**JavaScript Tasks**

Table of Contents

[Index File 2](#_Toc26389139)

[Task 01: Variables, Constants, Operators & Core Data Types 2](#_Toc26389140)

[Task 02: Functions 2](#_Toc26389141)

[Task 03: Simple Calculator 2](#_Toc26389142)

[Task 04: Simple Calculator - Arrays 3](#_Toc26389143)

[Task 05: Simple Calculator – Object Log 4](#_Toc26389144)

[Task 06: Simple Calculator – Conditional If 4](#_Toc26389145)

[Task 07: Monster Killer – Game 5](#_Toc26389146)

[Task 8: ROCK – PAPER – SCISSORS Game 7](#_Toc26389147)

# Index File

Run the index.html file to navigate the tasks/example

# Task 01: Variables, Constants, Operators & Core Data Types

**Html File:** Tasks\task\_01\_Variables\_Constants\_Operators\_CoreDataTypes.html

**JS File:** Assets\js\js\_01\_Variables\_Constants\_Operators\_CoreDataTypes.js

JavaScript is referenced towards the end, just before the ending body tag.



Take a look at the html file for the tasks that need to be completed and then look at the JavaScript file for the solution.

# Task 02: Functions

**Html File:** Tasks\task\_02\_Functions.html

**JS File:** Assets\js\js\_02\_functions.js

JavaScript is referenced towards the end, just before the ending body tag.



Take a look at the html file for the tasks that need to be completed and then look at the JavaScript file for the solution.

# Task 03: Simple Calculator

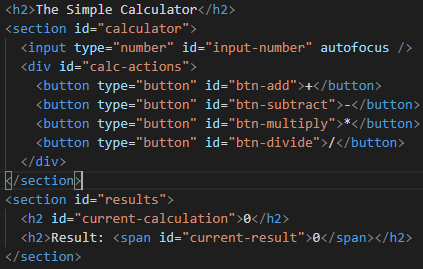
**Html File:** Tasks\task\_03\_simple\_calculator.html

**JS File:** Assets\js\js\_03\_simple\_calculator.js

JavaScript is referenced towards the end, just before the ending body tag.



The calculator Html



Take a look at the html file for the tasks that need to be completed and then look at the JavaScript file for the solution.

# Task 04: Simple Calculator - Arrays

**Html File:** Tasks\task\_04\_simple\_calculator\_arrays.html

**JS File:** Assets\js\js\_04\_simple\_calculator\_arrays.js

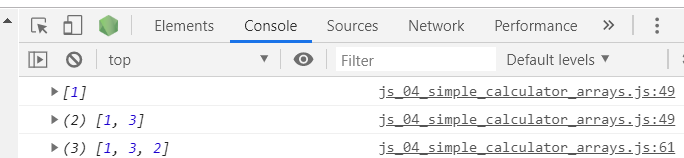
JavaScript is referenced towards the end, just before the ending body tag.



This example builds on Task 3, only a few lines have been added to store the number entered in an array and then display it via console.log(…)The calculator Html

Take a look at the html file for the tasks that need to be completed and then look at the JavaScript file for the solution.

Console Log



# Task 05: Simple Calculator – Object Log

**Html File:** Tasks\task\_05\_simple\_calculator\_object\_log.html

**JS File:** Assets\js\js\_05\_simple\_calculator\_object\_log.js

JavaScript is referenced towards the end, just before the ending body tag.



This example builds on Task4.

We have created an object and then pushed this object into the log array and then displayed the array in the console.

|  |  |
| --- | --- |
| Result in console full array | Result in console single item |
|  |  |

# Task 06: Simple Calculator – Conditional If

**Html File:** Tasks\task\_06\_simple\_calculator\_conditional\_if.html

**JS File:** Assets\js\js\_06\_simple\_calculator\_conditional\_if.js

JavaScript is referenced towards the end, just before the ending body tag.



This example builds on Task 5. We have a lot of repeating code like functions for add, subtract, divide and multiply. We’ll create a single function for calculation and will pass in the proper parameters from these functions.

Take a look at the html file for the tasks that need to be completed and then look at the JavaScript file for the solution.

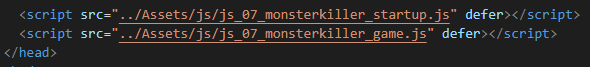
Have introduced a string method .toLowerCase() here as well. You can read more about string methods by going to this link: <https://www.w3schools.com/js/js_string_methods.asp>

You can implement the same with **switch** statement. See if you can follow the below article and can do it: <https://www.w3schools.com/js/js_switch.asp>

# Task 07: Monster Killer – Game

This game covers all that we have learned from Task 01-06.

JavaScript files are referenced in the head section



There are two JavaScript files. Only work under the ***\_game js file***.

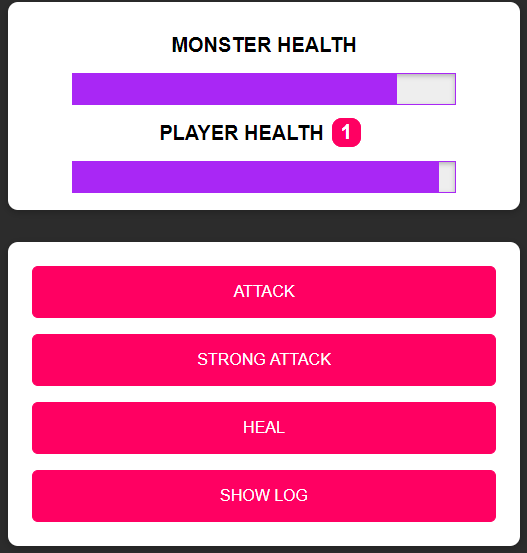
The **\_startup.js** we haven't covered yet so use as is. You can still go ahead and try to understand and do online research, the best way to learn new stuff.

The html file has been built for this as well and all tasks will be added towards the bottom.

This task uses:

* Ternary operator: It is a single line if statement, [check here](#_Ternary_Operator_–) for more details.

Once all wired up, run the app and also look at the console in developer tools by clicking the Show Log button.



|  |  |
| --- | --- |
| **Custom log print with loops** | **Log print with simple console.log** |
|  |  |

# Task 8: ROCK – PAPER – SCISSORS Game

**Html File:** Tasks\task\_08\_rock-paper-scissors\_game.html

**JS File:** Assets\js\js\_08\_rock-paper-scissors\_game.js

This game covers all that we have learned from Task 01-07.

JavaScript files are referenced in the head section



We have 3 radio buttons the get the user input. Then the user clicks the Game button. The javascript check the users choice against the computer choice and then declares a winner.

# Task 09: Simple Calculator – Using Bind()

**Html File:** Tasks\task\_09\_simple\_calculator\_using\_bind.html

**JS File:** Assets\js\js\_09\_simple\_calculator\_using\_bind.js

JavaScript is referenced in the head section.



This example builds on Task 5.

Create a single operations functions instead of Add/Multiply/Subtract/Divide. Pass the operation to perform via Bind() when creating the event listener.