

# Front End Software Development

Front End Technologies (weeks 7 - 12)

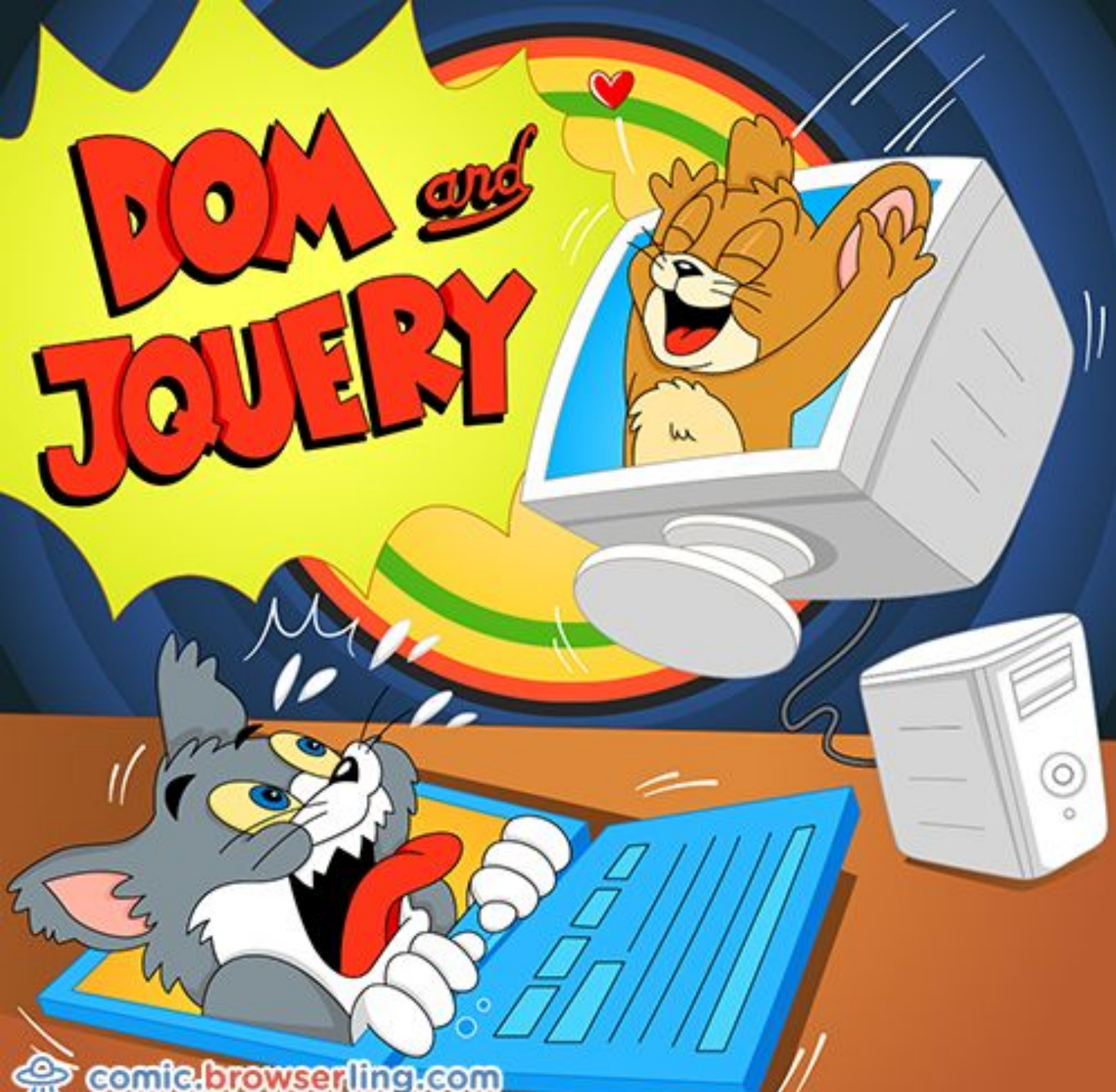
Week 05



# Agenda

- Questions
- Installing and Using JQuery
- Interacting with Elements using JQuery
- AJAX





Questions

# jQuery

## (Installation)

- <https://jquery.com/>
- Installation
  - `npm init -y`
  - `npm install jquery -s`

```
<script  
src="node_modules/jquery/dist/jquery.js"></script>
```

- Note: jQuery is **slowly** being replaced / deprecated.

**WHY?** It is a victim of its own success. Most of its API is now incorporated directly into the browser. But jQuery usage is so widespread, you're guaranteed to run into it for **YEARS** to come. Even if your code doesn't use it directly, some 3<sup>rd</sup> party tool or library likely will.

