, mouseout()is triggered when you're over children elements as well.

.hover() combines the mouseenter and mouseleave events

- $('#selector').hover(function1, function2);

Example:
Let's change the color of the original trilogy movies when the mouse goes over the original dive, then change them back when the mouse leaves.

```
function fnmouseenter() {
  $("#original li").css("color", "#FFD700");
}

function fnmouseleave() {
  $("#original li").css("color", "white");
}

$("document").ready(function () {
      $("#original").hover(fnmouseenter, fnmouseleave);
});
```

- hover binds two handlers to the matched elements, to be executed when the mouse pointer enters and leaves the elements.

If these two functions will never be called by anything else then we can embed them directly in the hover() function and remove their names. These are called anonymous functions.

```
$("document").ready(function () {
  $("#original").hover(
    function () {
      $("#original li").css("color", "#FFD700");
    },
    function () {
      $("#original li").css("color", "white");
    });
});
```

If you ever want to remove the event handler, use on() to attach it and later you can use off() to remove it. (clicking on the original div removes the mouseenter and mouseleave event handlers)

```
$("document").ready(function () {
  $("#original").on("mouseenter", fnmouseenter);
  $("#original").on("mouseleave", fnmouseleave);
  $("#original").click(
    function(){
      $("#original").off("mouseenter", fnmouseenter);
      $("#original").off("mouseleave", fnmouseleave);
    }
  );
```

Event Object http://api.jquery.com/category/events/event-object/
Even though all the modern browsers now implement the W3C standard event API, there are still some subtle differences between how the browsers pass Event information and properties to Event handlers.
The jQuery unified Event Object ensures cross browser consistency when you're handling event information.

pageX is the mouse x position
pageY is the mouse y position

Lab: Create a web page that uses jQuery to select, filter, and manipulate the page in at least three different ways and respond to at least two different events.