jQuery Cheat Sheet Functionality (use jquery.com for documentation)

Opperators

.typeof
- check what type the file is

.val
- get the div with id=element and set the value to 'Foo'

EX. $(div#element**.val('Foo')**);

first-child
- selection based on first element of parent group $('element**:first-child**')
- afflict elements of first span in parent container.

EX. $('span**:first-child**');

last-child
- works the same as the first child selector

^=
- with use of the ^= we will find an input that starts with a value of "Events" (SQL Like statement / regex opening char)

EX. $('input[value^="Events"]');

$=
- with the use of $= get the input that has a value ending with "Events" (regex closing char)

EX. $('input[value$="Events"]'); //

*** =**
- get all nodes that contain a specified value
- use of the *= value will gather all inputs that contain the specified value.

EX. $('input[value*="Events"]');

:eq(value)
- used to access DOM object at position value. div:eq(3); - searches DOM for the third available DIV

:nth-child(value)
- get the child at the nth position specified by value.

.each
- for iterating through loop values a specified

$(this)
- jQuery Object, this = DOM Object

NOTES:
- jQuery is case sensitive.

Selectors

What is a selector?

A way to select a node (tag, element) from the DOM for control,
A selector can select single or multiple at the same time.

Selector syntax
- $(selectorExpression) or jQuery(selectorExpression)

Selecting nodes bassed on tag Name
javascript - document.getElementById('idValue');
jQuery - $('p') gets all p tags, $('a') selects all anchor tags
Selecting multiple tags
jQuery - $('p, a, span') select all of these tags in the DOM.
Selecting Descendants
$('ancestor descendant') selects all descendants of the anchor
ex. $('table tr') - gets all the table row(tr) references of a table

Selecting Elements by ID

= look through DOM and find the ID specified
$('#IDtoFind')

Selecting Elements by Class Name

. = look through DOM and find all class ID matching
$('.ClassToFind'); - select a single
$('.firstClass, .secondClass'); - select multiples

access a particular value of a class
$('div.ClassToFind') locates only the div values.
$('span.ClassToFind') locates only the span with the class value

Selecting nodes by attribute value

Find a specified node that has a value set to it use the [] within the selector
[] = where. EX. $(jquery)('a(get Anchor)[(where)value='setValue']');
NOTE: single = when evaluating with jQuery.

ex. $('a[valueToFind]'); // This says get the anchor tag with the value attached
$('a[title='testAnchor']); // gets the anchor tag with the title attribute set to 'testAnchor'
$('input[type="hidden"]'); // get all the input boxes where the type is hidden

Selecting input Nodes

: = select input elements on page (form elements, images, textboxes, select, input etc)
var imputs = $(':input'); selects all input tags

$(inputs[1]).val(); gets the value of the selected input box
// get all values from a form simply through looping in jQuery
$(':input').each(// check for each input type object (including images, text, select, etc)
function() { // start the inner (annonymous) function

```
        var element = $(this);  // get the current node being parsed
        // once the element is selected the jQuery library is now available
        element.css('color', 'White');                  // direct access to values
without a jQuery wrapper

});
```

Using Selector specifications

:contains = locate a specified value and select that node.

```
// locate the div that has within its tags a value matching "TEST Value",  Case must
match.
EX. $('div:contains("TEST Value")');
// working with tables ( lmao )
EX. $('tr:odd') - gets the odd table rows ( to adjust color, mouseover, size, alternating )
EX. $('tr:even') - gets the even table rows
```

CSS styles with jQuery

```
// locate all the divs and give them a style of background-color
$('div').css('background-color', 'Green');
// setting a html value to a selected div
$('div').each(function() {
        // use the this key word to access the current value in the each loop of divs
        alert($(this).html());
});
```

Accessing the DOM with jQuery

Simple each loop example:

```
        $('div').each(
                // declairing a function with a (index) value will set it to iteration grabbing.
                function(index) {
                // Use of index can be used now to gather data from value
                        // EX.  if(index = "value") {
                                        // do some task on the item
                                }// End if

                        // create a variable of the value being iterated through.
                        // use the $() with "this" to grab the current value of loop
                        var testable = index + " " + $(this).text();
                }
```

```
        );

Medium each loop example:
var html = '';
// get all divs from the DOM and loop each
$('div').each(
        // start the function to affect each of the values being set up
        function(index, elem(optional)) {
                html += 'Value at ' + index + ": " is $(this).text();
        }// End function
        // append the values to the current element
        $(#elementById).html(html);

);
```

AFFECT ATTRIBUTES

```
var value = $('#mydiv').attr('title');              // get the title value
$('img').attr('title', 'My Image Title'); // set image title

JSON ( java script object notation )
-----------------------------------
A JSON object has a stadard setup.  Attribute, seperator, value (attr:'value')

$('img').attr({
        title:'My Image Title',
        style:'border:2px solid Red;color:Blue; background-color:Green',
        value:'new Image'
});

Object
{ // open JSON object
        FirstName:'Nick',
        LastName:'Bennett',
        Address: {      // nested inner JSON object
                street:'Elgin S',
                streetNum:'402'
        } // End of inner object
}// End of outter object
```

JSON object using css

```
$('div').css ({
        // quoted or unquoted value in css
        'color':'#ccc',
        'font-weight':'bold'                (no trailing comma will cause IE crash)
```

```
});
```

CHAINING

Javascript chaining allows for multiple values to be set through the jQuery object

```
$('div.newDiv').attr (
        {
                // attribute values set here
        }// End the attr
)       // Start chainging the css values to the attr
.css('background-color', 'Red')
.css('color', 'Black')
.css('font-size', '2em');
```

ALTERNATIVE CHAINING

```
$('div.newDiv').attr({
        title:'new Title'
}).css ({
        'color':'white',
        'background-color':'green')
});
```

Adding/Removing Nodes

.append() - add to bottom
.appendto()
.prepend - add to top
.prependto()
.remove() - remove from DOM

```
// get the jQuery object and append(bottom) it to the class (.)phone
        $('<span>(office)</span>').appendTo('.phone');
// get the class (.)phone and append it with the new jQuery Object
        $('.phone').append('<span>(office)</span>);
```

.wrap()
- wrap a child node with a new parent node

```
// get the object with class .state and wrap it with the new div
        $('.state').wrap('<div class="US_State"></div>');
```

.remove()

```
// locate the classes .phone and .location and remove them from the DOM
// removes the parent and also all the children (caution)
        $('div.tester').remove();
        $('.phone, .location').remove();
```

Manipulating CSS classes

.addClass() - checks for existing class, adds if class doesn't exist. adds a space between existing class if does
.hasClass() - check to see if the selected DOM object has a class (boolean)
.removeClass() - check and remove a class from a DOM object
.toggleClass() - turn a class on and off if it exists.

.addClass
```
$('p').addClass('classOne');
$('p').addClass('classOne classTwo');
```

.hasClass
```
if($('p').hasClass('testerClass')) {
        // checking of class and manipulate that class
}// End if
```

.removeClass
```
$('p').removeClass('classOne');                 // find and remove classOne
// remove all classes in a p tag
$('p').removeClass();
```

.toggleClass
```
$('#details').toggleClass('hightlight');            // locate the CSS class and add or remove from #details
// CSS class
<style type="text/css">
 .hightlight { background:yellow; }
</style>
```

** TEST ME OUT **
```
/* add to the html an onfucus='focusBlur(this)' and onblur='focusBlur(this)' event handler
   in the main script add the function. each time the function is called it will execute the toggle command and set the values.
focusBlur(element) {
  $(element).toggleClass('hightlight');
};

*/
```