**IT IS A MUST KNOW!**



# jQuery

*(Usage / Interaction)*

- CSS Selectors
  - id: `#name`
  - class: `.name`
- `$()` === `jQuery()`
- Methods
  - `.text()`, `.val()`, `.html()`, `.css()`, `.attr()`, etc.
    - Most methods, no parameter, GETs current value. Otherwise SETs value.
  - ```
$('p').html() // Hello!
```
  - `.append()`, `.prepend()`, `.before()`, `.after()`, etc.
  - `.remove()` **vs** `.empty()`
  - `.hide()`, `.show()`





# EVENTS

## HANDLERS

.bind( type, data, fn )  
 .one( type, data, fn )  
 .trigger( type, data )  
 .triggerHandler( type, data )  
 .unbind( type, data )

## ERROR

.error( )  
 .error( fn )

## INTERAC

.hover( fnIN, fnOUT )  
 .toggle( fnIN, fnOUT )  
 .blur( )  
 .change( )  
 .click( )  
 .dblclick( )  
 .focus( )  
 .select( )  
 .submit( )  
 .unload( )  
 .unblur( )  
 .unblur( fn )

## MOUSE

.mousedown( fn )  
 .mousemove( fn )  
 .mouseout( fn )  
 .mouseover( fn )  
 .mouseup( fn )

## KEYBOARD

.keydown( )  
 .keydown( fn )  
 .keypress( )  
 .keypress( fn )  
 .keyup( )  
 .keyup( fn )

## WINDOW

.load( fn )  
 .scroll( fn )  
 .resize( fn )

## PAGE

.ready( fn )

## ANIMATE

.stop( )  
 .queue( )  
 .queue( callback )  
 .queue( queue )  
 .dequeue( )  
 .animate( params, duration )  
 .animate( params, options )

# TRAVERSING

## FILTER

.hasClass( class )  
 .filter( expr )  
 .filter( fn )  
 .is( expr )  
 .map( callback )  
 .not( expr )  
 .slice( start, end )

## ACCESS

.each( callback )  
 .size( )  
 .length  
 .get( )  
 .get( index )  
 .index( subject )

## FIND ( expr )

.add( e )  
 .children( e ), .siblings( e )  
 .contents( )  
 .find( e )  
 .next( e ), .nextAll( expr )  
 .parent( e ), .parents( e )  
 .prev( e ), .prevAll( e )

## CHAIN

.andSelf( )  
 .end( )

# MANIPULATING

## INSIDE ( content )

.append( c )  
 .appendTo( c )  
 .prepend( c )  
 .prependTo( c )

## OUTSIDE ( content )

.after( c )  
 .before( c )  
 .insertAfter( c )  
 .insertBefore( c )

## AROUND

.wrap( html )  
 .wrap( element )  
 .wrapAll( html )  
 .wrapAll( element )  
 .wrapInner( html )  
 .wrapInner( element )

## REPLACE

.replaceWith( c )  
 .replaceAll( c )

## CLEAR

.empty( )  
 .remove( e )

## CLONING

.clone( )  
 .clone( true )

# Interacting (jQuery)

```
<p id="welcome">Hello!</p>
<ul>
  <li>Step 1</li>
  <li class="active">Step 2</li>
</ul>
```

```
$(document).ready(function() {
  $('#welcome').html('Goodbye!');
  $('.active').css(
    {'text-decoration':'underline'});
});
```

