

CS3226

WEB PROGRAMMING AND APPLICATIONS

LECTURE 05 - JAVASCRIPT (JQUERY)

DR STEVEN HALIM

STEVENHALIM@GMAIL.COM

For 'not newbies': Try this [JS quiz](#) and [jQuery quiz](#) @
W3Schools first

OUTLINE

- Vanilla JavaScript
- jQuery
- JavaScript with jQuery
- DOM
- Drawing on HTML5 Canvas
- AJAX
- Extra Challenges
- JS Quirks

VANILLA JAVASCRIPT A.K.A. ECMASCRIPT*:

1. Is a de-facto standard client-side scripting (and interpreted*) language
2. Has original purpose to enhance the functionality and appearance of webpages (dynamic client-side content)
3. Runs on the web browser (Chrome, Firefox, IE, Safari, Opera, ...)

4. Is not Java

(you will see this very soon, e.g. dynamic typing*)

5. Has **event-driven** model

(we will discuss this after we cover the basics)

6. Has access to **Document Object Model (DOM)**

(we will see some quick examples on the next slide)

SAMPLE (SEQUENCE)

The test line with id="seq"

Example (Vanilla) JavaScript code for demonstration:

```
// 'document' is a very important object in JavaScript
document.getElementById("seq").innerHTML = "Hi"; // this is comment
document.getElementById("seq").innerHTML = "Hi"; // common typo :(
document.getElementById("seq").innerHTML = Date(); // object 'Date'
window.alert("This is annoying"); // 'window' object is optional
var i = prompt("Enter a number:"); // declare + initialize a variable
j = prompt("Enter another number:"); // is 'var' compulsory?
document.getElementById("seq").innerHTML = "i+j = " + (i+j);
document.getElementById("seq").innerHTML = (parseInt(i)+parseInt(j));
console.log(10%8*(3+5)); // % is modulo, notice precedence
console.log(10>7 && 1==1); // relational + logical operator
var k = 7; // try removing the semicolon, it 'works', but 'dangerous'
k = "dynamic typing"; // changing type... it is possible :0
```

JQUERY: WRITE LESS, DO MORE



We will use [jQuery](#) to simplify our web programming life, especially the front-end part (more [later](#))

Disclaimer: [Other JavaScript libraries](#) are not officially supported, please explore them on your own

Note: One of the Lab TA: Erin Teo, is assigned to concentrate on another emerging JavaScript framework: [AngularJS](#)

JQUERY BASICS

To use jQuery, simply add from this Content Delivery Network (CDN)*:

```
<script src="http://code.jquery.com/jquery-2.2.0.min.js"></script>
```

([open source](#)) to your HTML (<head>/end of <body>)

Again, is <script> best placed in <head> element?

Answer

CSS-STYLE SELECTOR

Now, the way we select HTML elements is simpler

The test line with id="seq2"

```
// previously
document.getElementById("seq2").innerHTML = "Hi";
document.getElementById("seq2").innerHTML = "Hi"; // common typo :(
// now :)
$("#seq2").html("<i>Hi</i>"); // CSS selector, shorter, less typo
// you can use single ' or double " quotes
$("#seq2").text("<i>Hi again</i>"); // can be safer
```


SELECTION STATEMENTS

The test line with id="sel"

```
var c = parseInt(prompt("Menu: 1=green, 2=blue, other:change corner"))
if (c == 1)
    // document.getElementById("sel").style.color = "green";
    $("#sel").css("color", "green");
else if (c == 2) { // we can use {} to group more than one statements
    $("#sel").css("color", "blue");
    $("#sel").html("I change this one too");
}
else // the else part is optional
    // in the past, doing this is very troublesome...
    $("#sel").css("borderRadius", "15px");
/* This is an example of a multiline comment:
The if-selection statement in JavaScript is very much similar
to what you are used to in C/C++/Java languages */
```

A SMART GREETING

Not weekend :(

```
<script>
var cur = new Date(); // btw, check the page source for more info
var d = cur.getDay(); // load this page on weekend vs non weekend :)
switch (d) { // now I am using switch -- another selection statement
  case 0: // Sunday
  case 6: // Saturday, let the two cases be treated as weekend
    $("#greeting").html("Ah, weekend");
    break;
  default:
    $("#greeting").html("Not weekend :(");
    break;
}
</script>
```

REPETITION STATEMENTS

The test line with id="rep"

```
var a = [2, 1, 7, 8, 3]; // introducing a simple JavaScript 1D array
var i, sum = 0;
for (i = 0; i < a.length; i++)
    sum += a[i];
$("#rep").html("The sum is " + sum); // 21

i = 0; sum = 0;
while (i < a.length)
    sum += a[i++];
$("#rep").html("The sum is " + sum); // 21

i = 0; sum = 0;
do // btw, what is the corner case that breaks this do-while loop?
    sum += a[i++];
while (i < a.length)
$("#rep").html("The sum is " + sum); // 21
```

