



* ABOUT THIS GUIDE 5

**TABLE OF CONTENT**

* LEGEND 6
* SESSION SCHEDULE 7
* TOPIC 1: GETTING STARTED 9
* SUBTOPIC 1: ICE BREAKER 9
* SUBTOPIC 2: IN THIS SESSION – SESSION OBJECTIVES 11
* TOPIC 2 – INTRODUCTION TO JAVASCRIPT 12
* SUBTOPIC 1: WHAT IS JAVASCRIPT? 12
* SUBTOPIC 2: JAVASCRIPT SYNTAX 13
* SUBTOPIC 3: ATTRIBUTES OF SCRIPT TAGS 14
* TOPIC 3 – WHAT IS PROGRAMING ALL ABOUT? 16
* TOPIC 4: HOW TO ADD A SCRIPT IN YOUR PAGE? 17
* SUBTOPIC 1: WHERE TO ADD THE JAVASCRIPT CODES? 17
* SUBTOPIC 2: EXAMPLE 17
* TOPIC 5: STARTING WITH PROGRAMMING 22
* SUBTOPIC 1: VARIABLES 22
* SUBTOPIC 2: OPERATORS 23
* SUBTOPIC 3: FUNCTIONS 26
* SUBTOPIC 4: CONDITIONAL STATEMENTS 28
* SUBTOPIC 5: LOOPING 31
* TOPIC 6: DOCUMENT OBJECT MODEL 34
* SUBTOPIC 1: USING DOT NOTATION 34
* SUBTOPIC 2: DOCUMENT OBJECT 35
* SUBTOPIC 3: FORM COLLECTION 36
* SUBTOPIC 4: FORM ELEMENTS 37
* SUBTOPIC 5: IMAGES COLLECTION 40
* SUBTOPIC 6: DIFFERENT TYPES OF OBJECTS 43
* TOPIC 7: JAVASCRIPT- OBJECT ORIENTED PROGRAMMING 46
* SUBTOPIC 1: IN THIS SESSION – SESSION OBJECTIVES 46
* TOPIC 8 – OBJECTS-BASED PROGRAMMING CONCEPTS 47
* SUBTOPIC 1: WHAT ARE OBJECTS? 47
* TOPIC 9: BUILDING OBJECTS IN JAVASCRIPT 49
* SUBTOPIC 1: CREATE NEW OBJECTS 49
* SUBTOPIC 2: SET PROPERTIES OF NEW OBJECTS 50
* TOPIC 10: JAVASCRIPT AND HTML 53
* TOPIC 11: VARIABLES AND DATA TYPES 54
* SUBTOPIC 1: VARIABLES 54
* SUBTOPIC 2: DATA TYPES 55
* TOPIC 12: CREATING AND MANIPULATING ARRAYS 57
* TOPIC 13: MANAGING EVENTS 61
* TOPIC 14: SUMMARY 63
* SUBTOPIC 1: OBJECTIVES MAPPING ACTIVITY 63
* SUBTOPIC 3: FINAL TIPS AND SUMMARIZATION 65
* TOPIC 15: LAB EXERCISES 67

This Participant’s Guide addresses an ILT program of 100 hours duration based on the **ITES Functional Skills Training** course. The program can be divided into 10 sessions.

**ABOUT THIS GUIDE**

This Participant’s Guide will help the participants gain a thorough understanding of the basic functional skills required in the ITES industry.

The features of this guide are as follows:

* It provides hands on experience of the basic Web Design functional skills that are required in the ITES industry.
* It provides references to the page numbers of the ILT version of this program. The instructor will discuss the points as mentioned in this guide when displaying the various screens in the course.
* It provides spaces after each screen content for the participants to note down any important points being explained by the instructor or write their own comments related to the same content.
* It includes activities that you will perform in the class and instructions to carry out the activities.
* It includes a ‘References’ section that provides a list of books and other sources that can help you enhance your knowledge on the concepts that are taught in the class.
* It has a ‘Pre-Reading Material’ section that contains concepts that you need to study before the related topic is taught in the class.
* It provides the approximate duration of each topic. However, the duration may vary depending on the length of discussion and/or questions that you as learners may have on a particular topic.
* It includes an explanation for the icons used in the guide. These icons will help you distinctly identify the instructions that need to be followed to understand any particular section. The icons are explained on the next page.

The following legend explains the various icons used in this guide.

**LEGEND**

|  |  |
| --- | --- |
| Icon | Description |
|  | Approximate duration of each section |
|  | Project course screen |
|  | Discussion |
|  | Group Activity |
|  | Activity |
|  | Important |

**SESSION SCHEDULE**

|  |  |  |
| --- | --- | --- |
| **Topic** | **Sub-topic** | **Duration in minutes** |
| 1. Getting Started | * Ice Breaker | 15 minutes |
| 1. Introduction to Javascript | * What is JavaScript? * JavaScript Syntax * Attributes of Script Tags | 15 minutes |
| **Total duration** | | **30 minutes** |
| 1. Getting Started | * Ice Breaker | 15 minutes |
| 1. What is programing all about? |  | 10 minutes |
| 1. How to add a script in your page? | * Where to add the JavaScript codes? * Example | 10 minutes |
| 1. Starting With Programming | * Variables * Operators * Functions * Conditional Statements * Looping | 60 minutes |
| 1. Document Object Model | * Using Dot Notation * Document Object * Form Collection * Form Elements * Images Collection * Different Types of Objects | 60 minutes |
| 1. Summary | * Recap * Final tips and summarization | 10 minutes |
| **Total duration** | | **170 minutes** |
| 1. Object Oriented Programming | * Ice Breaker | 15 minutes |
| 1. Object-based Programming Concepts | * What are objects? | 10 minutes |
| 1. Building Objects in Java | * Create new objects * Set properties of new objects | 10 minutes |
| 1. JavaScript and HTML | * NA | 10 minutes |
| 1. Variables and Data Types | * Variables * Data Types | 10 minutes |
| 1. Creating and Manipulating Arrays | * NA | 10 minutes |
| 1. Managing Events | * NA | 10 minutes |
| 1. Summary | * Recap * Final tips and summarization | 10 minutes |
| 1. Check Your Understanding |  |  |
| **Total duration** | | **120 minutes** |

# 

**SUBTOPIC 1: ICE BREAKER**

**TOPIC 1: GETTING STARTED**

The ILT course will be projected.



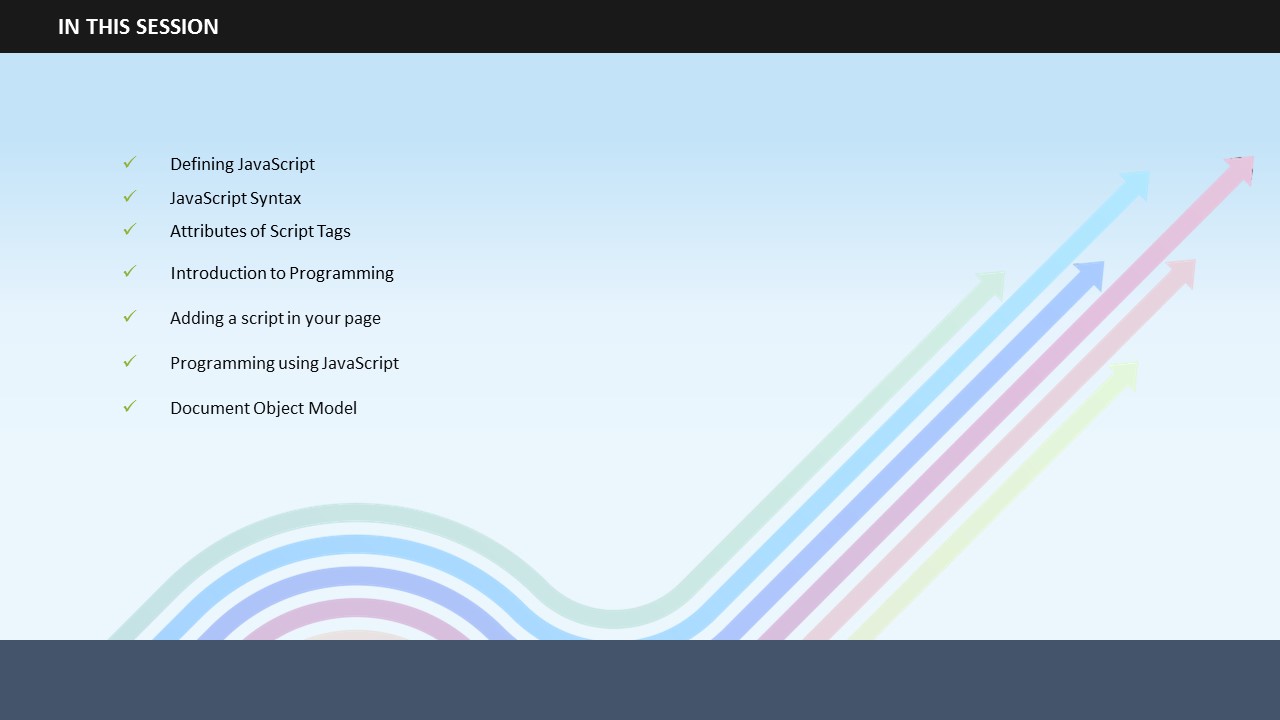
Take part in the activity described below:

**Lost on a Deserted Island (Part 1)…**

You have been travelling on a ship and it got caught in a very bad storm. Following a shipwreck, you find yourself stranded on remote deserted island.

