

GOVERNMENT PROPERTY
NOT FOR SALE

Technical Vocational Livelihood

11

QUARTER

2

Computer Programming



01000011011011110110110111000001110101
0111010001100101011100100010000001010000
0111001001101111011001110111001001100001
0110110101101101011010010110111001100111

Computer Programming (ICT) – Grade 11
Quarter 2 – Module 11: JavaScript Functions
First Edition, 2020

Republic Act 8293, section 176 states that: No copyright shall subsist in any work of the Government of the Philippines. However, prior approval of the government agency or office wherein the work is created shall be necessary for exploitation of such work for profit. Such agency or office may, among other things, impose as a condition the payment of royalties.

Borrowed materials (i.e., songs, stories, poems, pictures, photos, brand names, trademarks, etc.) included in this module are owned by their respective copyright holders. Every effort has been exerted to locate and seek permission to use these materials from their respective copyright owners. The publisher and authors do not represent nor claim ownership over them.

Published by the Department of Education Division of Pasig City

Development Team of the Module

Writer : Dan Reinnier C. Amigo
Editor : Ma. Lerma I. Cantanero
Reviewer : Rowena O. Dimagiba
Illustrator:
Layout Artist:
Management Team: Ma. Evalou Concepcion A. Agustin
OIC-Schools Division Superintendent
Dr. Aurelio G. Alfonso
OIC-Assistant Schools Division Superintendent
Dr. Victor Javena
OIC – Chief Curriculum Implementation Division

Education Program Supervisors

Librada L. Agon EdD (EPP/TLE/TVL/TVE)
Liza A. Alvarez (Science/STEM/SSP)
Bernard R. Balitao (AP/HUMSS)
Joselito E. Calios (English/SPFL/GAS)
Norlyn D. Conde EdD (MAPEH/SPA/SPS/HOPE/A&D/Sports)
Wilma Q. Del Rosario (LRMS/ADM)
Ma. Teresita E. Herrera EdD (Filipino/GAS/Piling Larang)
Perlita M. Ignacio PhD (EsP)
Dulce O. Santos PhD (Kindergarten/MTB-MLE)
Teresita P. Tagulao EdD (Mathematics/ABM)

Printed in the Philippines by _____

Department of Education – Division of Pasig City

Office Address: Caruncho Avenue, San Nicolas, Pasig City
Telefax: 641-88-85 / 682-2819
E-mail Address: divisionofpasig@gmail.com



Computer Programming

11

QUARTER 2

MODULE

11

JavaScript Functions

Writer : Dan Reinnier C. Amigo
Editor : Ma. Lerma I. Cantanero
Validator/Reviewer : Rowena O. Dimagiba



Introductory Message

For the Facilitator:

Welcome to the Computer Programming for the ICT Module on JavaScript Functions!

This module was collaboratively designed, developed and reviewed by educators from Schools Division Office of Pasig City headed by its Officer-In-Charge Schools Division Superintendent, Ma. Evalou Concepcion A. Agustin in partnership with the Local Government of Pasig through its mayor, Honorable Victor Ma. Regis N. Sotto. The writers utilized the standards set by the K to 12 Curriculum using the Most Essential Learning Competencies (MELC) while overcoming their personal, social, and economic constraints in schooling.

This learning material hopes to engage the learners into guided and independent learning activities at their own pace and time. Further, this also aims to help learners acquire the needed 21st century skills especially the 5 Cs namely: Communication, Collaboration, Creativity, Critical Thinking and Character while taking into consideration their needs and circumstances.

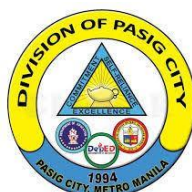
In addition to the material in the main text, you will also see this box in the body of the module:



Notes to the Teacher

This contains helpful tips or strategies that will help you in guiding the learners.

As a facilitator you are expected to orient the learners on how to use this module. You also need to keep track of the learners' progress while allowing them to manage their own learning. Moreover, you are expected to encourage and assist the learners as they do the tasks included in the module.



For the Learner:

Welcome to the Computer Programming for the ICT Module on JavaScript Functions!

The hand is one of the most symbolized part of the human body. It is often used to depict skill, action and purpose. Through our hands we may learn, create and accomplish. Hence, the hand in this learning resource signifies that you as a learner is capable and empowered to successfully achieve the relevant competencies and skills at your own pace and time. Your academic success lies in your own hands!

This module was designed to provide you with fun and meaningful opportunities for guided and independent learning at your own pace and time. You will be enabled to process the contents of the learning material while being an active learner.

This module has the following parts and corresponding icons:



Expectation - These are what you will be able to know after completing the lessons in the module



Pre-test - This will measure your prior knowledge and the concepts to be mastered throughout the lesson.



Recap - This section will measure what learnings and skills that you understand from the previous lesson.



Lesson- This section will discuss the topic for this module.



Activities - This is a set of activities you will perform.



Wrap Up- This section summarizes the concepts and applications of the lessons.



Valuing-this part will check the integration of values in the learning competency.



Post-test - This will measure how much you have learned from the entire module. Ito po ang parts ng module.