A MORE COMPLEX CONTROL FLOW

The test line with id="cf"

```
var a = [2, 1, 7, 8, 3], i, sum;
for (i = sum = 0; i < a.length; i++)
  if (a[i]%2 == 0) // an even number
    continue; // skip this iteration
  else if (sum > 7)
    break; // stop the associated loop
  else
    sum += a[i];
$("#cf").html("The sum is " + sum); // 8
```

```
// some coding geeks (CS3233 :O) usually do this
for (i = sum = 0; i < a.length && sum <= 7; i++)
  sum += a[i]%2 == 0 ? 0 : a[i]; // T_or_F ? if_true : if_false :)
$("#cf").html("The sum is " + sum); // 8
```

JAVASCRIPT FUNCTION

The test line with id="fun"

```
function sum(A) { // Give a 1D JS array 'A' as the input
  var ans = 0;
  for (var i = 0; i < A.length; i++)
    ans += A[i];
  return ans;
}

$("#fun").html("The sum is " + sum([1, 2, 3, 4])); // 10
```

JAVASCRIPT RECURSIVE FUNCTION

The test line with id="fib-sample"

```
function fib(n) { // try http://visualgo.net/recursion.html
  if (n <= 1) return n; // base case
  return fib(n-1) + fib(n-2); // recursive case
}

var output = "";
for (var i = 0; i <= 10; i++) // don't try large numbers
  output += "fib(" + i + ") = " + fib(i) + ", ";
$("#fib-sample").html(output);
```

JAVASCRIPT ARRAYS - 1D

So far we have seen:

```
var A = 7; // variable with single value
var B = [2, 3, 5, 7, 11]; // a simple 1D array
console.log(B[3]); // example access 7
```

Now let's check a few other operations on array, e.g.

```
// used to provide space buffer of size N
var N = 5, D = new Array(N), E = D.length;
var F = new Array(5, 1, 10, 2); // alternative array declaration, FYI
F.sort() // the content is now sorted (the previous content is gone)
F.sort(function(a, b) { return a-b; }) // sort ascending numerically
```

More References

MULTIDIMENSIONAL ARRAYS

We can create **multi-dimensional** (usually 2D) array in JS:

```
var G = [ [1, 2, 3], [4], [5, 6], [7] ]; // row has different length
console.log(G.length + ' ' + G[0].length + ' ' + G[1].length); // 4 3 2
```

We usually use (for) loops to iterate through the 2D array:

```
var total = 0;
for (var i = 0; i < G.length; i++) // iterate row by row
    for (var j = 0; j < G[i].length; j++) // each row has it's own length
        total += G[i][j]; // total should be 28 at the end

// alternative way
total = 0;
for (var row in G) // new syntax: for (var varname in arrayname)
    for (var col in G[row])
        total += G[row][col]; // total should also be 28 at the end
```


JAVASCRIPT (BUILT-IN) OBJECTS

JavaScript has several useful built-in objects: Math, String, Date, Number, document, window, localStorage, etc...

```
Math.sqrt(Math.pow(7, 2)); // sqrt(7^2) = sqrt(49) = 7
var str = "Steven";
str.replace(/e/g, "3"); // replace ALL occurrences of 'e' with '3'
var x = 7, y = 8;
console.log(x.toString(2) + ', ' + y.toString(2)); // dec to bin :)
// most frequent usage if not using jQuery: document.getElementById()
console.log(document.title + ', ' + document.URL);
function curTime() { console.log((new Date).getTime()); }
// milliseconds since 1 Jan 1970
var id = window.setInterval(curTime, 1000);
clearInterval(id); // 'window' is optional
if (localStorage.visitcount) // better than HTTP cookies
    localStorage.visitcount = parseInt(localStorage.visitcount)+1;
else
    localStorage.visitcount = 1;
```

[More References](#)

DEBUGGING JAVASCRIPT PROGRAM

The example inside the `<script>` section of this slide is buggy (right click to view the page source)

We will open Google Chrome Developer Tool (CTRL+SHIFT+I) and use Google Chrome built-in debugging feature to find and fix that bug

DOCUMENT OBJECT MODEL (DOM):

- The specification is defined by the [W3C](#)
- It is a collection of objects making up the JavaScript runtime environment specific to the page the JS is executing in
- These objects give JS access to browser window content, properties, events

We access these with JS class document and so far we have seen the all important `document.getElementById('the_id')`