# jQuery: Cheat Sheet

jQuery API/1.2 <a href="http://jquery.com">http://jquery.com</a>		EVENTS		CORE UI EFFECTS		
SELECTORS		<b>HANDLERS</b> .bind( type, data, fn ) .one( type, data, fn ) .trigger( type, data ) .triggerHandler( type, data ) .unbind( type, data )  <b>MOUSE</b> .mousedown( fn ) .mousemove( fn ) .mouseout( fn ) .mouseover( fn ) .mouseup( fn )  <b>WINDOW</b> .load( fn ) .resize( fn )	<b>ERROR</b> .error() .error( fn )  <b>KEYBOARD</b> .keydown() .keydown( fn ) .keypress() .keypress( fn ) .keyup() .keyup( fn )  <b>PAGE</b> .ready( fn )	<b>INTERACTION</b> .hover( fnIN, fnOUT ) .toggle( fnIN, fnOUT ) .blur() .blur( fn ) .change() .change( fn ) .click() .click( fn ) .dblclick() .dblclick( fn ) .focus() .focus( fn ) .select() .select( fn ) .submit() .submit( fn ) .unload() .unload( fn )  .unblur() .unblur( fn )	<b>SHOW / HIDE</b> .show() .show( speed, callback ) .hide() .hide( speed, callback ) .toggle()  <b>ANIMATE</b> .stop() .queue(), .queue( callback ), .queue( queue ) .dequeue() .animate( params, duration, easing, callback ) .animate( params, options )	<b>SLIDE ( speed, callback )</b> .slideDown( s, c ) .slideUp( s, c ) .slideToggle( s, c )  <b>FADE</b> .fadeIn( speed, callback ) .fadeOut( speed, callback ) .fadeTo( speed, opacity, callback )
#id, tag, .class, * elm1, elm2, elmN ancestor descendant parent > child parent/child prev + next prev ~ siblings  :first :last :not( selector ) :even :odd :eq( index ) :gt( index ) :lt( index )  :contains( text ) :empty :has( selector ) :parent  E[@attr] E[@attr=val] E[@attr^=val] (begins) E[@attr\$=val] (ends) E[@attr*=val] (contains) E[@attr=val][@attr=val] (both)  :nth-child( index ) :first-child :last-child :only-child :input :text :password :radio :checkbox :submit :image :reset :button :file :hidden						
CSS		ATTRIBUTES		HTML		
.css( name, value ) .css( properties )  .height( value ) .width( value )  .addClass( class ) .removeClass( class ) .toggleClass( class ) .offset()	.attr( name ) .attr( key, value ) .removeAttr( name )  .attr( properties ) .attr( key, function )					
CSS		ATTRIBUTES		HTML		
CSS		ATTRIBUTES		HTML		
CSS		ATTRIBUTES		HTML		
CSS		ATTRIBUTES		HTML		
CSS		ATTRIBUTES		HTML		
CSS		ATTRIBUTES		HTML		
CSS		ATTRIBUTES		HTML		
CSS		ATTRIBUTES		HTML		
CSS		ATTRIBUTES		HTML		
CSS		ATTRIBUTES		HTML		
CSS		ATTRIBUTES		HTML		
CSS		ATTRIBUTES		HTML		
CSS		ATTRIBUTES		HTML		
CSS		ATTRIBUTES		HTML		
CSS		ATTRIBUTES		HTML		
CSS		ATTRIBUTES		HTML		
CSS		ATTRIBUTES		HTML		
CSS		ATTRIBUTES		HTML		
CSS		ATTRIBUTES		HTML		
CSS		ATTRIBUTES		HTML		
CSS		ATTRIBUTES		HTML		
CSS		ATTRIBUTES		HTML		
CSS		ATTRIBUTES		HTML		
CSS		ATTRIBUTES		HTML		
CSS		ATTRIBUTES		HTML		
CSS		ATTRIBUTES		HTML		
CSS		ATTRIBUTES		HTML		
CSS		ATTRIBUTES		HTML		
CSS		ATTRIBUTES		HTML		
CSS		ATTRIBUTES		HTML		
CSS		ATTRIBUTES		HTML		
CSS		ATTRIBUTES		HTML		
CSS		ATTRIBUTES		HTML		
CSS		ATTRIBUTES		HTML		
CSS		ATTRIBUTES		HTML		
CSS		ATTRIBUTES		HTML		
CSS		ATTRIBUTES		HTML		
CSS		ATTRIBUTES		HTML		
CSS		ATTRIBUTES		HTML		
CSS		ATTRIBUTES		HTML		
CSS		ATTRIBUTES		HTML		
CSS		ATTRIBUTES		HTML		
CSS		ATTRIBUTES		HTML		
CSS		ATTRIBUTES		HTML		
CSS		ATTRIBUTES		HTML		
CSS		ATTRIBUTES		HTML		
CSS		ATTRIBUTES		HTML		
CSS		ATTRIBUTES		HTML		
CSS		ATTRIBUTES		HTML		
CSS		ATTRIBUTES		HTML		
CSS		ATTRIBUTES		HTML		
CSS		ATTRIBUTES		HTML		
CSS		ATTRIBUTES		HTML		
CSS		ATTRIBUTES		HTML		
CSS		ATTRIBUTES		HTML		
CSS		ATTRIBUTES		HTML		
CSS		ATTRIBUTES		HTML		
CSS		ATTRIBUTES		HTML		
CSS		ATTRIBUTES		HTML		
CSS		ATTRIBUTES		HTML		
CSS		ATTRIBUTES		HTML		
CSS		ATTRIBUTES		HTML		



# AJAX

(Asynchronous  
JavaScript And XML)

