

# COMS/SE 319: Software Construction and User Interface Spring 2019

## LAB Activity 2 – JavaScript

### Task 1: PLAY WITH JAVASCRIPT (GETTING STARTED)

#### Assumptions:

- You already know HTML basics.
- If not – then browse through quickly  
<https://www.w3schools.com/html/default.asp>

HTML is mostly about formatting --- so should be easy to grasp.

#### Learning Objectives:

- **Students will:**
  - learn how to embed js in HTML files
  - learn how to debug js code on browser
  - learn about variable types

#### Resource:

All the links shown in the snapshot below have a wealth of information. Please read first.

About JS in general: <https://www.w3schools.com/js/default.asp>

About JS variables: [https://www.w3schools.com/js/js\\_variables.asp](https://www.w3schools.com/js/js_variables.asp)

#### Step 1:

READ [https://www.w3schools.com/js/js\\_where.asp](https://www.w3schools.com/js/js_where.asp)

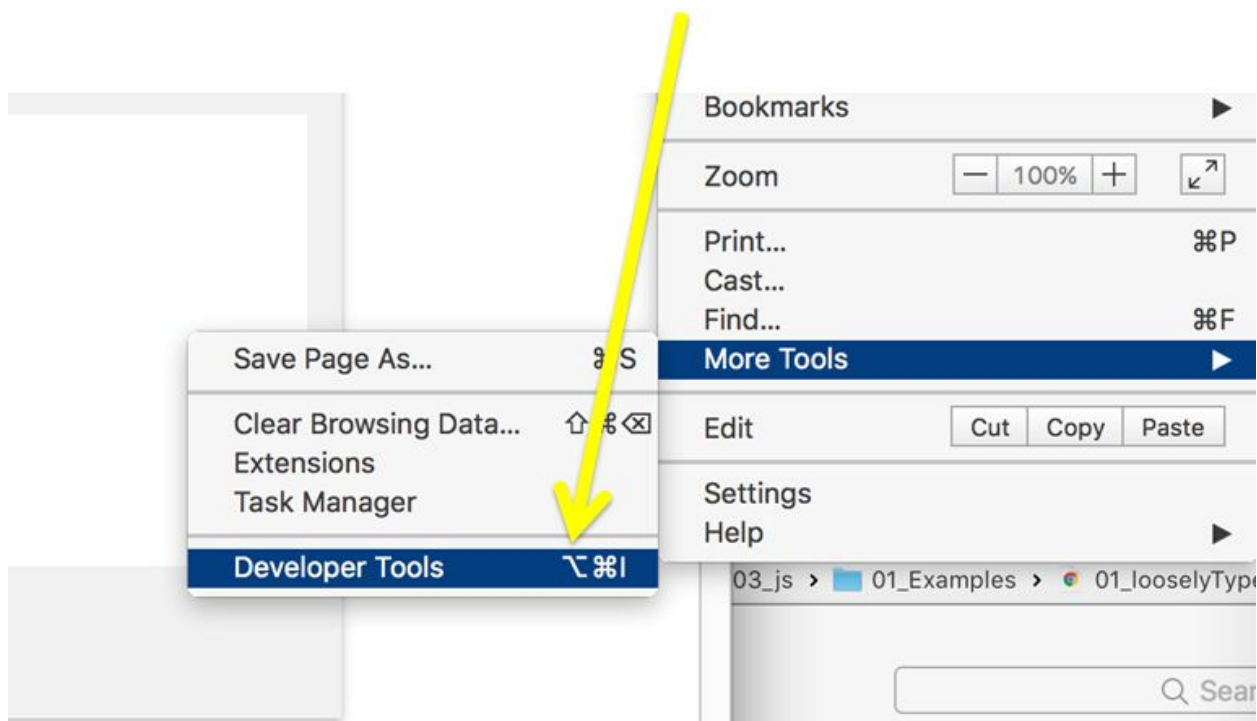
### **Step 2:**

- READ 01\_looselyTyped.html (provided inside zip folder)
- Double click on it (should run the file in the browser).
- We will assume you are using Google Chrome.

NOTE: Since the browser can interpret js (i.e. js runs on the client side), we do not need to have a server to play with js.

### **Step 3:**

Start the debugger. Click on settings, more tools, and the developer tools (or use the shortcut)



### **Step 4:**

Play with the different tabs of the debugger. In particular (Elements, Console, Sources, and Network).