We will use Google Chrome Developer Tool (the Elements tab) to show the DOM tree of this lecture note

We will access/manipulate DOM with jQuery library as the direct 'VanillaJS' method is too complex

References: [Node at Mozilla](#), [Core DOM Objects](#)

DOM MANIPULATION

JS enables you to alter the DOM of *any* website...

```
var ps = document.querySelectorAll("p"); // select all paragraphs
for (var i = 0; i < ps.length; i++) {
  ps[i].innerHTML = "HACKED?"; ps[i].style.color = "red"; } // change
```

With jQuery:

```
$("p").html("HACKED?") // easier with jQuery, notice the method chain
.css("color", "red"); // and implicit loop of all 'p' elements
```

This is not a <p>, so it will not change. Press F5 (Refresh) before continuing if you try the code above and things will go back to normal again as this 'hack' is only on client-side...

DOM MANIPULATION WITH JQUERY

Below, the `<div>` element with id "test" contains an initially empty anchor `<a>` with id "anch" and `<button>` with id "disappear" and will be manipulated via jQuery calls using in the script on the next slide

empty

```
/* CSS for this example */
#test { border: 2px red; border-style: ridge; padding: 5px; }
/* notice dynamic addition and removal of this CSS style */
.redstuff { background: red; border-radius: 25px; }
```

```
// JavaScript for this example
$(function() { // or $(document).ready();
    // this is a very important concept: Wait until the DOM is ready
    $("#anch").html("http://visualgo.net");
    $("#anch").attr("href", "http://visualgo.net"); // new stuff: "att
    $("#anch").on("mouseover", function() { // register event
        $("#test").addClass("redstuff");
    });
    $("#anch").on("mouseout", function() { // register another event
        $("#test").removeClass("redstuff");
    });
    $("#disappear").html("Click me, then I will go away... slowly");
    $("#disappear").on("click", function() {
        $("#disappear").fadeOut(1000); // effect: disappear in 1s (1000ms
    });
});
```

JAVASCRIPT EVENT HANDLING

JS can be used to intercept and handle events that are happening on a HTML5 page

Just now, we have seen a button-click example

Do something with my onclick event (id="demo")

Recap: An easy way to attach a JavaScript code/function to handle events is with jQuery:

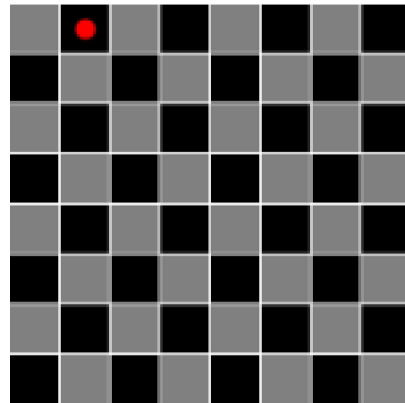
```
$("#demo").on("click", function(event) {  
    alert("Yey, you really listening to my onclick event :)");  
    console.log(event); // open Console and see what you can do with 'e'  
});
```

[More References](#)

DRAWING ON HTML5 CANVAS

Basically, HTML5 canvas is just an empty space for us to draw with JavaScript (web browser does not relay layout content when things are drawn in canvas)

Those who have seen/use OpenGL (Computer Graphics) will realize the similarities



[More References](#)

```
var w = 200, h = 200;
// memorize this... I prefer to make the variable ctx global
var ctx = document.getElementById("samplecanvas").getContext("2d");

function DrawChessBoard(n) { // let's make an n*n chessboard
  var scale = w/n;
  for (var i = 0; i < n; i++) {
    var c = i%2;
    for (var j = 0; j < n; j++) {
      c = 1-c; // flip
      ctx.fillStyle = (c == 0 ? "black" : "grey");
      // a square, leave 1 pixel spacing so that you can see the gaps
      ctx.fillRect(j*scale, i*scale, scale-1, scale-1);
    }
  }
}
```

```
// put circle at cell (r,c), 1-based indexing, of an n*n chessboard
function DrawCircle(r, c, n) {
    var scale = w/n;
    // the center of cell (r, c) is at coordinate (cx, cy)
    var cx = (c-0.5)*scale; cy = (r-0.5)*scale;

    ctx.strokeStyle = "red"; // color
    ctx.lineWidth = "5"; // thicker line
    ctx.beginPath();
    // Outer circle (anticlockwise)
    ctx.arc(cx, cy, scale/2-10, 0, Math.PI*2, true);
    // Use this one for Pacman style
    // ctx.arc(cx, cy, scale/2-10, -Math.PI/4, Math.PI/4, true);
    ctx.stroke();
}
```

```
DrawChessBoard(8); // standard chessboard
DrawCircle(1, 2, 8);

var cur = 1;
var anim;
function Animate() { // try calling this
    DrawChessBoard(8); // redraw
    DrawCircle(1, cur++, 8);
    if (cur == 9) {
        clearInterval(anim); // stop the animation
        cur = 1;
    }
}
// uncomment this to trigger an animation
// anim = window.setInterval(Animate, 1000);
}
```

ASYNCHRONOUS JAVASCRIPT AND XML (AJAX)

What is AJAX?