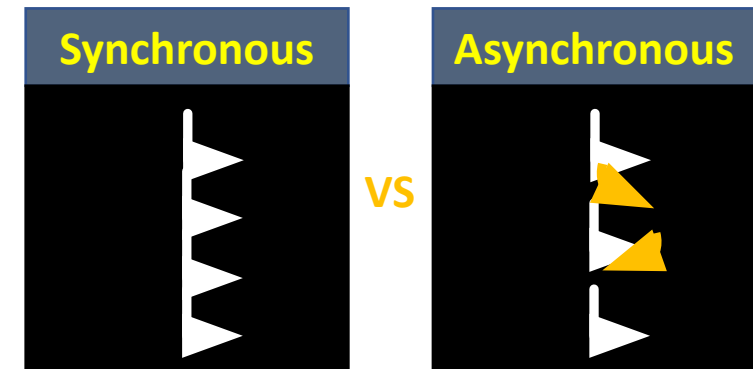
- Synchronous vs Asynchronous
  - Allows work to continue, don't wait for response
  - Uses callbacks
- `$.get()`, `$.post()`
  - Specialized versions of `$.ajax()`



```
$.get('https://randomuser.me/api/', function(data) {  
  console.log(data);  
});
```

Is shortcut for

```
$.ajax({  
  url: 'https://randomuser.me/api/',  
  method: 'GET',  
  dataType: 'json',  
  success: function(data) {  
    console.log(data);  
  }  
});
```

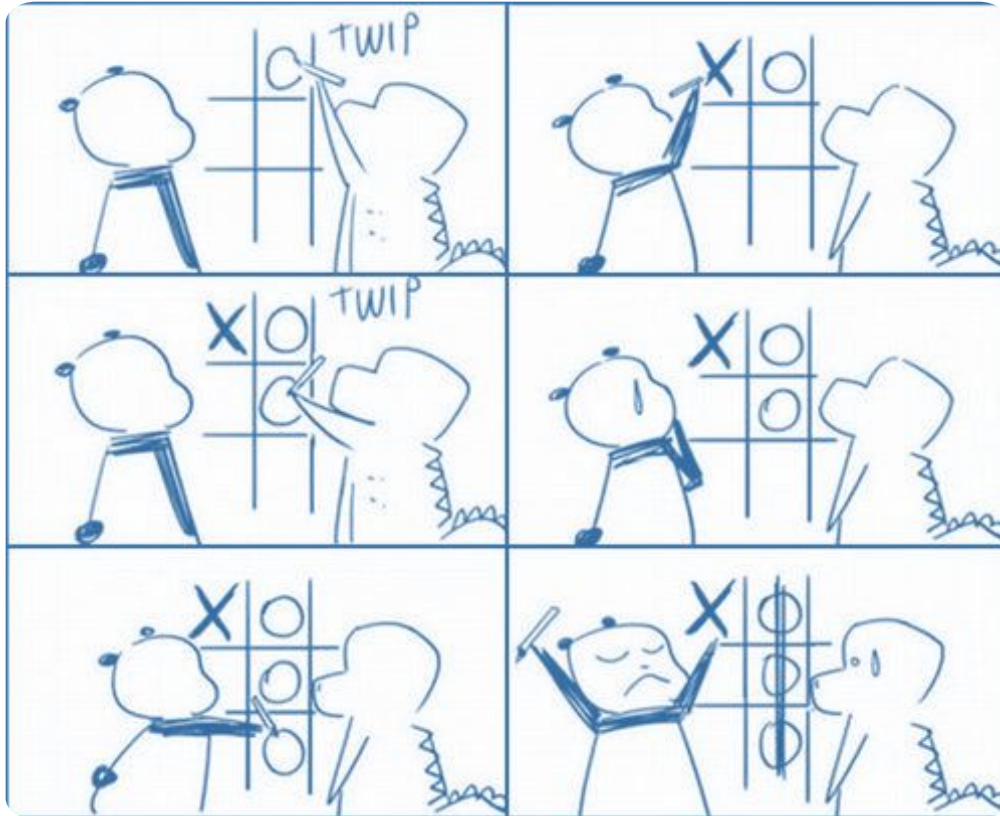






# DEMO

jQuery: AJAX & DOM  
**OH MY!**



## Assignment: (*Tic-Tac-Toe*)

Using any of the tools you've worked with so far, create a game of **tic-tac-toe**.

- A heading should say whether it is X's or O's turn and change with each move made.
- Create a tic-tac-toe grid using your HTML element of choice.
  - When a cell in the grid is clicked, an X or O should appear in that spot depending on whose turn it is.
- A button should be available to clear the grid and restart the game.
- When a player has won, or the board is full and the game results in a draw, a Bootstrap alert or similar Bootstrap component should appear across the screen announcing the winner.