JS Event Handling

What is an event?

WOO!

...now how about in the context of JavaScript?

What is an event?

- A JavaScript event is a callback that gets fired when something happens related to the Document Object Model on your website
- For instance, when an element is clicked, or perhaps hovered over
- An event handler can be attached to an element so that when a specific event happens, a specific "callback" function gets fired

Listening for events - native JS

```
document.
getElementById("someId").onclick = function(){
  alert('you clicked me')
};
```

Listening for events - native JS

```
//reference the document (page we're //currently on)
document.


//lookup an element by ID
getElementById("someId").


//attach an event handler - an anonymous
// function to execute - when the elem is //clicked
onclick = function(){
  alert('you clicked me') };
```

Listening for events - jQuery

jQuery provides an easy to use syntax for listening to JavaScript events and providing functions to execute as callbacks to the event being fired

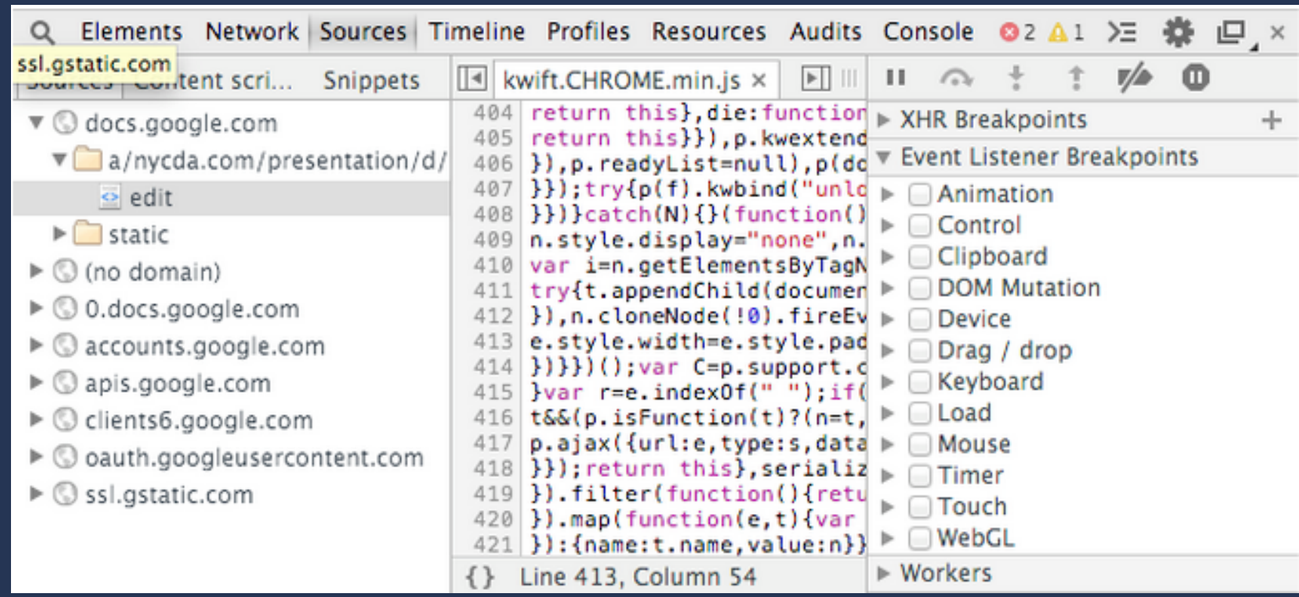
Listening for events - jQuery

```
//this code is for the usual "click" event
$("#someElem").on("click", function(){
  alert('you clicked me!')
});

//we can actually monitor for any custom event
$("#someElem").on("someEvent", function(){
  alert('some event just happened')
});
```

Using Chrome Inspector to intercept events

If you go to Sources -> Event Listener Breakpoints (on the right), you can listen for common events (not custom events), like onclick and onhover.



Firing custom events - trigger

Using jQuery's .trigger() method, we can fire custom events on an element

```
$("#someElem").trigger("someEvent");
```

Exercise

Create a listener for a custom event on an element on boilerplate HTML page

Trigger the event from the console to test out firing your callback

Resources

Exercises: Events