Answer

AJAX EXAMPLE

```
function revealans(question_id) {
  if (!localStorage.getItem("weakpwd"))
    localStorage.setItem("weakpwd", prompt("What is the password?"));
  $.ajax({ // jQuery again, a bit complex if not done using jQuery...
    url: "answer.php?q=" + question_id + "&p=" +
      localStorage.getItem("weakpwd") // we will learn PHP soon
  }).done(function(data) {
    if (data == "-1") {
      alert("Not authenticated, find out what is the password!");
      localStorage.removeItem("weakpwd"); // ask for password again
      return; // don't do anything
    }
    $("#"+question_id).html(data);
    $("#"+question_id+"_button").hide(1000);
  });
}
```

JAVASCRIPT OBJECT NOTATION (JSON)

What is JSON?

Answer

JSON EXAMPLE

For example, this relatively easy-to-understand (and formatted) JSON describes a small undirected unweighted graph in [VisuAlgo MST visualization](#) (CS2010)

```
{
  "v1":{
    "0":{"cx":180,"cy":200},
    "1":{"cx":340,"cy":140},
    "2":{"cx":320,"cy":280}
  },
  "e1":{
    "0":{"vertexA":0,"vertexB":1,"weight":2},
    "1":{"vertexA":1,"vertexB":2,"weight":3},
    "2":{"vertexB":0,"vertexA":2,"weight":4}
  }
}
```

Click this [long URL](#) with that long query string

Now open Google Chrome Console and type

```
// var G = JSON.parse(JSON_string) method is called by default  
var G = {"v1":{"0":{"cx":180,"cy":200},"1":{"cx":340,"cy":140},"2":{"cx":500,"cy":100}}
```

Then you can access the values like this:

```
G.v1[0].cx // should be 180  
G.el[1].weight // should be 3
```

To convert JavaScript object into JSON, we use:

```
var JSON_string = JSON.stringify(obj)
```

Instead of XMLs, now we usually pass JSONs between client-side JavaScript and server-side **PHP** via AJAX*

EXTRA CHALLENGES

#1 MULTIPLICATION TABLE

Create a simple JavaScript code to create a multiplication table of size 10×10 that shows the multiplications of two integers $a \times b$ where $a, b \in [1..10]$, then highlight all table cells that contain square numbers

What is the solution?

Answer

#2 MINI* VISUALGO ONLINE QUIZ

Create a simple JavaScript code/function so that you create a random but simple Mathematics question:

Given $A^* = 0$ and $B^* = 0$, what is $C^* = A+B$?

☐ 0 ☐ 1 ☐ 2 ☐ 3 ☐ 4

Status bar...

What is the solution?

Answer

#3 SORTING TABLES

Examine [CS3233](#) ranklist table

Given a table with several rows and columns (the content of the table will be frequently updated), how to use JavaScript to automatically sort the rows based on column final sum?

What is the solution?

Answer

JS QUIRKS (1)

I will close JavaScript lecture by mentioning a few JavaScript quirks: "some characteristics/behaviors of JavaScript that may surprise programmers who are used to other programming languages"

```
7 + 8 // is 15  
'7' + '8' // is '78', but what about  
'7' * '8' // ?  
'7' - '8' // ?
```

```
0 == false // oh yeah  
1 == true  // yeah  
2 == true  // check this
```

```
var B = [0, 1, 2, 3];  
B[7] = 1; // error: array index out of bound? nope, it resizes itself
```

JS QUIRKS (2)

```
function sum() {  
  console.log(arguments.length);  
}
```

```
sum(); // what will you see?  
sum(7); // what will you see?  
sum(7, 2); // what will you see?
```

```
[] + [] // for fun: try this 4 combinations  
[] + {} // what do you think you will see?  
{ } + []  
{ } + { }
```

There are a few more quirks like this
(Google around, some of them are very technical)...

THE END

If you haven't do it,
try this [JS quiz](#) and [jQuery quiz](#) @ W3Schools for self check

FAQ

1. Why teach jQuery instead of VanillaJS?
2. Why teach jQuery instead of AngularJS?
3. Why jQueryUI is not taught?
4. Why certain part of JS/jQuery is not taught?
5. Why ??JS is not taught?