**Part 1:** If given a choice in this situation to carry one object, what would you choose and why?

**Part 2:** Work together in a group to improve your chances of survival on the island by combining your objects together. If necessary, your group can add more objects to the list, but it is mandatory to use the objects already chosen by each group member.



**Your space:**

|  |
| --- |
|  |
|  |
|  |
|  |
|  |

## 

**SUBTOPIC 2: IN THIS SESSION – SESSION OBJECTIVES**

In this session, you will learn:

* How to define JavaScript?
* What is the syntax of JavaScript?
* What are the attributes of script tags?
* You will also learn about the attributes of script tags.
* What is programming?
* How to add a script in your page?
* How to start programming using JavaScript?
* What is Document Object Model?

**Your space:**

|  |
| --- |
|  |
|  |
|  |
|  |
|  |
|  |
|  |

# 



**INTRODUCTION TO JAVASCRIPT**

**SUBTOPIC 1: WHAT IS JAVASCRIPT?**



1. JavaScript is the most popular programming language in the world.
2. It is used to add interactivity to documents such as creating games, enabling movement of things on the screen, interaction between webcam, microphone etc., events like a mouse click, modification of HTML and CSS of page after loading, and many more exciting and interesting things.
3. JavaScript and Java are completely different languages, both in concept and design.

## 

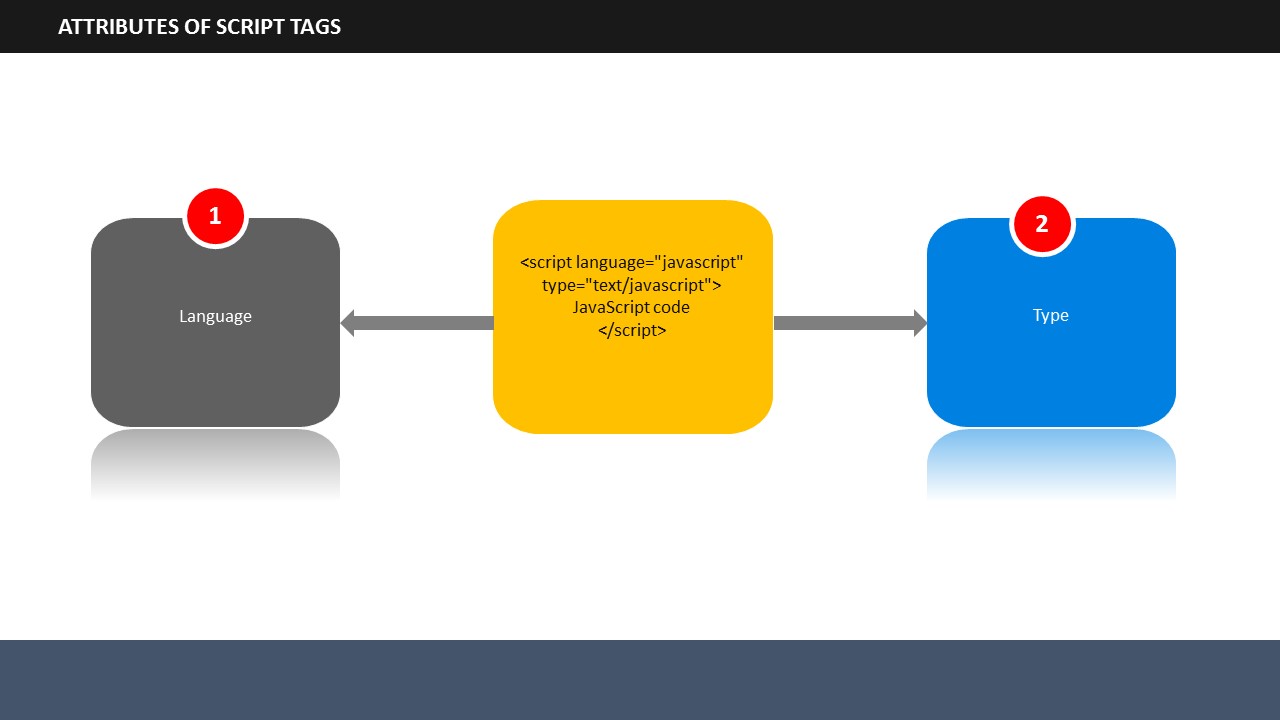
**SUBTOPIC 2: JAVASCRIPT SYNTAX**

1. JavaScript consists of JavaScript statements placed within the <script>... </script> HTML tags.
2. The <script> tags are placed anywhere within the web page though it is preferable to keep it within the <head> tags. The <script> tags tell the browser to begin interpreting the text between these tags as a script.
3. The syntax of JavaScript looks like this:

|  |
| --- |
| <script ...>  JavaScript code  </script> |

**Your space:**

|  |
| --- |
|  |
|  |
|  |
|  |



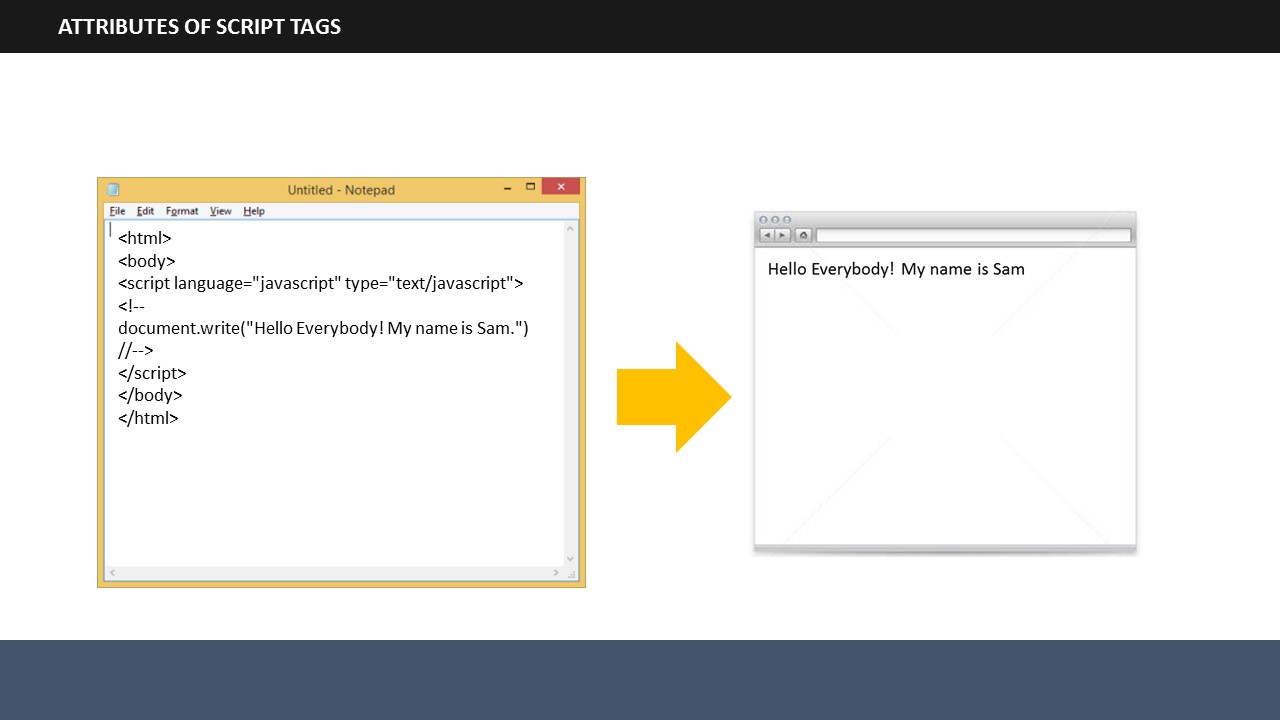
**SUBTOPIC 3: ATTRIBUTES OF SCRIPT TAGS**

1. A script tag has two attributes: Language and Type.
2. **Language**: It specifies the scripting language. Generally, its value is “javascript”.
3. **Type**: It specifies the scripting language in use. Its value is set to "text/javascript".
4. Using these two attributes the JavaScript segment will appear like this:

|  |
| --- |
| <script language="javascript" type="text/javascript">  JavaScript code  </script> |

**Your space:**

|  |
| --- |
|  |
|  |
|  |
|  |



1. If you want to print some text, for example “Hello Everybody! My name is Sam!” on your webpage, the script would look like this:

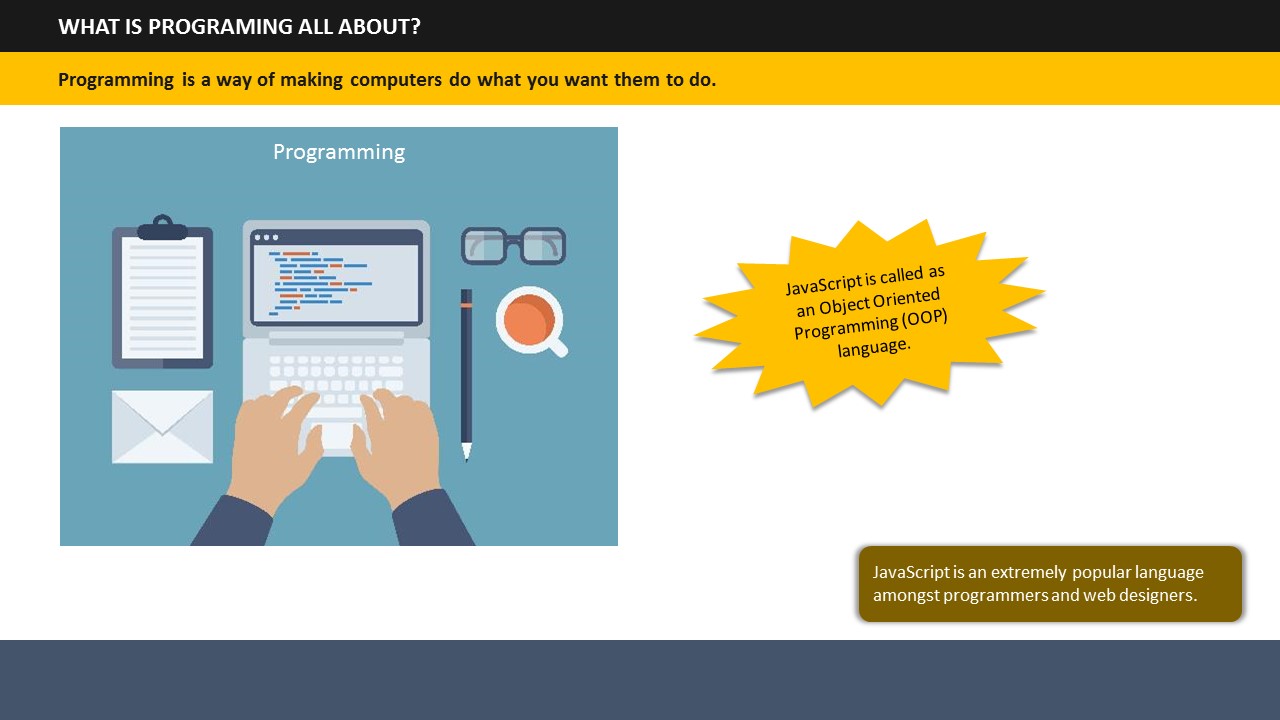
|  |
| --- |
| <html>  <body>  <script language="javascript" type="text/javascript">  <!--  document.write("Hello Everybody! My name is Sam.")  //-->  </script>  </body>  </html> |

1. Above code will display following result on your webpage.

|  |
| --- |
| Hello Everybody! My name is Sam |

**Your space:**

|  |
| --- |
|  |
|  |
|  |
|  |
|  |
|  |



**TOPIC 3 – WHAT IS PROGRAMING ALL ABOUT?**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 1. Programming is a way of making computers do what you want them to do. To do this, we need programming languages using which we communicate with computers. 2. You have already learned about HTML and CSS which are also used to do programming. 3. Another language that is used to do programming is JavaScript. It is extremely popular amongst programmers and web designers. 4. JavaScript called as an Object Oriented Programming or OOP language. In other words, JavaScript involves working with objects. 5. JavaScript can change HTML content, its attributes and style. It can also be used to validate data inputs.   **Your space:**   |  | | --- | |  | |  | |  | |  | |  |     **SUBTOPIC 1: WHERE TO ADD THE JAVASCRIPT CODES?**  **TOPIC 4: HOW TO ADD A SCRIPT IN YOUR PAGE?**     1. You can insert JavaScript in <head> section or <body> section or both head as well body sections of HTML pages. 2. It is advisable to place scripts at the bottom of the <body> section to improve page loading. 3. You can also use the script externally if you plan to use the same code in different web pages. 4. The syntax of JavaScript looks like this:  |  | | --- | | <script ...>  JavaScript code  </script> |   **SUBTOPIC 5: jQuery- CSS classes**  **Inserting Script in <Head> Section**     1. The following example shows the JavaScript placed in the <head> section of an HTML page:  |  | | --- | | <!DOCTYPE html>  <html>  <head>  <script>  function myFunction() {  document.getElementById("demo").innerHTML = "I am fine. Thank you!";  }  </script>  </head>  <body>  <h1>JavaScript in Head Section</h1>  <p id="demo">Hello! How are you?</p>  <button type="button" onclick="myFunction()">Try it</button>  </body>  </html> |  1. Above code will display following result on your webpage.  |  | | --- | |  |  1. The function is invoked in this script when the Try It button is clicked. It will give the following result:  |  | | --- | |  |   **Inserting Script in <Body> Section**       1. The following example shows the JavaScript placed in the <body> section of an HTML page:  |  | | --- | | <!DOCTYPE html>  <html>  <body>  <h1>JavaScript in Body Section</h1>  <p id="demo">Would you like to have tea or coffee?</p>  <button type="button" onclick="myFunction()">Try it</button>  <script>  function myFunction() {  document.getElementById("demo").innerHTML = "Coffee, please.";  }  </script>  </body>  </html> |  1. Above code will display following result on your webpage.  |  | | --- | |  |  1. The function is invoked in this script when the Try It button is clicked. It will give the following result:  |  | | --- | |  |   **Your space:**   |  | | --- | |  | |  | |  | |  | |  | |  | |  | |  |  TOPIC 5: STARTING WITH PROGRAMMINGSUBTOPIC 1: VARIABLES    1. Variables can be seen as containers that can store data values. You can place data into these containers and name these containers. Whenever, you wish to refer to the stored data, you can access the required container. 2. Examples of variables.  |  | | --- | | var a = 101;  var b = 102;  var c = 103; |   Here a, b, and c are the containers which are storing their corresponding values 101, 102, 103.  **Your space:**   |  | | --- | |  | |  | |  | |  | |  |  SUBTOPIC 2: OPERATORS    1. JavaScript uses the following types of operators:  * Arithmetic Operators * Assignment Operators * Comparison Operators * Logical Operators      1. The arithmetic operators function as the addition, subtraction, multiplication, division, and modulus operators, and operate very similarly to other languages. 2. In addition to these basic arithmetic operators, there are two more operators which increase or decrease the value of variable by 1.      1. The assignment operators assign a value to a variable.      1. The comparison operators determine if the two operands or numbers meet the given condition.   **Your space:**   |  | | --- | |  | |  | |  | |  | |  |      1. The logical operators are *and*, *or*, and *not*. The and operator along with or operator accepts two operands or numbers and provides their associated logical result, while the third operator determines the inverse of the given value, and returns the boolean: true values become false, or false values become true.   **Your space:**   |  | | --- | |  | |  | |  | |  | |  | |  | |  | |  |  SUBTOPIC 3: FUNCTIONS    1. A Function in JavaScript refers to a group of reusable codes. This helps in saving time as a web designer will not be required to write the same codes over and over again. 2. A function is written using the following syntax:  |  | | --- | | function name(parameter1, parameter2, parameter3) {  code to be executed  } |  1. A function is executed when a call to that function is made anywhere within the script, the page, an external page, or by an event. 2. Functions can return values using the return keyword.     **Your space:**   |  | | --- | |  | |  | |  | |  | |  |      |  | | --- | | function sayHello() {  alert("hello!");  }  sayHello(); |  1. Example:  If you type the example code and preview your page in a browser, you will see hello! displayed. 2. Example:  |  | | --- | | function getDistance(speed, time) {  var distance = speed \* time;  return distance;  } |   **Your space:**   |  | | --- | |  | |  | |  | |  | |  |  SUBTOPIC 4: CONDITIONAL STATEMENTS    1. The conditional statements help in changing the flow of programs. In other words, they allow different actions to be taken for different decisions. 2. JavaScript supports the following conditional statements:  * **If**:You can use this statement to specify execution of a block of code if a specified condition is true. * **If…Else**: You can use this statement to specify execution of a block of code when the same condition is false. * **If...Else…If**: You can use this statement to specify new conditions to test provided the first condition is false.   **Your space:**   |  | | --- | |  | |  | |  | |  | |  |   Examples:     1. **If:** The syntax for if statement is shown on the screen. The example shows how if statements can be used to decide if a person is eligible for voting or not basis the age. If the entered age is greater than 18, the output will say “Eligible for voting”. In this case, the variable is 20 which is greater than 18, so the result would say, Eligible for voting.      1. **If…Else:** In the previous example, the variable was 20, so the result was true. But, if the variable is 17 which is less than 18, the statement will show the output associated with else part of the statement. That is “Not eligible for voting”.      1. **If…Else…If**: In the example, the entered variable is Basketball. The if statement will be false for this case since it defines tennis ball, while the variable is cricket ball. So, the decision goes to the first else…if statement which is the defining condition for cricket ball. Since the first else…if statement also doesn’t match, so the decision is passed to the second statement. The second else…if statement is defining the basketball, so our variable input matches with it. Hence the result on the screen would show as Basketball.   **Your space:**   |  | | --- | |  | |  | |  | |  | |  | |  | |  | |  |  SUBTOPIC 5: LOOPING    1. Loops are used to execute a code again and again, but every time with a different value. 2. JavaScript supports the following types of loops:  * For * For/in * While * Do/while  1. The for loop loops through a block of code multiple times. 2. The for/in loop loops through the properties of an object. 3. The while loop loops through a block of code as long as a specified condition is true. 4. The do/while loop executes the code block once before checking if the condition is true, then it will repeat the loop as long as the condition is true.  |  | | --- | |  |  1. The syntax for each loop is:  |  |  | | --- | --- | | For | for (initialization; test condition; iteration statement){  Statement(s) to be executed if test condition is true  } | | For/in | for (variablename in object){  statement or block to execute  } | | While | while (expression){  Statement(s) to be executed if expression is true  } | | Do/while | do{  Statement(s) to be executed;  } while (expression); |   **Your space:**   |  | | --- | |  | |  | |  | |  | |  | |  | |  | |  |  TOPIC 6: DOCUMENT OBJECT MODELSUBTOPIC 1: USING DOT NOTATION    1. The dot notation is used to access the properties of objects. You can read and write the properties of an object using dot notation. 2. For Example:  |  | | --- | | // Getting object properties  emp.name // ==> Rehman  emp.age // ==> 31  // Setting object properties  emp.name = "John" // <== John  emp.age = 20 // <== 20 |   **Your space:**   |  | | --- | |  | |  | |  | |  | |  | |  | |  | |  |  SUBTOPIC 2: DOCUMENT OBJECT    1. A document object is the HTML document shown in a browser window. The way in which document content is accessed and modified is called the Document Object Model or DOM. 2. The objects are generally organized in a hierarchy in an HTML document. The hierarchy is as follows: 3. **Window object**: It is at the top level in the hierarchy. 4. **Document object**: The document object that contains the contents of a page. Each HTML document loaded into a browser window becomes a document object. 5. **Form object**: Any object that is enclosed in the <form> tags falls into the category of form objects. 6. **Form control elements**: The form control elements contain all the elements that define a form object. For example, radio buttons, text fields, checkboxes etc.   **Your space:**   |  | | --- | |  | |  | |  | |  | |  |  SUBTOPIC 3: FORM COLLECTION    1. Document objects have certain properties. 2. One of such properties is form elements collection. The form collection returns a collection of all elements in a form. 3. The syntax for forms collection command is document.forms.length     **Your space:**   |  | | --- | |  | |  | |  | |  | |  | |  | |  | |  |  SUBTOPIC 4: FORM ELEMENTS   **TYPES OF FORM ELEMENTS**     1. Form elements can be of several types such as:  * Button * Checkbox * Hidden * Password * Radio * Reset * Select * FileUpload * Submit * Text * textArea  1. You can refer to a form element through its name or its index number in an array. 2. Each of these elements has its own properties in JavaScript.   **DESCRIPTION OF FORM ELEMENTS**     |  |  | | --- | --- | | Button | A push button that provides functionality other than submit or reset button | | Checkbox | A square shaped checkbox which can be checked on or off | | Hidden | A hidden field | | Password | A password text field in which each text entry appears as an asterisk (\*) | | Radio | A radio button similar to checkbox but round in shape | | Reset | A reset button | | Select | A selection list | | FileUpload | A file upload element that allows users to provide a file as input for a form submission | | Submit | A submit button | | Text | A text field | | textArea | A multiline text entry field |   **TYPE PROPERTIES**     1. Each of the form elements has a type property. This property is a string value that shows the type of input element. 2. The strings reflected by the various type properties for each form element are as follows:  * Button ->“button" * Checkbox -> "checkbox" * Hidden ->"hidden" * Password ->"password" * Radio ->"radio" * Reset ->"reset" * Select (list) ->“select-one” * FileUpload ->"file" * Submit ->"submit" * Text ->"text" * textArea ->“select-mutiple”   **Your space:**   |  | | --- | |  | |  | |  | |  | |  |  SUBTOPIC 5: IMAGES COLLECTION    1. The images collection returns a collection of all <img> elements in the document. 2. The syntax for image collection is document.images.length   **Your space:**   |  | | --- | |  | |  | |  | |  | |  |      1. For Example:  |  | | --- | | <!DOCTYPE html>  <html>  <body>  <img src="whiterose.jpg" alt="flower" width="150" height="113">  <img src="redrose.jpg" alt="flower" width="152" height="128">  <img src="pinkrose.jpg" alt="flower" width="42" height="42">  <p>Click the button to display the number of images in the document.</p>  <button onclick="myFunction()">Try it</button>  <p id="demo"></p>  <script>  function myFunction() {  var x = document.images.length;  document.getElementById("demo").innerHTML = x;  }  </script>  </body>  </html> |      1. Above code will display following result on your webpage.  |  | | --- | |  |  1. The function is invoked in this script when the Try It button is clicked. It will give the following result:  |  | | --- | |  |   **Your space:**   |  | | --- | |  | |  | |  | |  | |  | |  |  SUBTOPIC 6: DIFFERENT TYPES OF OBJECTS    1. JavaScript has two types of objects:    1. User-Defined Objects    2. Built-In Objects 2. The user-defined objects are created using a constructor function called Object(). The user-defined objects can be created using the Object() constructor. The constructor initializes a new object and assigns properties to it. 3. The built-in objects can be accessed from anywhere in your program. They will function in the same way irrespective of the browser used. The built-in objects available are:  * Number Object * Boolean Object * String Object * Array Object * Date Object * Math Object * Regular Expression Object   **Your space:**   |  | | --- | |  | |  | |  | |  | |