EXPECTATIONS

At the end of this module the learners is expected to:

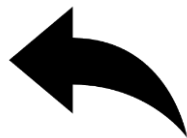
1. understand JavaScript function;
2. compare function name and variable name;
3. create function names; and,
4. execute a code using a function.



PRETEST

Directions: On the space provided before each number, write TRUE if the statement is correct otherwise write FALSE.

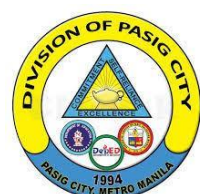
- _____ 1. JavaScript functions are containers for storing data values.
- _____ 2. JavaScript variable is a block of code designed to perform a particular task.
- _____ 3. Local variables can only be accessed even outside the function.
- _____ 4. Function is the keyword that starts declaring a function.
- _____ 5. A function doesn't have to have parameters.



RECAP

JavaScript variables are used to store data values. Describe JavaScript variables.

JavaScript operators are often used to compute these variables. What are the kinds of operators? Give a few examples for each.





LESSON

JavaScript Functions

A JavaScript function is a block of code designed to perform a particular task. These are blocks of code that can be named and reused. It is executed when "something" invokes it (calls it).

Declaring a Function

A JavaScript function is defined with the function keyword, followed by a name, followed by parentheses (). Function names can contain letters, digits, underscores, and dollar signs (same rules as variables). The parentheses may include parameter names separated by commas. The code to be executed, by the function, is placed inside curly brackets: { }

```
1 function addTwoNumbers(x, y)
2 {
3     return x + y;
4 }
```

Image 1 <https://www.javascript.com/learn/functions>

- ❖ **function** is the *keyword* that *starts declaring a function*.
- ❖ **addTwoNumbers** is the *function's name*, which is customizable — just like variable names.
- ❖ **(x, y)** are *parameters*, variable names for the inputs a function will accept.
- ❖ **return** is the *keyword* that *exits the function* and *shares an optional value outside*.

Function parameters are listed inside the parentheses () in the function definition. Function arguments are the values received by the function when it is invoked. Inside the function, the arguments (the parameters) behave as local variables.

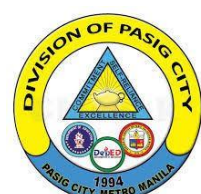
Function Invocation

The code inside the function will execute when "something" **invokes** (calls) the function:

- ❖ When an event occurs (when a user clicks a button)
- ❖ When it is invoked (called) from JavaScript code
- ❖ Automatically (self invoked)

Using Functions

When JavaScript reaches a return statement, the function will stop executing. If the function was invoked from a statement, JavaScript will "return" to execute the code after the invoking statement. Functions often compute a return value. The return value is "returned" back to the "caller".



Note that a function doesn't have to have parameters.

Example:

```
function greetThePlanet() {  
    return "Hello world!";  
}  
greetThePlanet();
```

Image 2.1
<https://www.javascript.com/learn/functions>

Output:

```
"Hello world!"
```

Image 2.2
<https://www.javascript.com/learn/functions>

If a function does have parameters, you'll need to provide values inside the parentheses when using the function.

Example: Calculate the product of two numbers, and return the result:

```
var x = myFunction(4, 3); // Function is called, return value will end up in x  
  
function myFunction(a, b) {  
    return a * b; // Function returns the product of a and b  
}
```

Image 3.1 https://www.w3schools.com/js/js_functions.asp

Output: The result in x will be:

```
12
```

Image 3.2 https://www.w3schools.com/js/js_functions.asp

Why use Functions?

- You can reuse code: Define the code once, and use it many times.
- You can use the same code many times with different arguments, to produce different results.

The () Operator Invokes the Function

Example: Convert Fahrenheit to Celsius

```
function toCelsius(fahrenheit) {  
    return (5/9) * (fahrenheit-32);  
}  
document.getElementById("demo").innerHTML = toCelsius(77);
```

Image 4.1 https://www.w3schools.com/js/js_functions.asp

Output:

```
25
```

Image 4.2 https://www.w3schools.com/js/js_functions.asp



Using the example above, **toCelsius** refers to the function object, and **toCelsius()** refers to the function result. Accessing a function without () will return the function object instead of the function result.

Example Accessing a function without (): Convert Fahrenheit to Celsius

```
function toCelsius(fahrenheit) {  
    return (5/9) * (fahrenheit-32);  
}  
document.getElementById("demo").innerHTML = toCelsius;
```

Image 5.1 https://www.w3schools.com/js/js_functions.asp

Output:

```
function toCelsius(f) { return (5/9) * (f-32); }
```

Image 5.2 https://www.w3schools.com/js/js_functions.asp

Functions Used as Variable Values

Functions can be used the same way as you use variables, in all types of formulas, assignments, and calculations.

