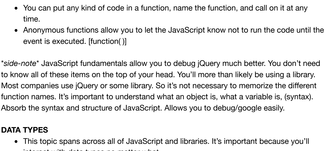
EVENTS

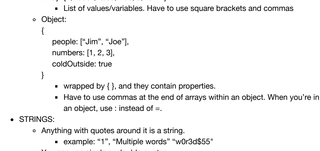
- JavaScript is driven by events - it lets JavaScript know when to do things, and what to do.
- Not that many methods of input for web data (keyboard and mouse clicks). This is driven through events.
- Most useful events:
  - onclick - when you click on an object
  - onload - when the page loads - wrap all your jQuery in an “onload” so it waits until the page loads before running any code.
  - onfocus - glowing border look - lets you know what is in focus
  - onblur - when the item is out of focus
  - onscroll - use any time you scroll. Good for trigger hotpoint.
  - onsubmit - for a form.
- An Example Link
  - This is old school, don't use the example above - you don't want your JavaScript in your HTML
- The attributes are set in JavaScript now. We tell it to select the HTML element, and then list the event.
- <script type="text/javascript">
  document.getElementById("myLink").onclick = function ( ){alert( );};
  </script>
- Anytime you use { } in JavaScript, you usually put a function ( ) before them.
- You can put functions in variables
- Try it out in JSbin
- .innerHTML - to be able to change HTML content using JavaScript

- JavaScript is very specific - you have to have capitalization and spaces match the HTML reference.
- JavaScript does camel case: example:
  - CSS: margin-left
  - JavaScript: marginLeft
- You can use .querySelector to be able to select ID’s or classes in HTML:

- If you change multiple elements, you don’t want to copy “document.querySelector” multiple times. Here’s the easy solution to that:
  - Create a variable that equals “document.querySelector(‘.myLink’)
  - Reuse that variable
  - Make sure the JavaScript is calling an HTML class with .myLink (IDs in HTML are #myLink).



- Alerts have to go within functions for onclick, otherwise it will run without clicking the link:

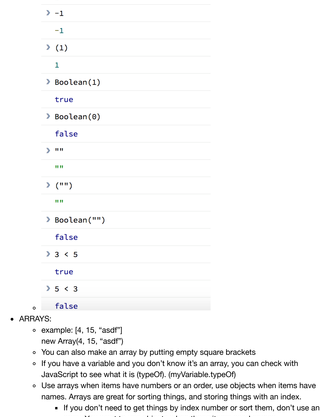


- You can put any kind of code in a function, name the function, and call on it at any time.
- Anonymous functions allow you to let the JavaScript know not to run the code until the event is executed. [function( )]

“side-note” JavaScript fundamentals allow you to debug jQuery much better. You don’t need to know all of these items on the top of your head. You’ll more than likely be using a library. Most companies use jQuery or some library. So it’s not necessary to memorize the different function names. It’s important to understand what an object is, what a variable is, (syntax). Absorb the syntax and structure of JavaScript. Allows you to debug/google easily.

DATA TYPES

- This topic spans across all of JavaScript and libraries. It’s important because you’ll interact with data types no matter what.
- Here are the different data types:
  - String: “Joe”
    - Anytime you have text, it has to be a string. Has to be wrapped in quotes.
  - Number: 5
    - Act like we expect them to, no quotes.
  - Boolean: true
    - True or false
  - Array: [1, “Joe”, false, true, 3, “Jim”]
    - List of values/variables. Have to use square brackets and commas
  - Object:
    {
    people: [“Jim”, “Joe”],
    numbers: [1, 2, 3],
    coldOutside: true
    }
    - wrapped by { }, and they contain properties.
    - Have to use commas at the end of arrays within an object. When you’re in an object, use : instead of =.
- STRINGS:
  - Anything with quotes around it is a string.
    - example: “1”, “Multiple words” “w0rdd$55”
  - You can use single or double quotes.
    - “One” ‘Another’
  - Quotes and other special characters must be escaped
    - “To quote Steve Jobs: \”Quote\””
    - This is because if you use quotes, it’s going to close the previous string.
    - To escape a character, put a \. You’re telling it “this is not part of programming, it’s part of the string.
      - If you want to add a \, do a \\.
    - If your strings break around symbols, you probably need to escape it (put a \)
    - Another way to mitigate this is combining single and double quotes.
  - Anything combined with a string, becomes a string.
    - “1” + 358 will result in “1358”
  - JavaScript has lots of ways to modify strings.
    - “Google rocks”.replace(‘Google’, ‘Apple’) = “Apple rocks.”
    - JavaScript can format text (string manipulation)
      - example: don’t ask for a specific format for a date (1/1/2016 vs 1-1-16) - ways to manipulate this in your code without frustrating your users
- NUMBER/INTEGERS:
  - Infinity: Infinity
    - This is a keyword - it’s the actual value Infinity.
    - Infinity is a reserved word. You cannot make a variable called Infinity since it’s been reserved by JavaScript. Other reserved words are **var** and **function**.
  - Hex: 0xFF
  - Float or Floating Point: 1.23
    - The . can float back and forth. It is only precise up to 15 digits.
    - Anytime you combine a number with a float, it becomes a float.
  - NaN: NaN
    - Not a Number (NaN). You’ll see this in errors all the time. Trying to put something that’s not a number where a number will go.
  - Precise up to 15 digits: 999999999999999
  - String * 1 = number: “301” * 1
    - If you have a string and you want to use it as a number, do it * 1 to convert it to a number
    - This is good for forms. When a user inputs something that should be a number, it’s entered as a string. When you do * 1 - it converts it to a number.
  - Number + string = string: 41 + “”
- BOOLEANS:
  - Used a lot for comparisons
  - Boolean (1 + 1) or (1 + 1) or 1 + 1
    - Are True:
      - 1
      - -1
      - true
      - “false”
      - 100
      - 3.14
      - 3 < 112
    - Are False:
      - 0
      - -0
      - false
      - “”
      - undefined
      - null
      - NaN



- ARRAYS:
  - example: [4, 15, “asdf”]
    new Array(4, 15, “asdf”)
  - You can also make an array by putting empty square brackets
  - If you have a variable and you don’t know it’s an array, you can check with JavaScript to see what it is (typeOf). (myVariable.typeOf)
  - Use arrays when items have numbers or an order, use objects when items have names. Arrays are great for sorting things, and storing things with an index.
    - If you don’t need to get things by index number or sort them, don’t use an array. You want to use objects when those items need names.
  - Sort. You can sort arrays. There’s a function called sort.
  - Length. You can find out how many items are in the array. You can put things in arrays to figure out how many items there are.
  - Join. You can bring all of the items in an array to form a string.
- VARIABLES:
  - JS variables have to be identified with unique names, or else they’ll be over-written.
  - Identifiers name variables
  - They can contain letters, numbers, _, $
    - But names must always start with a letter
  - Names are case sensitive.
  - Reserved words (JavaScript keywords) cannot be used as names.
  - Declaring
    - Combing Declarations
    - Redeclaring - you can redeclare variables and it won’t erase it
  - Data types - every single variable has a data type based on its value.
  - Lifetime - where you declare a variable determines it’s lifetime.

Worked on sixteen project for about an hour.

UPCOMING PROJECTS

- No assignments yet, but coming up:
  - Creating a basic to-do app with JavaScript
  - Adding basic JavaScript to sixteen project