The ILT course will be projected.

# 

# 

**SUBTOPIC 1: IN THIS SESSION – SESSION OBJECTIVES**

**TOPIC 7: JAVASCRIPT- OBJECT ORIENTED PROGRAMMING**

In this session, you will learn about:

* Building objects in JavaScript
* JavaScript and HTML
* Data types, variables and operators
* JavaScript control structures
* Using loops
* Creating and manipulating arrays
* Using object hierarchy
* Managing events

**Your space:**

|  |
| --- |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |

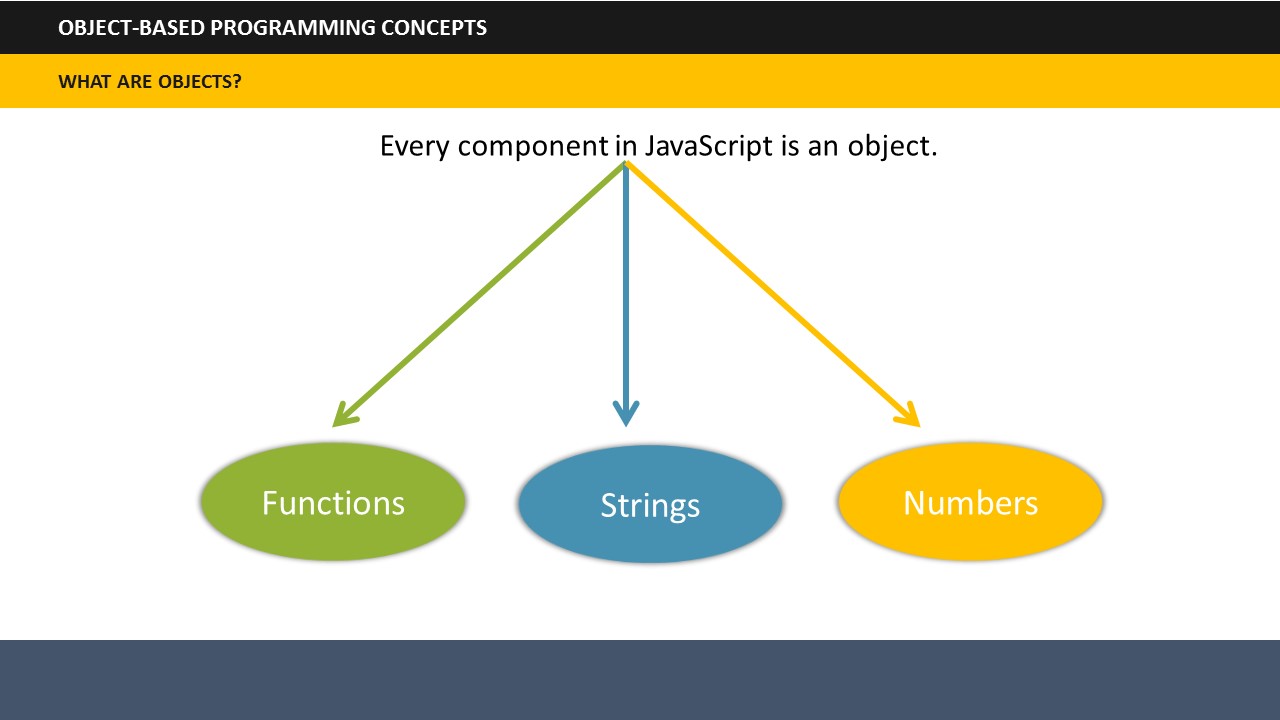
# 

**SUBTOPIC 1: WHAT ARE OBJECTS?**

**TOPIC 8 – OBJECTS-BASED PROGRAMMING CONCEPTS**

## 

1. JavaScript is called as an Object Oriented Programming or OOP language. In other words, JavaScript involves working with objects.



1. Every component in JavaScript is an object and these components include Functions, Strings, and Numbers. That means all the Functions, Strings, and Numbers can be called as objects in JavaScript.

**Your space:**

|  |
| --- |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |

# 

**TOPIC 9: BUILDING OBJECTS IN JAVASCRIPT**

**SUBTOPIC 1: CREATE NEW OBJECTS**

1. The key to creating objects is the **Object.create** function. This is like creating child objects from a parent object.
2. You can easily link the function to the name of the object from which you want to create your new object.