Instead of using a variable to store the return value of a function as in this example:

```
var x = toCelsius(77);  
var text = "The temperature is " + x + " Celsius";
```

Image 6.1 https://www.w3schools.com/js/js_functions.asp

You can use the function directly:

```
var text = "The temperature is " + toCelsius(77) + " Celsius";
```

Image 6.2 https://www.w3schools.com/js/js_functions.asp

Local Variables

Variables declared within a JavaScript function, become LOCAL to the function. Local variables can only be accessed from within the function.

```
// code here can NOT use carName  
  
function myFunction() {  
    var carName = "Volvo";  
    // code here CAN use carName  
}  
  
// code here can NOT use carName
```

Image 7 https://www.w3schools.com/js/js_functions.asp



Since local variables are only recognized inside their functions, variables with the same name can be used in different functions. Local variables are created when a function starts, and deleted when the function is completed.



ACTIVITIES

Using functions, provide for the syntax to the following problems.

Example Problem: Convert Fahrenheit to Celsius

Syntax:

```
function toCelsius(f) {  
    return (5/9) * (f-32);  
}  
document.getElementById("demo").innerHTML = toCelsius(77);
```

1. Get the output “Computer PROGRAMMING”
2. Print your name and, (your name above your section)
3. Print your section (section under your name)

Your Name

No. 2 and 3 should have this output

Your Section

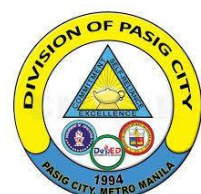
4. Declare the function as variable. Calculate the difference of two numbers (11 and 12), and return the result (answer should be negative).
5. Declare the function as variable. Combine strings. Output should be: “Jose Rizal”



WRAP-UP

This module discussed the JavaScript Functions, it’s syntax and how to use it. Relate the following words and phrases inside the word pool to summarize the what you have learned in this lesson.

Variable	Function	function keyword	function name
	parameter	argument	block of codes
return	declare	value	caller
			invoke





VALUING

Directions: Read and answer the following questions carefully in two to three sentences for each number.

1. Why do you think programmers are taught how to use JavaScript function?

2. In writing codes, how important is it that you can identify the difference between variables and functions?

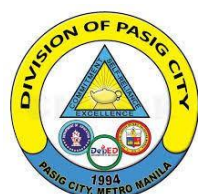
3. Give an example where using functions is necessary?



POSTTEST

Directions: Fill in the blanks to complete the following statements.

1. JavaScript function is executed when "something" _____ it (calls it).
2. To declare a JavaScript function, we use the _____ keyword.
3. Function arguments are the values _____ by the function when it is invoked.
4. The function will stop executing when JavaScript reaches a _____ statement.
5. _____ can only be accessed from within the function.





KEY TO CORRECTION

Pre-test:	Post-test:
1. False	1. invokes
2. False	2. function
3. False	3. received
4. True	4. return
5. True	5. Local variables

Activity: (function name per students may vary)

1.

```
function SubjectName() {  
    return "Computer PROGRAMMING";  
}  
SubjectName();
```

2. and 3.

```
<p id="name"></p>  
<p id="sec"></p>  
<script>  
    function myName() {  
        return "Your Name";  
    }  
    function mySection() {  
        return "Your Section";  
    }  
    document.getElementById("name").innerHTML = myName();  
    document.getElementById("sec").innerHTML = mySection();  
</script>
```

4.

```
var x = minus(1, 12);  
function minus(a, b) {  
    return a - b;  
}
```

5.

```
<p id="demo"></p>  
<script>  
    var x = join("Jose", " ", "Rizal");  
    document.getElementById("demo").innerHTML = x;  
    function join(a, b, c) {  
        return a + b + c;  
    }  
</script>
```

Wrap-up (students may come up with their own paragraph)

A JavaScript **function** is a **block of code** designed to perform a particular task. It is executed when something **invokes** it. A JavaScript function is defined with the **function keyword**, followed by a name, followed by parentheses ().

Function names can contain letters, digits, underscores, and dollar signs (same rules as **variables**). The parentheses may include **parameter names** separated by commas. Function **arguments** are the values received by the function when it is invoked. When JavaScript reaches a **return** statement, the function will stop executing. The return value is "returned" back to the "**caller**".

Variables **declared** within a JavaScript function, become LOCAL to the function.

References

- "JavaScript Functions". Accessed September 3, 2020 10:10am.
https://www.w3schools.com/js/js_functions.asp
- "Functions". Accessed September 3, 2020 10:12am
<https://www.javascript.com/learn/functions>

Photo Credits:

- Image 1 <https://www.javascript.com/learn/functions>
- Image 2.1 <https://www.javascript.com/learn/functions>
- Image 2.2 <https://www.javascript.com/learn/functions>
- Image 3.1 https://www.w3schools.com/js/js_functions.asp
- Image 3.2 https://www.w3schools.com/js/js_functions.asp
- Image 4.1 https://www.w3schools.com/js/js_functions.asp
- Image 4.2 https://www.w3schools.com/js/js_functions.asp
- Image 5.1 https://www.w3schools.com/js/js_functions.asp
- image 5.2 https://www.w3schools.com/js/js_functions.asp
- Image 6.1 https://www.w3schools.com/js/js_functions.asp
- Image 6.2 https://www.w3schools.com/js/js_functions.asp
- Image 7 https://www.w3schools.com/js/js_functions.asp