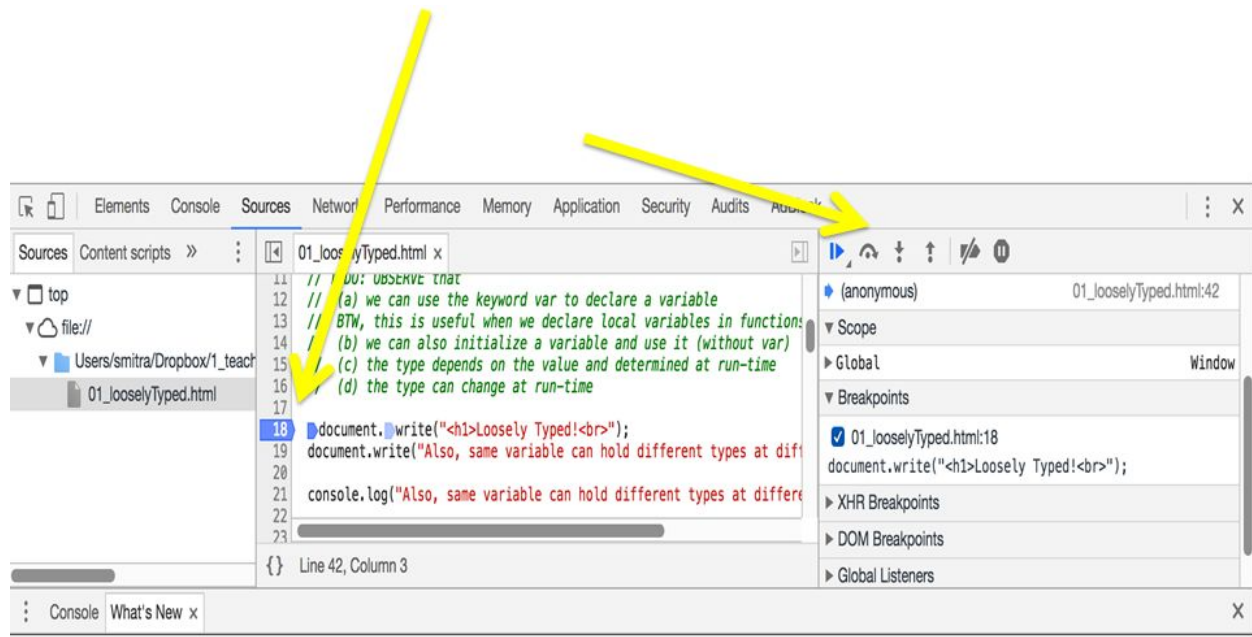
### **Step 5:**

- Run the 01\_looselyTyped.html (provided) by double-clicking on it.
- Think about the difference between document.write and console.log.
- Think about the different types of data that are demonstrated in this example.
- Think about how to find the type of data item.

### **Step 6:**

Run the code in debug mode by

1. inserting a breakpoint (click on line number)
2. refreshing the browser (to reload the program)
3. stepping over a statement.



## **Task 2: PLAY WITH JAVASCRIPT VARIABLES**

### **Learning Objectives:**

- **Students will:**
  - learn about var, let, and const

### **Resource:**

All the links shown in the snapshot below have a wealth of information. Please read first.

About JS variables: [https://www.w3schools.com/js/js\\_variables.asp](https://www.w3schools.com/js/js_variables.asp)

About var, let, and const: <http://wesbos.com/javascript-scoping/>

More about let and const: <http://wesbos.com/let-vs-const/>

### **Step 1:**

- Read about var, let, and const: <http://wesbos.com/javascript-scoping/>
- Read more about let and const: <http://wesbos.com/let-vs-const/>

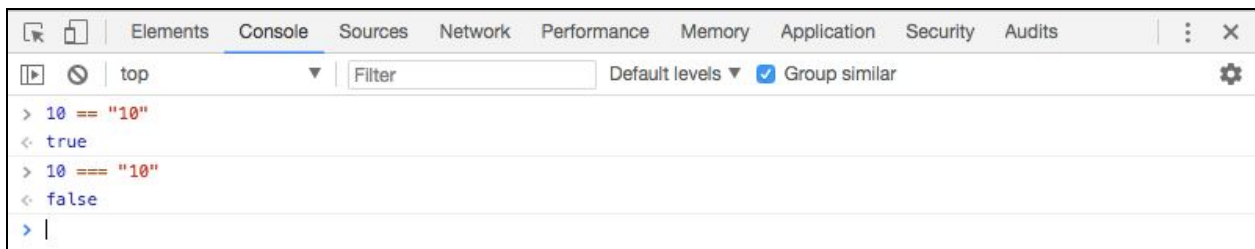
### Step 2:

- READ 02\_varsDeclarations.html (provided)
- Double click on it (should run the file in the browser).
- What do the following scopes mean?
  - Global
  - Function
  - Block
- How do var, let, and const differ from each other?

### Step 3:

- READ 03\_equality.html(provided)
- Double click on it (should run the file in the browser).
- **What is the difference between == and ===? -Think about it and please answer the quiz question in Canvas.**

Note that you can type in JavaScript in the console window and see results!



The screenshot shows a web browser's developer console with the 'Console' tab selected. The console displays the following JavaScript code and its results:

```
> 10 == "10"
< true
> 10 === "10"
< false
> |
```

The console interface includes a toolbar at the top with icons for opening, closing, and refreshing the console, as well as a search bar and a 'Filter' dropdown. The 'Default levels' dropdown is set to 'Default levels', and the 'Group similar' checkbox is checked.

## Task 3: Play with Javascript FUNCTIONS

### Learning Objectives

- Students will:
  1. learn about js functions:
    - function declarations vs function expressions
    - declaration hoisting
    - function names as pointers to function objects

**Resource:**

All the links shown in the snapshot below have a wealth of information. Please read first.

About JS functions: [https://www.w3schools.com/js/js\\_functions.asp](https://www.w3schools.com/js/js_functions.asp)

Also, function scope: [https://www.w3schools.com/js/js\\_scope.asp](https://www.w3schools.com/js/js_scope.asp)

**Step 1:**

1. Read 04\_funcs\_declaration.html
2. Run it by double clicking on the file
3. Think about the two ways that functions can be declared.
4. Think about the differences between the two in how they behave.
5. Think about what happens on lines 36 and 37.

**Step 2:**

1. READ 05\_funcs\_hoisting.html (provided)
2. Double click on it (should run the file in the browser).
3. Think about what function declaration hoisting means.

**Step 3:**

1. READ 06\_funcs\_pointers.html (provided)
2. Double click on it (should run the file in the browser).
3. Explain what happens in line 34?
4. Explain what happens on line 48 and why that is different from line 34.

**Task 4: Functions as First-Class Objects****Learning Objectives:**

- learn what is meant by first class objects
- learn that functions are first-class objects in javascript.

**Resource:**

[https://en.wikipedia.org/wiki/First-class\\_function](https://en.wikipedia.org/wiki/First-class_function)

Useful Blog:

<https://hackernoon.com/effective-functional-javascript-first-class-and-higher-order-functions-713fde8df50a>

**Step 1:**

READ [https://en.wikipedia.org/wiki/First-class\\_function](https://en.wikipedia.org/wiki/First-class_function)

When are functions said to be first class objects?

**Step 2:**

READ 07\_funcs\_firstClass.html (provided)

On which line is

- a) a function being assigned to a variable?
- b) a function being passed as a parameter? [Please answer it in Canvas Quiz]
- c) a function being declared inside a function?
- d) a function being returned from a function?

**Step 3:**

Double click on 07\_funcs\_firstClass.html . Make sure you understand the results!

**Task 5: CLOSURES**

**Learning Objectives:**

- Students will:

Play with functions in order to see examples of closures.

### **Step 1:**

- READ

[https://www.w3schools.com/js/js\\_function\\_closures.asp](https://www.w3schools.com/js/js_function_closures.asp)

and

<http://javascriptissexy.com/understand-javascript-closures-with-ease/>

- What is a javascript closure?

### **Step 2:**

- READ 08\_funcs\_closures.html (provided)
- READ 09\_1\_funcs\_useOfClosures.html
- READ 09\_2\_funcs\_useOfClosures.html

### **Step 3:**

- READ 10\_0\_funcs\_closures3.html
- READ 10\_1\_funcs\_closures3.html
- READ 10\_2\_funcs\_closures3.html

**If you uncomment these lines below in 10\_2\_funcs\_closures3.html file, what is the output?**

**[Please answer it in Canvas Quiz]**

```
//document.write(x[0]() + "<br>");  
//document.write(x[1]() + "<br>");
```

### **Step 4:**

- READ 11\_funcs\_closuresReuseCode.html
- Note how we are able to use "higher order functions" (functions that take other functions as parameters) to abstract out common functionality. Makes the codes much easier to write (and read).

## **Task 6: THIS**

### **Learning Objective:**

- Students will:
  - learn about simple js objects.
  - understand "this" usage in js

### **Step 1:**

- READ 12\_1\_funcs\_objectsThis.html
- Run the code (by double-clicking the file) and then explain what happens.

### **Step 2:**

- READ 13\_objectsAsArrays.html
- Run the code (by double-clicking the file) and then explain what happens.

### **Step 3:**

- Read the SampleProgram (match.js and lab.html).
- Run the code (by double-clicking the HTML) and then explain what happens.

### **Step 4:**

- NOTE that there are some other js files in the folder. You can play with them. As they include some concepts/tools we haven't gone over --- you can also ignore them for now.

**Please submit only the Lab Activity Quiz questions in Canvas. No need to submit other questions mentioned in this pdf.**

---