**Your space:**

|  |
| --- |
|  |
|  |
|  |
|  |
|  |
|  |

## 

**SUBTOPIC 2: SET PROPERTIES OF NEW OBJECTS**

To set the properties of the newly created object:

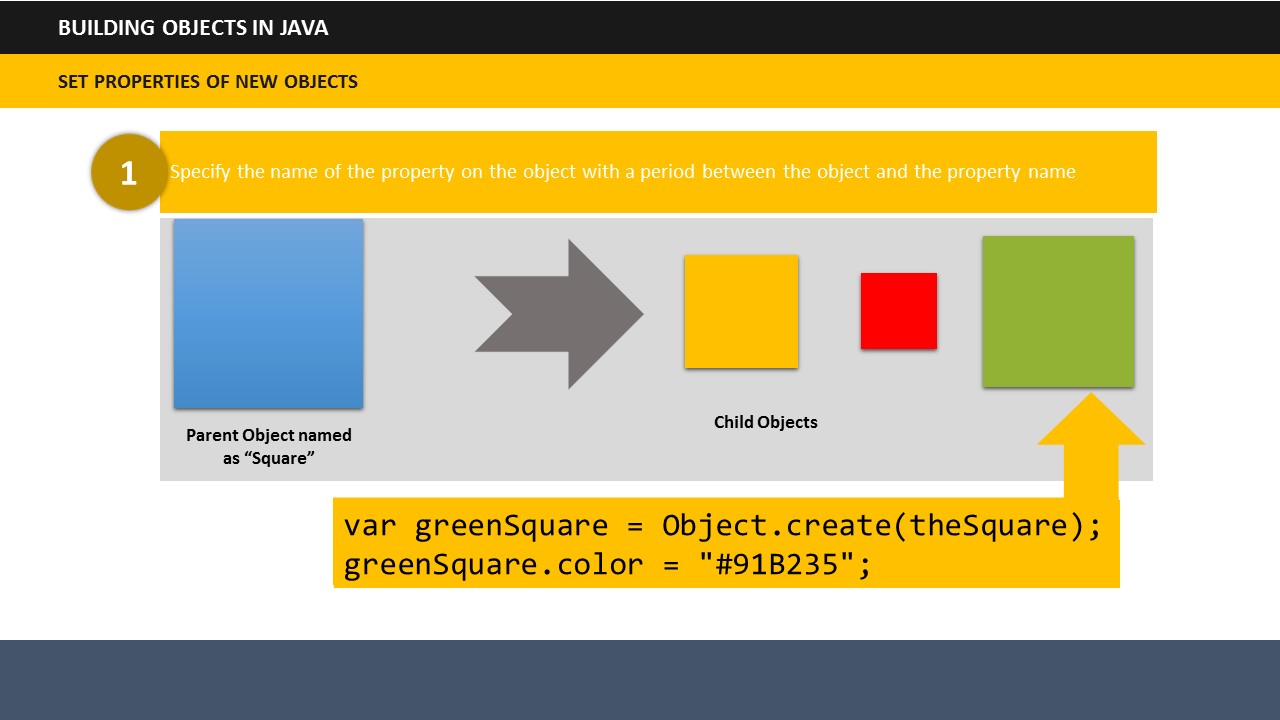
* Specify the name of the property on the object with a period between the object and the property name.

OR

* Set the property using bracket syntax.

**Your space:**

|  |
| --- |
|  |
|  |
|  |
|  |
|  |
|  |



**Example:** **Method 1**  
If you want to set the color property of the green square using the first method, the script will look like this:

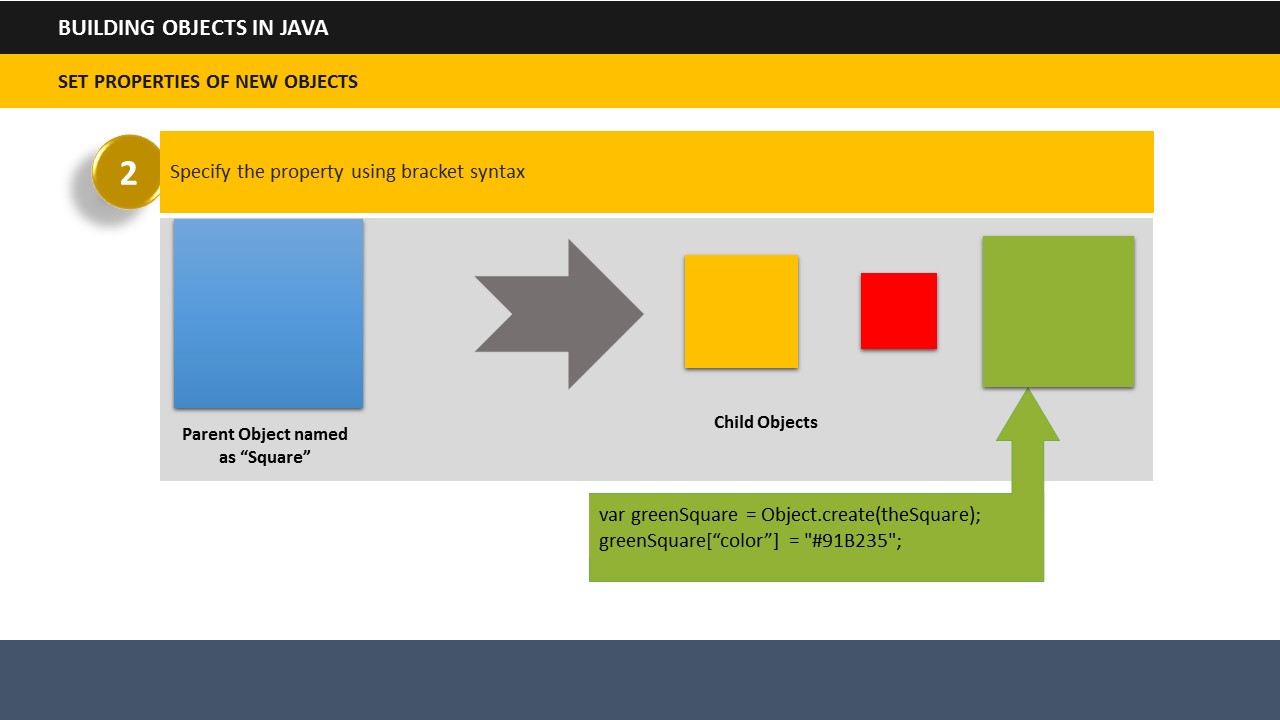
|  |
| --- |
| var greenSquare = Object.create(theSquare);  greenSquare.color = "#91B235"; |

1. Notice the color property on our green circle object is set to a value of #91B235.

**Your space:**

|  |
| --- |
|  |
|  |
|  |
|  |
|  |

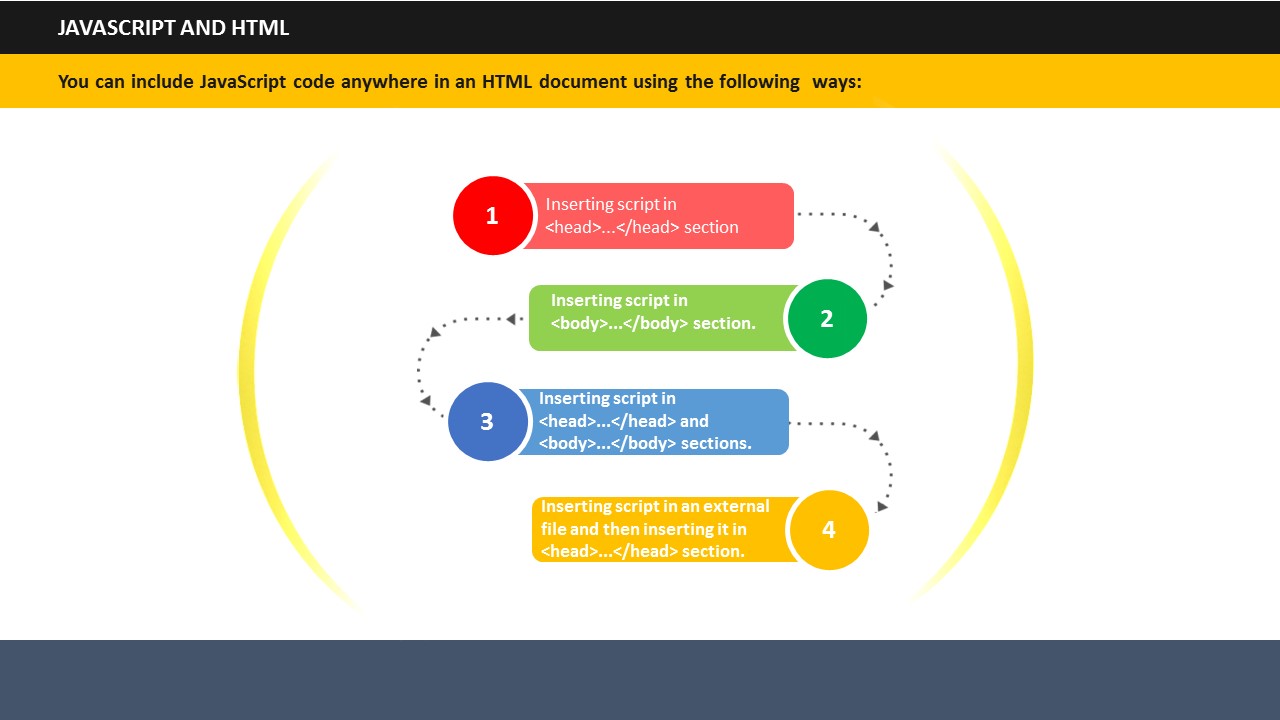
|  |
| --- |
|  |

**Example:** **Method 2**  
If you want to set the color property of the green square using the second of bracket syntax method, the script will look like this:

|  |
| --- |
| var greenSquare = Object.create(theSquare);  greenSquare[“color”] = "#91B235";color = "#91B235"; |

