

Subject Code: CS3 Computer Science 3 **Module Code:** 8.0 JavaScript Overview

Lesson Code: 8.1 Syntax (script elements, JavaScript rules)

Time Frame: 30 mins.



Time Allocation: 1 min.

After completing this module, you are expected to:

- Know the keywords in JavaScript
- Incorporating/embedding JavaScript inside the HTML documents



Time Allocation: 2 mins.

Based on your learnings in the past quarters, HTML wasn't initially supposed to control the display of web pages in a browser. Web pages were static when HTML was first created, which means they could not change or alter after being rendered by the browser. However, people started to understand that greater interactivity and improved graphic design would make the web more usable as the web expanded beyond a small academic and scientific community. There are more intuitively and outwardly engaging websites that are emerging as commercial applications of the internet developed. HTML on its own, are used to deliver inactive web pages. An inactive website made up of HTML will be considered to be roughly identical to a printed book; you will be able to examine it or move around in it but the content is fixed.

Responding to the desire for greater web page interactivity, a new web programming language is needed. JavaScript fills this need.

In this module, you will learn how to use JavaScript, basic syntax and the rules of JavaScript.



Time Allocation: 17 mins.

What is JavaScript?

JavaScript is a programming language that brings interactivity to our sites and custom behaviors. It is a client-side scripting language which means that like other programming languages do, it runs on the user's computer and not on the server. This also means that it is based on the settings and capabilities of the browser. It may not even be available because a user has chosen to turn it off or because a device doesn't support it, which good developers should keep in mind and plan for. In addition, it is also referred to as a dynamic and loosely typed programming language.

LG-MACALALAD-CS3-8.1 CS 3 Page **1** of **9**



It is a dynamic language because of what you see on paper when you review the code is not the outcome that you will get when you run the code in the browser. This is because JavaScript uses a data structure that is loosely typed, which means you can use all of the declarations (var, const, let) without defining the type of variable.

Data in JavaScript is typed loosely, meaning we do not automatically tell JavaScript what a variable is. We don't have to programmatically define that variable as a number if we set a variable to a value of 7, then JavaScript automatically recognizes it as a number.

Why is it called JavaScript?

"LiveScript" was the initial name of JavaScript when it was first developed. The name was changed to JavaScript to enable its developer to position this new "unused language" as the "younger brother" of Java, when at that time it was the more popular language of the two.

The language was then submitted for standardization to the ECMA world organization. By that point, Netscape did not allow the utilization of the "JavaScript" name, therefore the standardized language is known as ECMAScript.

JavaScript is a language that every big browser on the market currently supports (including Firefox, Chrome, Internet Explorer, Safari and Opera) and many web browsers as well. Initially developed by Netscape.

A standard version of the core JavaScript language is specified by the <u>ECMA-262</u> <u>Specification</u>

Below is example of the ECMA standard derived from the ECMA-262 Specification:

- JavaScript is a lightweight programming language that is interpreted.
- Open and cross-platform.

Client-side JavaScript

Client-side is the language's most common format. The script could be included in or referenced by an HTML document in the code to be translated by the browser.

It implies that HTML can incorporate client-interacting programs, control the browser, and make the content of the HTML powerful.

Advantages

- Less contact with the server: Before sending the page off to the server, you can validate user feedback to save the server action, which indicates less stacking on your server.
- Immediate feedback to the visitors: They do not get to be held up for a page reload to see within the occasion that they have neglected to enter something.
- Expanded interactivity: When a user hovers over an HTML element with a mouse or triggers it through a keyboard, you will be able to create web page interfaces that respond.

LG-MACALALAD-CS3-8.1 CS 3 Page **2** of **9**



• Greater interfacing: To give your web pages a richer GUI, you will be able to use JavaScript to integrate such items as drag-and-drop components and sliders.

Limitations

JavaScript is not possible to treat as a complete programming language because of its limitations:

- JavaScript has no capability to peruse or compose files.
- JavaScript has no support available for organizing applications
- It runs in a single execution thread.

How to use JavaScript in your web page

Using the <script> element

JavaScript codes run from inside a web page and type directly as a separate part of the web page file. The JavaScript programs found in a web page is called a *script*. Once the web browser encounters a script element, it interprets the statements that the element contains in its scripting engine.

