Wentworth Institute of Technology
COMP4650 – Web Development
Summer 2020, Prof. Jones Yu

Lab 7 - Assignment 2

Due: 11:59 pm, Sunday, July 19

(This is an individual assignment.)

In this lab, we want to help you practice on HTML, CSS and JavaScript. Please download the source files from the Blackboard for this lab.

1. EXERCISE

The tree structure of HTML document is called DOM tree. The Document Object Model (DOM) is a programming interface (API) for JavaScript to interact with HTML document. It turns static HTML documents into an interactive web application. Although we have introduced Developer Tools (such as Developer Tools in Chrome browser), there is a limitation to practice a full JavaScript program. In this problem, we want you to create an environment to write multi-line JavaScript codes inside the browser and apply it to current viewing page.

INSTRUCTIONS:

Step 1: Install Firebug Lite extension to your browser (such as Chrome, Firefox).
https://goo.gl/gLJZF4

Step 2: Go to any webpage and click on  icon at top right corner

Step 3: Inside "Console" tag, click on  icon at bottom right corner

Step 4: Type in your JavaScript codes inside the JavaScript editor and click on "Run" button (Figure 1)

Now, your environment is ready to apply written JavaScript program to viewing web page in the browser.

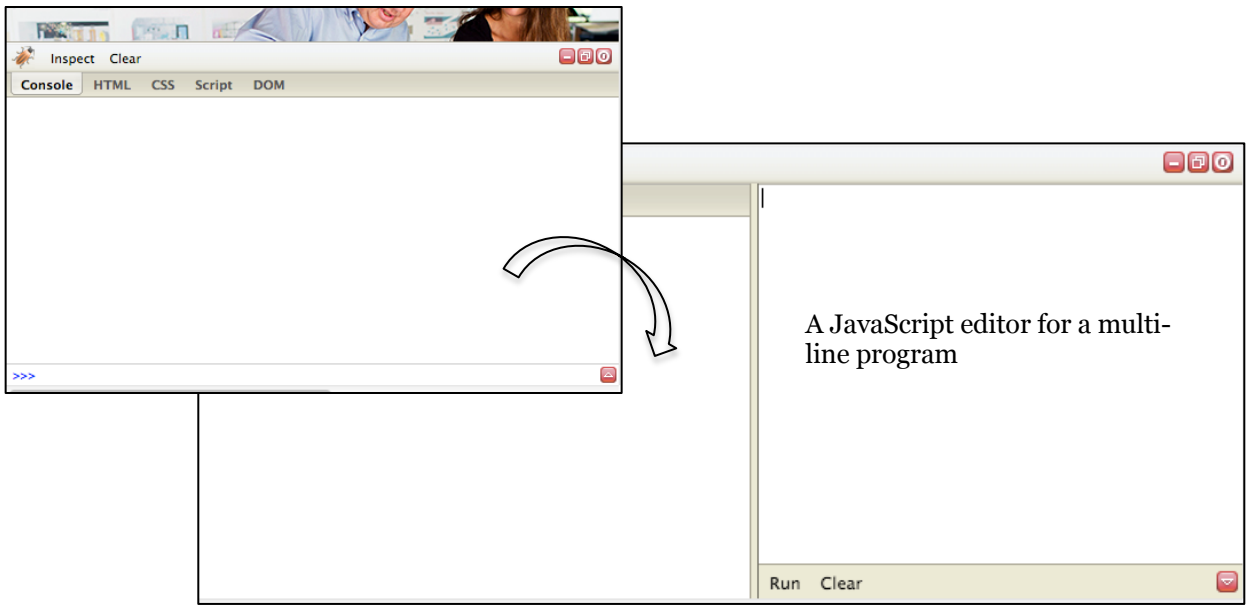

Figure 1: Using the Firebug Lite inside the browser

TASKS FOR EXERCISE:

Task 1: Go to "http://www.nytimes.com" and count the number of <p> nodes, nodes and <img> nodes do you have.

Task 2: Output URL information (src attributes) of all <img> nodes into console tab

Task 3: Output HTML content of all <p> nodes

Task 4: Output plain text content of all <p> nodes

Task 5: Tokenize text content of all <p> nodes and create a dictionary of all tokens with the count of their appearance

2. PROBLEM 1

You will create JavaScript prevalidation for the form listed below. Following are the suggestions of Instructions and Test procedure. (The source files are on the Blackboard.)

INSTRUCTIONS:

1. Link to an external JavaScript file in the head of the page so that you can write code in its own file
2. Define a CSS style to use when highlighting a blank field
3. Set up a listener on the form's submit event so that the code prevents submission of the form (`preventDefault()`) if either the title or description field is left blank or the accept license box is not checked
4. Enhance the JavaScript so that blank fields trigger a change in the appearance of the form

TEST:

1. Test the form in the browser. Try submitting the form with either field blank. You should see the fields highlighted. (Figure 2)
2. Your program needs to allow the user to correct the errors and re-submit the form. For any re-submission, your program should validate the form again.

Figure 2: The expected result of problem 1

3. Submission Requirement of Assignment 3:

Your Assignment 2 only needs to submit the answer of Problem 1. Each student needs to have his/her own submission and submits it to the Blackboard. The submission should be a .zip file that contains following resources:

1. A document (either .doc or .pdf format) that describes:

   - Problem description: a short description of the problem you are solving
   - Analysis:
     - What parts of HTML and JavaScript do you change to solve the problem?
     - What are the difficulties you encounter?
     - …
   - Source code: copy & paste your source code to the report (for this assignment, it means HTML and JavaScript codes)
   - Screenshots of sample runs: show screenshots of completed HTML

2. The folder that contains all the source codes, including HTML/CSS/JavaScript files, images, etc.