**Your space:**

|  |
| --- |
|  |
|  |
|  |
|  |
|  |



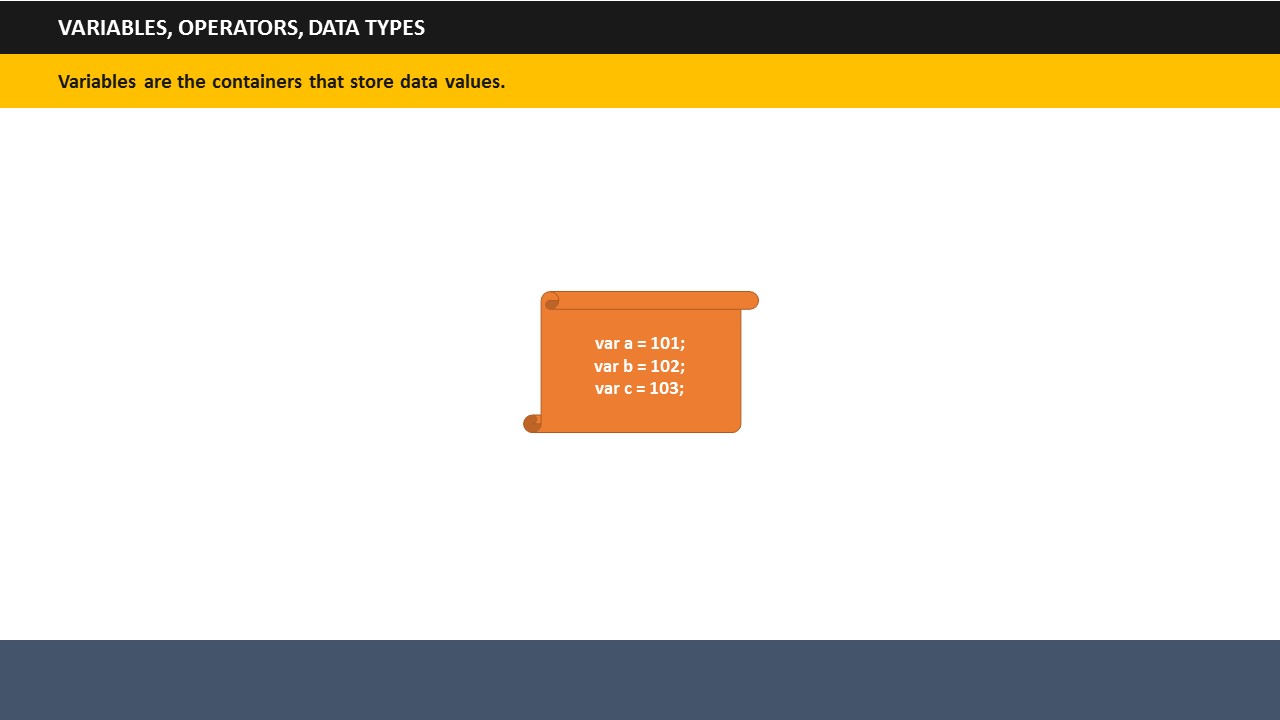
**TOPIC 10: JAVASCRIPT AND HTML**

1. JavaScript can change HTML attributes.
2. You can include JavaScript code anywhere in an HTML document using the following ways:

* Inserting script in <head>...</head> section.
* Inserting script in <body>...</body> section.
* Inserting script in <head>...</head> and <body>...</body> sections.
* Inserting script in an external file and then inserting it in <head>...</head> section.

**Your space:**

|  |
| --- |
|  |
|  |
|  |
|  |



**SUBTOPIC 1: VARIABLES**

**TOPIC 11: VARIABLES AND DATA TYPES**

1. Like many other programming languages, JavaScript uses variables, operators and data types.
2. Variables can be seen as containers that can store data values. You can place data into these containers and name these containers. Whenever, you wish to refer to the stored data, you can access the required container.
3. Examples of variables.

|  |
| --- |
| var a = 101;  var b = 102;  var c = 103; |

Here a, b, and c are the containers which are storing their corresponding values 101, 102, 103.

**Your space:**

|  |
| --- |
|  |
|  |
|  |
|  |
|  |

## 

## 

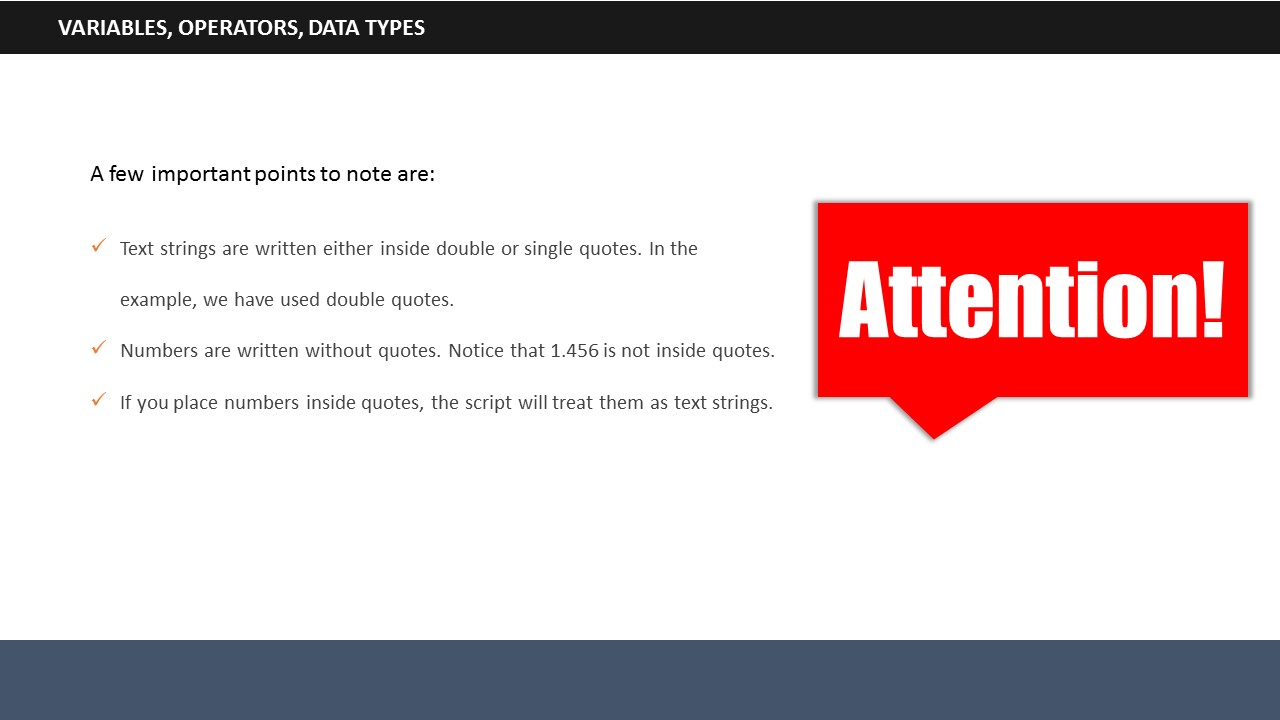
**SUBTOPIC 2: DATA TYPES**

1. Several types of data can be used in JavaScript, but this module focuses on two key types: Numbers and Text Strings.
2. Text string here refers to text values.
3. Example:

|  |
| --- |
| var x = 1.456;  var car = “Mercedes”;  var color = “Black”; |

**Your space:**

|  |
| --- |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |

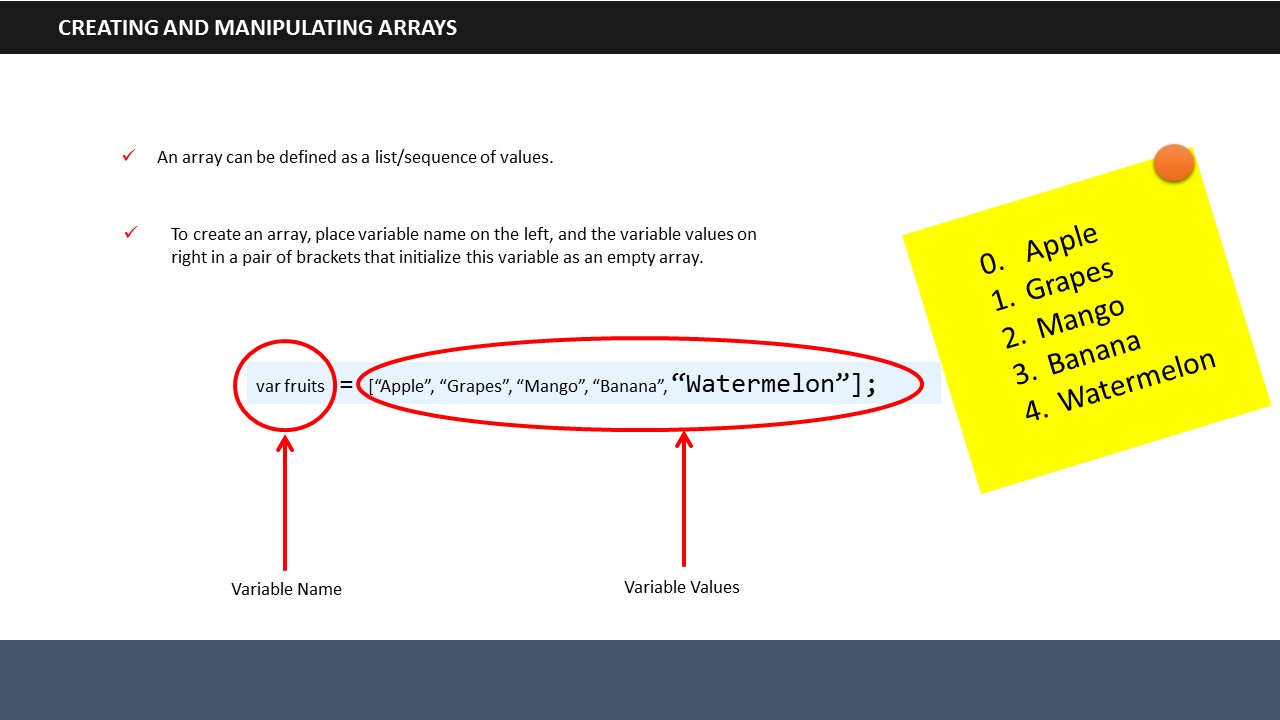


1. **Say:** A few important points to note are:

* Text strings are written either inside double or single quotes. In the example, we have used double quotes.
* Numbers are written without quotes. Notice that 1.456 is not inside quotes.
* If you place numbers inside quotes, the script will treat them as text strings.

**Your space:**

|  |
| --- |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |



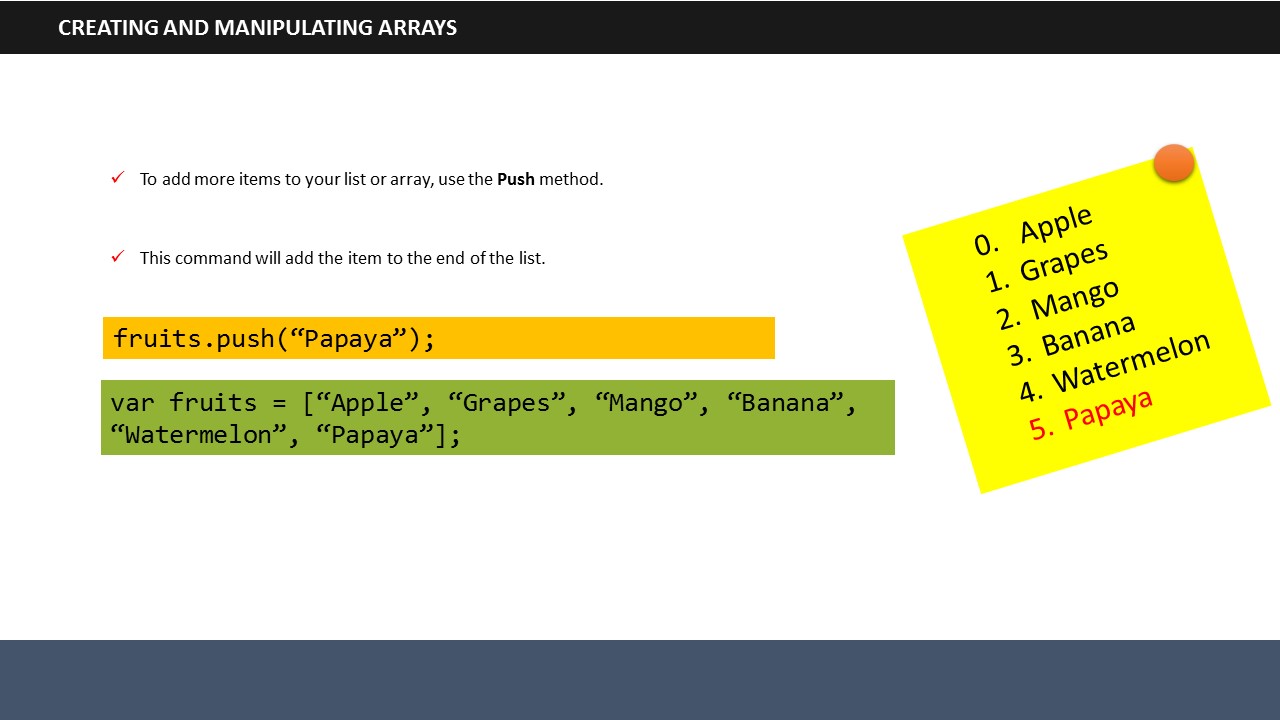
**TOPIC 12: CREATING AND MANIPULATING ARRAYS**

1. An array can be defined as a list/sequence of values. The value in array can be a number, object, function and so on. You can store any form of data in an array.
2. To create an array, place variable name on the left, and the variable values on right in a pair of brackets that initialize this variable as an empty array.
3. Example:

|  |
| --- |
| var fruits = [“Apple”, “Grapes”, “Mango”, “Banana”, “Watermelon”]; |

**Your space:**

|  |
| --- |
|  |
|  |
|  |
|  |
|  |

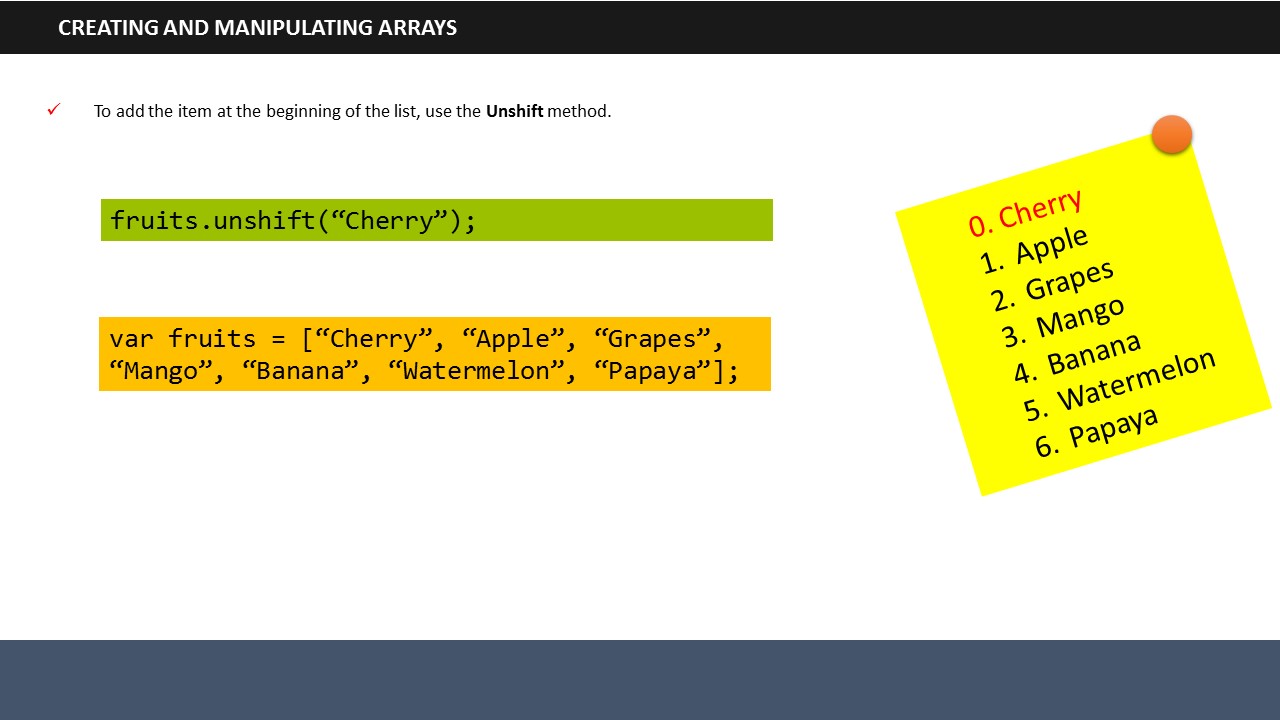


1. To add more items to your list or array, use the push method. This command will add the item to the end of the list.
2. For example, addition of Papaya to the fruit list.