Script can be embedded or placed in an external file right into a document and connected to a page like CSS. The <script> element uses both methods.

Embedded <script>

The code must be added as the content of a <script> element to embed a script on a page.

```
<script>
    Your JavaScript code goes here...
</script>
```

External <script>

In the external method, the src attribute is used to point a URL to a script file (with a .js suffix). The <script> element has no content in a case like this.

```
<script src="external_script.js"></script>
```

The benefits of using external style sheets is the application of the same script to different websites. Each external script, on the other hand, requires an additional server HTTP request, which shows down performance.



Note: Short JavaScript could also be included inside an HTML element, example:

Click Me
This would display a Hello World inside a dialog box when a user clicks on Click Me.

LG-MACALALAD-CS3-8.1 CS 3 Page **3** of **9**



Where to place the <script> element?

Typically, the <script> element is placed in the document header and/or at the very end of the body, and goes anywhere in the document. Sprinkling the script through the document is not recommended, as it would be hard to locate and maintain.

The end of the document, just before the </body> tag, is the recommended placement for most scripts, since the document and its DOM structure will be parsed by the browser. The information would then be ready and usable by the time it gets to the scripts and they can execute faster. In addition, moving the script to the bottom improves the perceived performance, simply because the download and execution of the script blocks the page's rendering.

Your First JavaScript code

Let us take a look at our first JavaScript code. The script is placed in the body section of the document and it produces an output that displays a simple text message and a variable value.

The following result will be provided by the sample code above –





Note: **document.write()** method is use to write the content to the current document only while that document is being parsed.

The **console.log()** method can also be used to quickly output a message or write data to the browser console.

LG-MACALALAD-CS3-8.1 CS 3 Page **4** of **9**





Note: Please check out the browser console by pressing the **F12** or **Fn + F12 key** on the keyboard then select the "Console" tab.

There are few common rules of syntax and conventions that work their way through all JavaScript.

- Case-sensitive. The function names, variables, language keywords, and all other identifiers have to be written continuously with a consistent capitalization of letters. For example, the variable named "myVariable", "myvariable", and "MYVariable" are treated as three different objects.
- If white space, like tabs and spaces, is not part of a text string and is enclosed in quotations, it will be ignored.
- When you use JavaScript, there are several coding conventions that you need to consider:
 - a. Naming Variable

For identifiers names, use camelCase. Names should all begin with a letter or underscore.

```
firstName = "Juan";
lastName = "dela Cruz";

fee = 20.00;
tax = 0.21;
```

b. Line Length

Be sure to avoid lines longer than 80 characters for readability. But if your JavaScript statement doesn't fit it into a single line, after a comma or an operator, the best place to split it.

```
document.getElementById(example).innerHTML =
"Hello Pisay.";
```

LG-MACALALAD-CS3-8.1 CS 3 Page **5** of **9**





Note: .getElementById(ID) is a document object method that returns the element with the id value ID.

innerHTML - this property is used to set or return the HTML content within an element.

c. Spaces around operators

Often put spaces in your JavaScript code between operators (=++-*/) because it makes it look nice and easy to interpret.

```
var totalFee = fee + (fee * tax);
```

- d. To avoid confusion, always use *lowercase filenames*.
- e. Statement Rules
 - Always terminate or end a simple statement with a semicolon.
 - At the end of the first line, placed an opening bracket.
 - Before the opening bracket, use one space.
 - Without leading spaces, placed the closing bracket on a new line.

Refer to the given example below:

```
<html>
<head>
<body>
<script>
   function compute() {
     fee = 20.00;
      tax = 0.21;
      var totalFee = fee + (fee * tax);
          return totalFee;
    }
        var firstName = "Juan",
            lastName = "dela Cruz";
                                      document.write("Customer
                         name: " +
                                     firstName + " " + lastName
                         + "</br>");
            document.write("Your Total Fee is:" + compute());
</script>
</body>
</html>
```

LG-MACALALAD-CS3-8.1 CS 3 Page **6** of **9**



At the end of the statement, a semicolon informs JavaScript that it is the end of a command. Ending each statement with a semicolon is a good practice, while line break can also trigger a command's end based on the JavaScript standard.

The following is the result of the sample code above –



Comments in JavaScript

You can leave comments throughout a JavaScript, allowing you to leave reminders and explanations in your code. Comments may be useful if another developer is editing the code in the future.

The following are the two ways of incorporating comments in JavaScript

• For a single line comment, two slash characters (//) are used at the beginning of a line. You may place a single line comment on the same line as a declaration, as long as it comes after a statement. It is not appropriate to close a single line comment, as a line break effectively closes it.

```
// This is a comment with a single line.
```

• The same syntax that you learned from CSS can be extended to multi-line comments. The browser disregards everything inside the /**/ characters and considers it as a comment.

```
/* This is a comment with several lines.
Anything that goes here is completely ignored when the script
is executed.
There is a need to close this type of comment. */
```



NAVIGATE

Time Allocation: 8 mins.

Review Questions (non-graded)

Instruction: Write the letter which best corresponds to the correct answer on the space provided.

1. JavaScript code in an HTML document goes within which HTML element?

- a. <style>
- b. <script>
- c. <head>
- d.

LG-MACALALAD-CS3-8.1 CS 3 Page **7** of **9**



- 2. Which of the following is true for JavaScript?
 - a. It is a dynamic language.
 - b. It is a language of lightweight, interpreted programming.
 - c. It is case-sensitive
 - d. All of the above
- 3. Which of the following syntax is right when using JavaScript to print "hello"?
 - a. document.write ("hello");
 - b. print ("hello");
 - c. message.write ("hello");
 - d. display.message ("hello");
- 4. How do you create a single-line comment in JavaScript?
 - a. // single-line comment begin with two slashes
 - b. /* a single line comment begins with a slash followed by an asterisk.
 - c. // a single-line comment begins with two slashes followed by an asterisk and ends with an asterisk followed by a slash */
 - d. /* a single-line comment begins with a slash followed by an asterisk and ends with an asterisk followed by a slash */
- 5. True/False: Ending each JavaScript statement with a semicolon is a good practice.



Note: This is a non-graded activity.



KNOT

Time Allocation: 1 min.

In summary,

- JavaScript is a programming language that brings interactivity to our sites and custom behaviors. It is lightweight and commonly used as a part of web pages that allow client-side scripts to link and build dynamic web pages. It is an interpreted computer programming language with capabilities that are object-oriented.
- "LiveScript" was the initial name of JavaScript when it was first developed. The name was changed to JavaScript to enable its developer to position this new "unused language" as the "younger brother" of Java, when at that time it was the more popular language of the two.
- The script could be included in or referenced by an HTML document in the code to be translated by the browser. It implies that a web page requires not an inactive HTML, but it can incorporate client-interacting programs, control the browser, and make the content of the HTML powerful.
- You can implement JavaScript using its statements that are placed on the web page inside the HTML tags <script>...</script>.

LG-MACALALAD-CS3-8.1 CS 3 Page **8** of **9**





An Introduction to JavaScript: (n.d.). The Modern JavaScript Tutorial. https://javascript.info/intro

JavaScript tutorial.(n.d.). ExJS, ggplot2, Python Data Persistence, Caffe2, PyBrain, Python Data Access, H2O, Colab, Theano, Flutter, KNime, Mean.js, Weka, Solidity. https://www.tutorialspoint.com/javascript/index.htm

Sasha Vodnik and Don Gosselin(2015). Introduction to JavaScript. In Sasha Vodnik and Don Gosselin (Sixth Edition), JavaScript (1-72). Australia, Brazil, Mexico, Singapore, United Kingdom, United States, Cengage Learning

Terry Felke-Morris (2015). A Brief Look at JavaScript and jQuery. In Terry Felke-Morris (Seventh Edition), Web Development and Design Foundations with HTML5 (575-634). England, Pearson

Prepared by:

LEO R. MACALALAD SST-III PSHS-CALABARZON Reviewed by:

ALINE TERESA MENDOZA SST V PSHS- MC

LG-MACALALAD-CS3-8.1 CS 3 Page **9** of **9**

^{© 2020} Philippine Science High School System. All rights reserved. This document may contain proprietary information and may only be released to third parties with the approval of management. Document is uncontrolled unless otherwise marked; uncontrolled documents are not subject to update notification.