|  |
| --- |
| fruits.push(“Papaya”); |

**Your space:**

|  |
| --- |
|  |
|  |
|  |
|  |
|  |

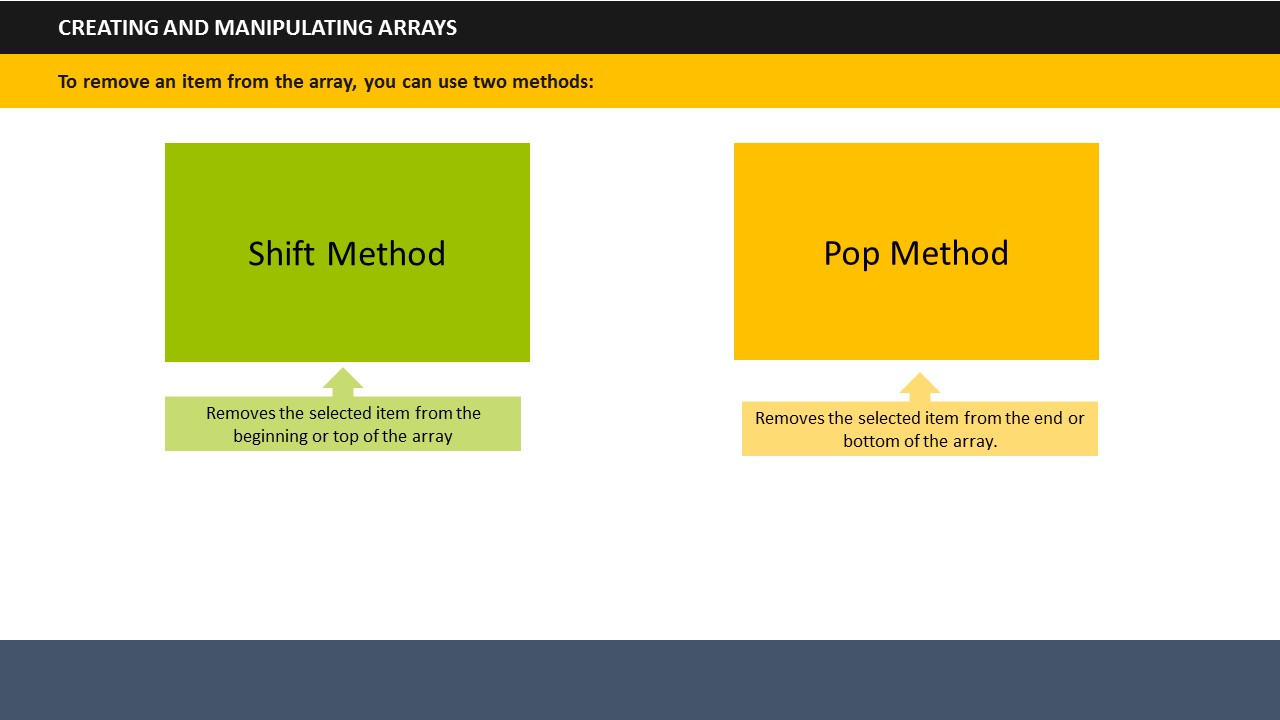


1. To add the item at the beginning of the list, use the unshift method.

|  |
| --- |
| fruits.unshift(“Cherry”); |

**Your space:**

|  |
| --- |
|  |
|  |
|  |
|  |
|  |



1. To remove an item from the array, you can use two methods: Shift and Pop.  
   * Shift Method: It removes the selected item from the beginning or top of the array.
   * Pop Method: It removes the selected item from the end or bottom of the array.

**Your space:**

|  |
| --- |
|  |
|  |
|  |

# 

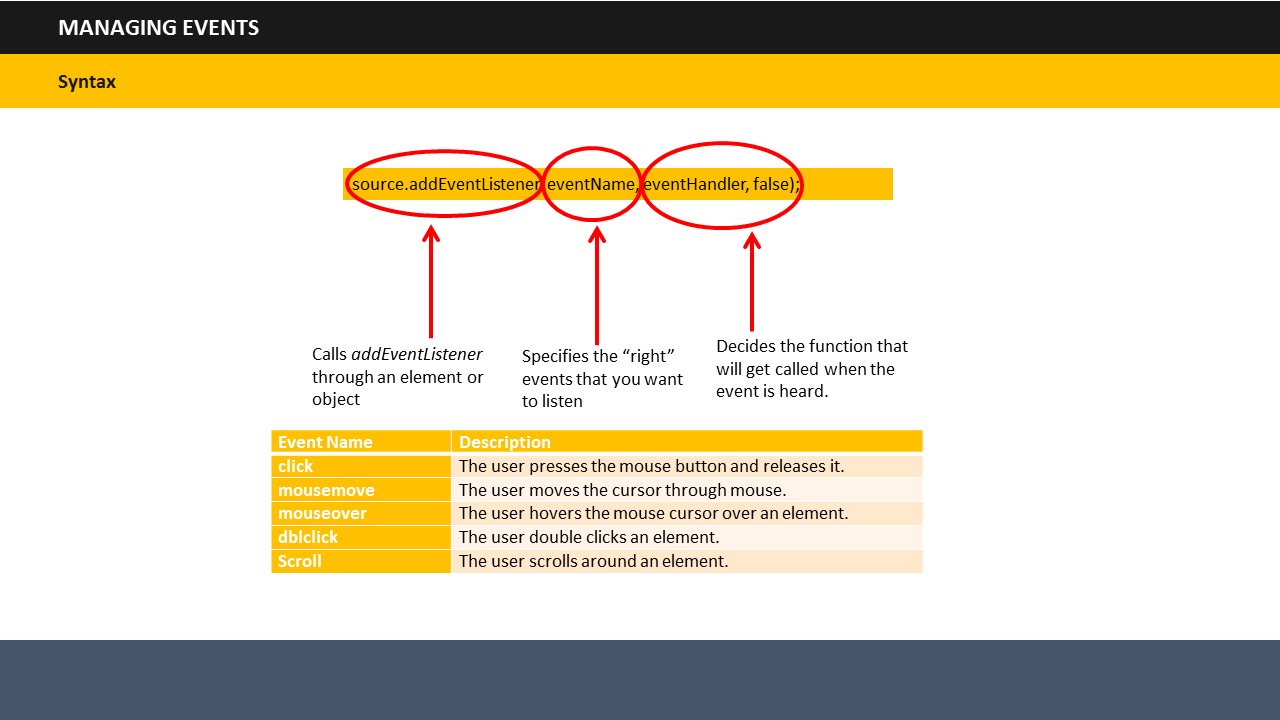
# 

**TOPIC 13: MANAGING EVENTS**

1. Events in JavaScript are the signals that happen due to some action.
2. Examples of an event could be pressing a keyboard button, clicking the mouse, loading of HTML and so on.
3. Events define what action should be taken when an action transpires.
4. To manage events, you need to listen for events and then react accordingly.
5. The function *addEventListener* is used to listen to the right event. This function immediately notifies what action should be taken if a right event takes place.

**Your space:**

|  |
| --- |
|  |
|  |
|  |



1. The syntax used for this is:

|  |
| --- |
| source.addEventListener(eventName, eventHandler, false); |

1. The Source helps to call *addEventListener* through an element or object.
2. The *Event Name* specifies the “right” events that you want to listen. A few examples are:

|  |  |
| --- | --- |
| **Event Name** | **Description** |
| click | The user presses the mouse button and releases it. |
| mousemove | The user moves the cursor through mouse. |
| mouseover | The user hovers the mouse cursor over an element. |
| dblclick | The user double clicks an element. |
| Scroll | The user scrolls around an element. |

1. The *Event Handler* decides the function that will get called when the event is heard.

**Your space:**

|  |
| --- |
|  |
|  |
|  |
|  |

# 

**TOPIC 14: SUMMARY**

**SUBTOPIC 1: OBJECTIVES MAPPING ACTIVITY**

# 

**Duration:** 10 Minutes

You can refer to the checklist given here to map the objectives outlined at the beginning of the session with the learnings from this session. This will help you analyze whether the learning objectives have actually been met or not.

* How to define JavaScript?
* What is the syntax of JavaScript?
* What are the attributes of script tags?
* What is programming?
* How to add a script in your page?
* How to start programming using JavaScript?
* What is Document Object Model?
* What are Events (Built in Objects)?
* Building objects in JavaScript
* JavaScript and HTML
* Data types, variables and operators
* JavaScript control structures
* Using loops
* Creating and manipulating arrays
* Using object hierarchy
* Managing events

**Your space:**

|  |
| --- |
|  |
|  |
|  |

**SUBTOPIC 2: SUMMARIZATION ACTIVITY**



**Duration:** 10 Minutes

It’s time to review what we have learned so far. Write in the space given in this guide, what you have learned today.

**Your space:**

|  |
| --- |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
| |  | | --- | |  | |  | |  | |  | |  | |  | |  | |  | |  | |  | |  |  |  | | --- | |  | |  | |  | |  | |  | |  | |

## 

## 

**SUBTOPIC 3: FINAL TIPS AND SUMMARIZATION**

**Your space:**

|  |
| --- |
|  |
|  |
|  |
|  |
|  |
|  |



**ITES Functional Skills Training**

# LAB EXERCISES

**TOPIC 15: LAB EXERCISES**

**Exercise 1: Write a program to call a function which performs a calculation, and returns the result.**

**Your space:**

|  |
| --- |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |

**Exercise 2: Write a program to add a string and two numbers.**

**Your space:**

|  |
| --- |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |

**Exercise 3 Write a program to join all elements of an array into a string.**

**Your space:**

|  |
| --- |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |

**Exercise 4: Write a program to loop through the elements of an object.**

**Your space:**

|  |
| --- |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |

**Exercise 5: Write a program to change the text of the selected option.**

**Your space:**

|  |
| --- |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |

**Exercise 6: Write a program to find the name of the first form in a document.**

**Your space:**

|  |
| --- |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |

**Exercise 7: Write a program to change the source of the light bulb.**

**Your space:**

|  |
| --- |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |

**Exercise 8: Write a program to explain onmouseover and onmouseout event.**

**Your space:**

|  |
| --- |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |