



|  |  |  |  |
| --- | --- | --- | --- |
| Version | Date | Author | Change Status |
| 1.0 | 01-07-2015 | Shweta |  |

**TABLE OF CONTENT**

* ABOUT THIS GUIDE 5
* LEGEND 6
* THE DOCUMENTS – NAME & DESCRIPTION 9
* SESSION SCHEDULE 10
* TOPIC 1 – GETTING STARTED 12
* SUBTOPIC 1: ACTIVITY 12
* SUBTOPIC 2: IN THIS SESSION – SESSION OBJECTIVES 15
* SUBTOPIC 3: ASSUMPTIONS 16
* SUBTOPIC 4: REQUIREMENTS 17
* SUBTOPIC 5: FREQUENTLY ASKED QUESTIONS (FAQs) 19
* TOPIC 2 – INTRODUCTION TO JAVASCRIPT 20
* SUBTOPIC 1: WHAT IS JAVASCRIPT? 20
* SUBTOPIC 2: JAVASCRIPT SYNTAX 21
* SUBTOPIC 3: ATTRIBUTES OF SCRIPT TAGS 22
* TOPIC 3 – WHAT IS PROGRAMING ALL ABOUT? 24
* TOPIC 4: HOW TO ADD A SCRIPT IN YOUR PAGE? 25
* SUBTOPIC 1: WHERE TO ADD THE JAVASCRIPT CODES? 25
* SUBTOPIC 2: EXAMPLE 26
* TOPIC 5: STARTING WITH PROGRAMMING 29
* SUBTOPIC 1: VARIABLES 29
* SUBTOPIC 2: OPERATORS 30
* SUBTOPIC 3: FUNCTIONS 33
* SUBTOPIC 4: CONDITIONAL STATEMENTS 35
* SUBTOPIC 5: LOOPING 38
* TOPIC 6: DOCUMENT OBJECTMODEL 40
* SUBTOPIC 1: USING DOT NOTATION 40
* SUBTOPIC 2: DOCUMENT OBJECT 41
* SUBTOPIC 3: FORM COLLECTION 42
* SUBTOPIC 4: FORM ELEMENTS 43
* SUBTOPIC 5: IMAGES COLLECTION 46
* SUBTOPIC 6: DIFFERENT TYPES OF OBJECTS 48
* TOPIC 7 – OBJECT ORIENTED PROGRAMMING 49
* SUBTOPIC 1: IN THIS SESSION – SESSION OBJECTIVES 49
* TOPIC 8 – OBJECTS-BASED PROGRAMMING CONCEPTS 51
* SUBTOPIC 1: WHAT ARE OBJECTS? 51
* TOPIC 9: BUILDING OBJECTS IN JAVASCRIPT 53
* SUBTOPIC 1: CREATE NEW OBJECTS 53
* SUBTOPIC 2: SET PROPERTIES OF NEW OBJECTS 54
* TOPIC 10: JAVASCRIPT AND HTML 57
* TOPIC 11: VARIABLES & DATA TYPES 58
* SUBTOPIC 1: VARIABLES 58
* SUBTOPIC 2: DATA TYPES 59
* TOPIC 12: CREATING AND MANIPULATING ARRAYS 61
* TOPIC 13: MANAGING EVENTS 64
* TOPIC 14: SUMMARY 66
* SUBTOPIC 1: RECAP 66
* SUBTOPIC 2: FINAL TIPS AND SUMMARIZATION 71
* TOPIC 15: CHECK YOUR UNDERSTANDING: 72
* TOPIC 16: LAB EXERCISES 76

# This Facilitator’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**

There should be 25 - 50 participants in this program.

**Please note**: Some of the group activities may be better facilitated with an even number of participants.

This Facilitator’s Guide will help you give the participants 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. You can discuss the points as mentioned in this guide when displaying the various screens in the course.
* It includes activities that you can conduct in the class and instructions to carry out the activities.
* It includes a ‘Check Your Understanding’ section which the facilitator can conduct as a quick quiz to test the learner’s understanding of the concepts that are 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 the participants might 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 when explaining a 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 |
|  | Flipchart |
|  | Group Activity |
|  | Activity |
|  | Student’s Guide |
|  | Checklist |
|  | Tips for the Facilitator |
|  | Important |

**CHECKLIST**

* Reach the training venue at least half an hour before the training commences.
* Check that the electrical connections in the room are working.
* Check the temperature of the room and ensure that it is comfortable.
* Keep in place all the required items - white board markers, duster, pens, flip chart sheets, flip chart stand, printouts, etc.
* Check that the computer you are going to display the presentation on is working.
* Check the projector and keep it ready for use.
* Carry the ILT version of the course on your computer, and have it ready for use.
* Ensure that you have gone through the entire course and mapped it to this guide. Be sure of what you need to do on each screen of the course.
* Ensure that all reference documents are kept in a single folder, along with the ILT version of the course for you to easily access during the training session.
* Take print outs of the guides/worksheets for the students and carry the required number of Student’s Guides to the room.
* Print the Facilitator’s Guide document.
* Keep a list of the names of the participants as a reference for yourself to help you identify them.



**PLEASE NOTE:** It is vital that you go through both the course and this guide thoroughly before you conduct your first session. This will increase your confidence while delivering the training.



* **Review** the instructions and activities listed in this Facilitator’s Guide.
* **Ensure** that you have all the teaching aids required for the session. The required teaching aids are:
  + - ILT package for the course
    - PC and LCD Projector machine and screen
    - Copies of the ***Participant’s Guide*** (one for each participant and a copy for yourself)
    - ***Activity*** and ***Response Sheets*** (cut-outs wherever required)
    - Whiteboard or flipcharts
    - Stationery items like pens, pencils, whiteboard markers, etc.
* **Review** the ILT version of the course to familiarize yourself with the information provided on each screen. Each screen is referenced in this guide. This will enable you to refer to the relevant screen when explaining it to the participants.



* “**Notes**” sections have been provided at various points in this guide. Note down points that you think are important and need to be further explained in the session.
* **Check** that the computer on which you are going to display the ILT is working and connected to the LCD projector.

# 

THE DOCUMENTS – NAME & DESCRIPTION

* **ITES\_WebDesign\_JavScript\_GettingStarted\_.pptx**: This is the ILT version of the course that should be used for the session.
* **ITES\_WebDesign\_JavScript\_GettingStarted\_Participant'sGuide.docx:** This is the Participant’s Guide for the session.
* [**Please note**: Distribute this document after discussing the Program Schedule (Keep a copy of the schedule on the Flipchart for reference)].

# 

**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 |

Present the program schedule for the session. Begin the session with a discussion on the detailed structure of the program and the topics to be covered here. (You can copy the schedule table into a PowerPoint presentation to project the schedule on the screen.)

****

* **NOTE**: Due to the nature of the training, we have not marked the breaks in the schedule. Thus, the total time for the session is approximately 3 hours.   
  Please distribute these breaks as per your discretion.



* Distribute the following to the participants:
* ITES\_WebDesign\_JavaScript\_GettingStarted\_Participant'sGuide.docx (Printouts)
* Pens
* Notepads

# 

# Project the ILT course.

**TOPIC 1 – GETTING STARTED**

**TOPIC**

**SUBTOPIC 1: ACTIVITY**



* Play the intro animation.Welcome the participants to the ILT Session on ***ITES*** *-* ***Basic* *Functional Skills – CSS*** by saying “Welcome to the course on Web Design. Today, we will discuss about the JavaScript and the basic concepts associated with it.”



* Introduce yourself to the participants. Share your name, and a brief professional profile, including: designation, tenure in the organization, domain of work, etc. (especially if most of the participants are unknown to you.)
* Now, have the participants take part in the activity described below.
* **NOTE**: You can either use your own idea/concept to do this, or use one of these activities. We are assuming that the participants by now know each other, and therefore, there’s no need for any introductions, etc.

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



This activity has two parts. Due to the limited time of time session, ***only Part 1 will be* *performed in this session***. Part 2 will be performed in the next session.

Give this situation to the participants: 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:** Ask each participant if given a choice in this situation to carry one object, what would that he/she carries and why? Encourage everyone to be creative. Their answers can range from the basic food and water to dog and guitar!

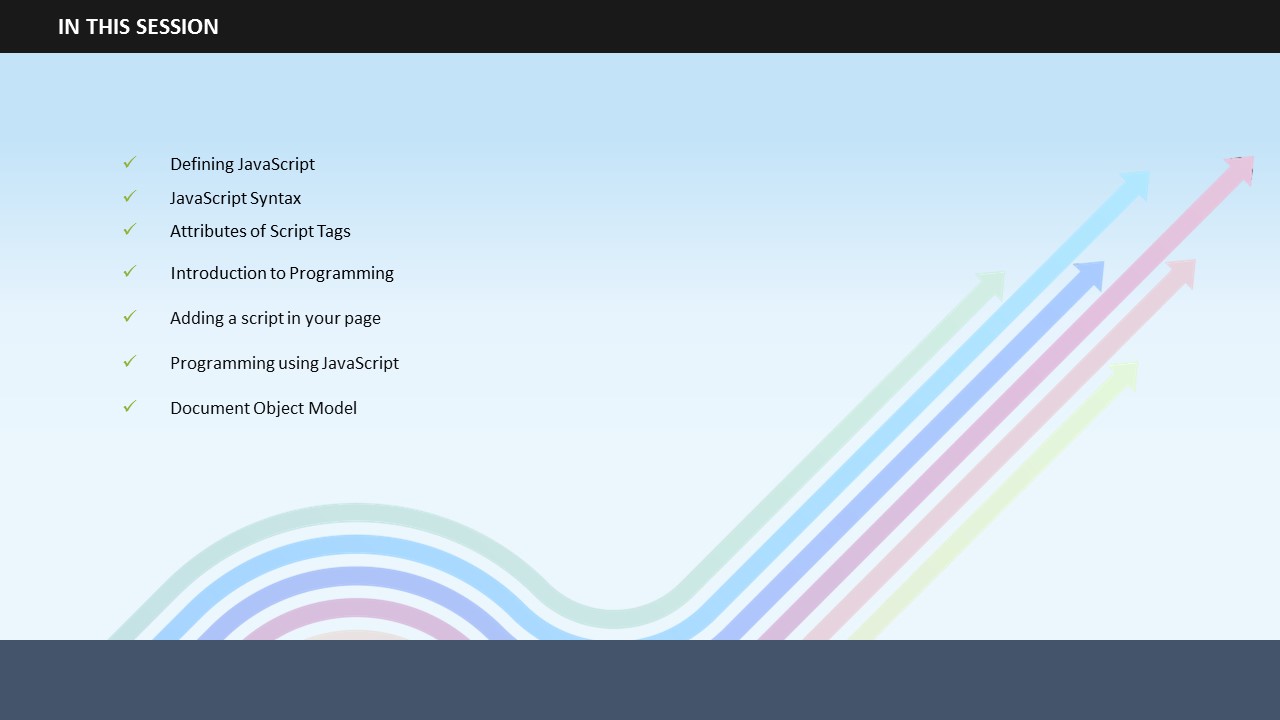
**Part 2:** Once everyone has chosen an object, divide the class in small groups and ask each group to work together to improve their chances of survival on the island by combining their objects together. If necessary, each group can add more objects, but they should use the objects already chosen by each group member.

The group that comes up with the most creative way of survival using these objects wins.

****

**NOTE**:

This activity is meant to get participants “hooked” onto the session from the very start. Thus, you must ensure that each participant is actively joining in, and ***having******fun*** while doing it.

 **Say:** Let us begin by outlining the main learnings of this session. This will help you understand what tasks you will be able to perform after completing this session.

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

1. **Say**: In this session, you will learn…
2. **Say**: … how to define JavaScript…
3. **Say**: … what is the syntax of JavaScript?
4. **Say**: You will also learn about the attributes of script tags.
5. **Say:** What is programming?
6. **Say:** How to add a script in your page?
7. **Say:** How to start programming using JavaScript?
8. **Say:** What is Document Object Model?

**Do:**

* Invite participant’s inputs on the objectives – asking them to state their own objectives/problem areas. In case such objectives/problem areas are relevant, write them on the flipchart. Remember to cover such points at the appropriate time in the training session. Also, remember to discuss them again during summarization.
* You can refer to these objectives during the summarization session to know whether the objectives have actually been met or not.

## 

**SUBTOPIC 3: ASSUMPTIONS**

MPTIONS

**Say:** At the beginning of this session, I am assuming that you have a working knowledge of HTML and CSS.

**Do:** Tell the assumption to the participants.

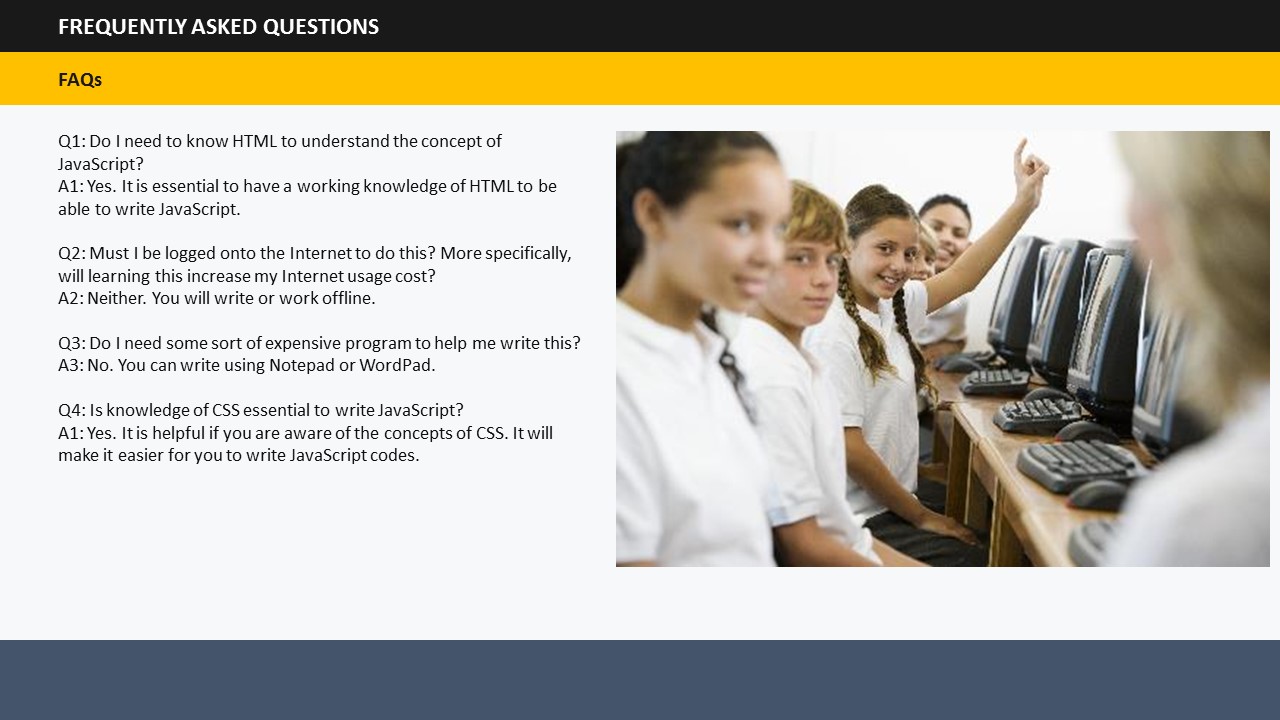
## 

**SUBTOPIC 4: REQUIREMENTS**

**Say**: Before learning about JavaScript, it is extremely important that you know what tools are basically required for JavaScript writing. To continue with your understanding of JavaScript, you will obviously need a personal computer.

**Say**: Along with a personal computer, you will also need a Web browser, such as Google Chrome, Mozilla Firefox, Internet Explorer, etc.

**Say**: Now that you have a personal computer and a Web browser, the only thing needed now is a word processor. If you have access to Windows "Notepad" or "WordPad" programs or the MAC "Simple Text" program, you can use that to get started. Fortunately, JavaScript is written in plain text, which means that you don’t need any fancy software programs to create your JavaScript files.



Say: I am sure you must have a lot of questions on your mind right now. Let me try and address some of them.

Q1: Do I need to know HTML to understand the concept of JavaScript?  
A1: Yes. It is essential to have a working knowledge of HTML to be able to write JavaScript.

Q2: Must I be logged onto the Internet to do this? More specifically, will learning this increase my Internet usage cost?  
A2: Neither. You will write or work offline.

Q3: Do I need some sort of expensive program to help me write this?  
A3: No. You can write using Notepad or WordPad.

Q4: Is knowledge of CSS essential to write JavaScript?  
A1: Yes. It is helpful if you are aware of the concepts of CSS. It will make it easier for you to write JavaScript codes.



1. **Say**: JavaScript is the most popular programming language in the world.

**Do:** Before telling what JavaScript is, encourage participants to share their understanding of JavaScript.

1. **Say:** 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.

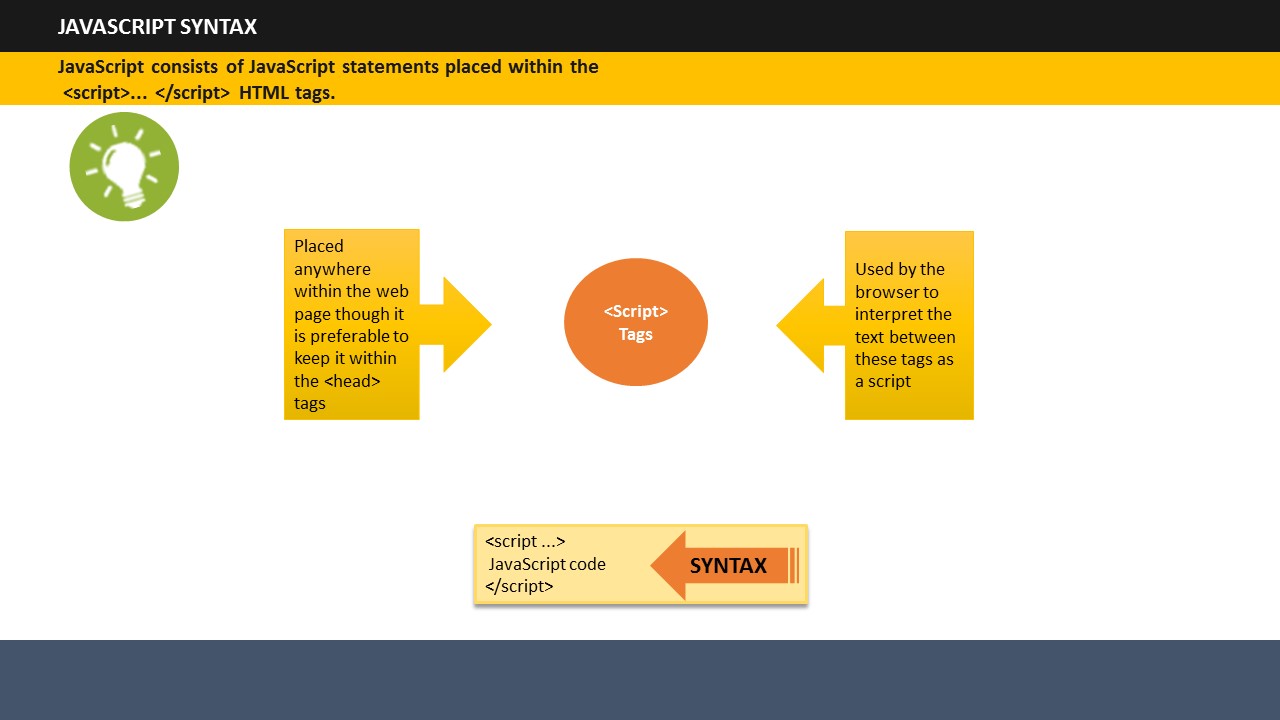
**Do:** Bring students attention to the point that JavaScript and Java are completely different languages, both in concept and design.

**SUBTOPIC 5: FREQUENTLY ASKED QUESTIONS (FAQs)**

**TOPIC 2 – INTRODUCTION TO JAVASCRIPT**

**SUBTOPIC 1: WHAT IS JAVASCRIPT?**

****



**SUBTOPIC 2: JAVASCRIPT SYNTAX**

1. **Say:** JavaScript consists of JavaScript statements placed within the <script>... </script> HTML tags.
2. **Say:** 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. **Say:** The syntax of JavaScript looks like this:

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

**Do:** Show this code on the screen.

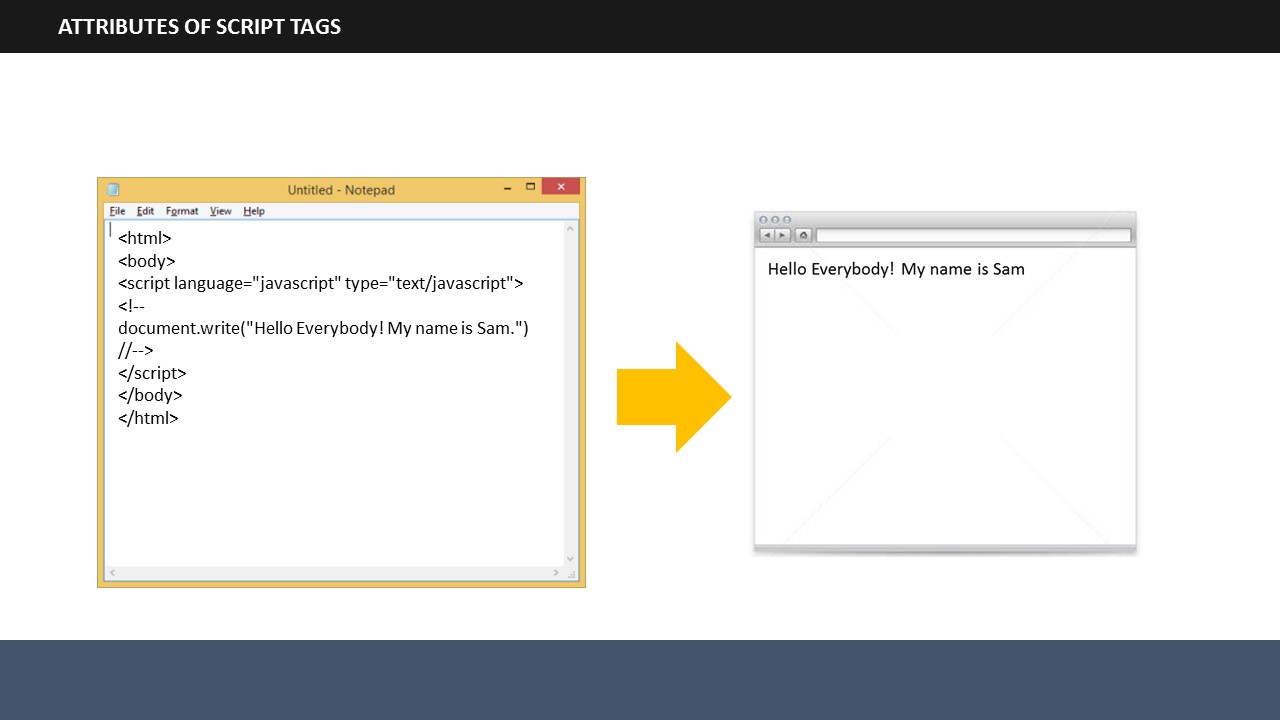
## 

**SUBTOPIC 3: ATTRIBUTES OF SCRIPT TAGS**

1. **Say:** 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. **Say:** Using these two attributes the JavaScript segment will appear like this:

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

**Do:** Show the codes on the screen.

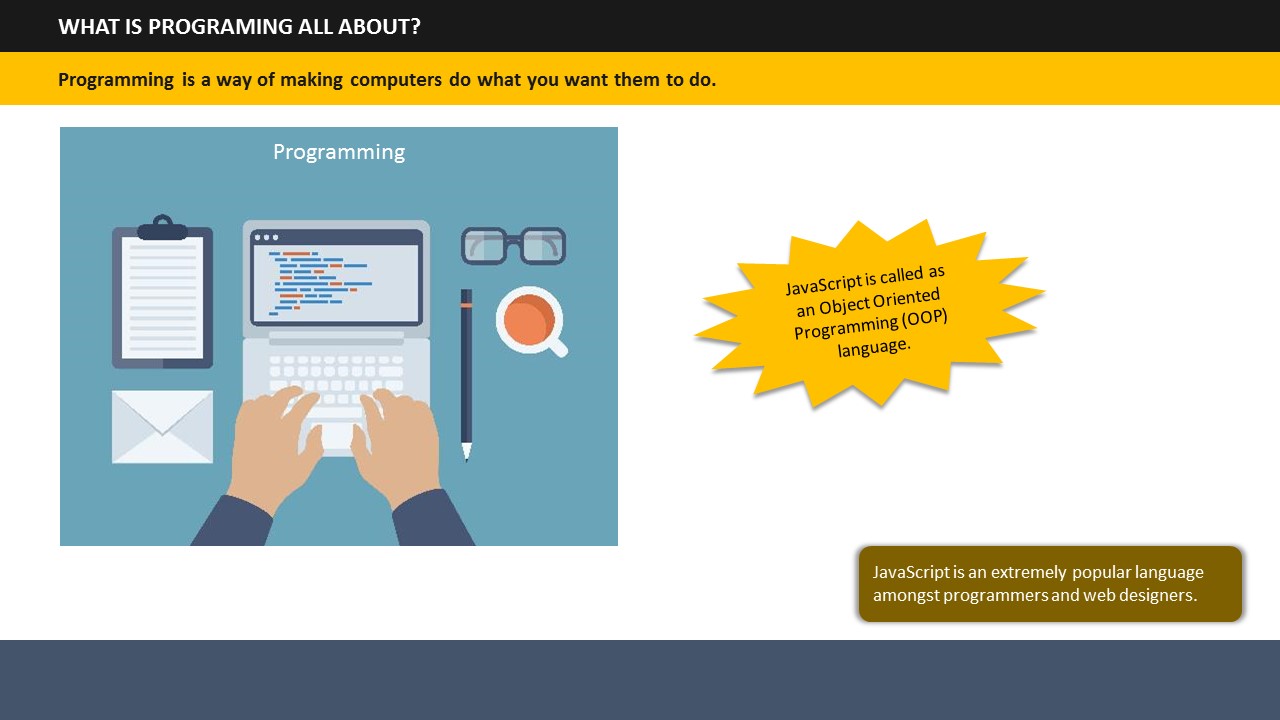


1. you want to print some text, for example “Hello Everybody! My name is Sam!” on your webpage, the script would look like this:

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

**Do:** Invite participants to try guessing how the script would look like. You can ask them to come and write it on the flipchart for the class. Tell them to try it even if they fear that they may not give the correct answer. Appreciate their efforts irrespective of the correct/incorrect answer given by them.   
  
**Do:** Next, show the codes on the screen.

1. So, on writing the codes as discussed just now, the following result will show on your webpage.

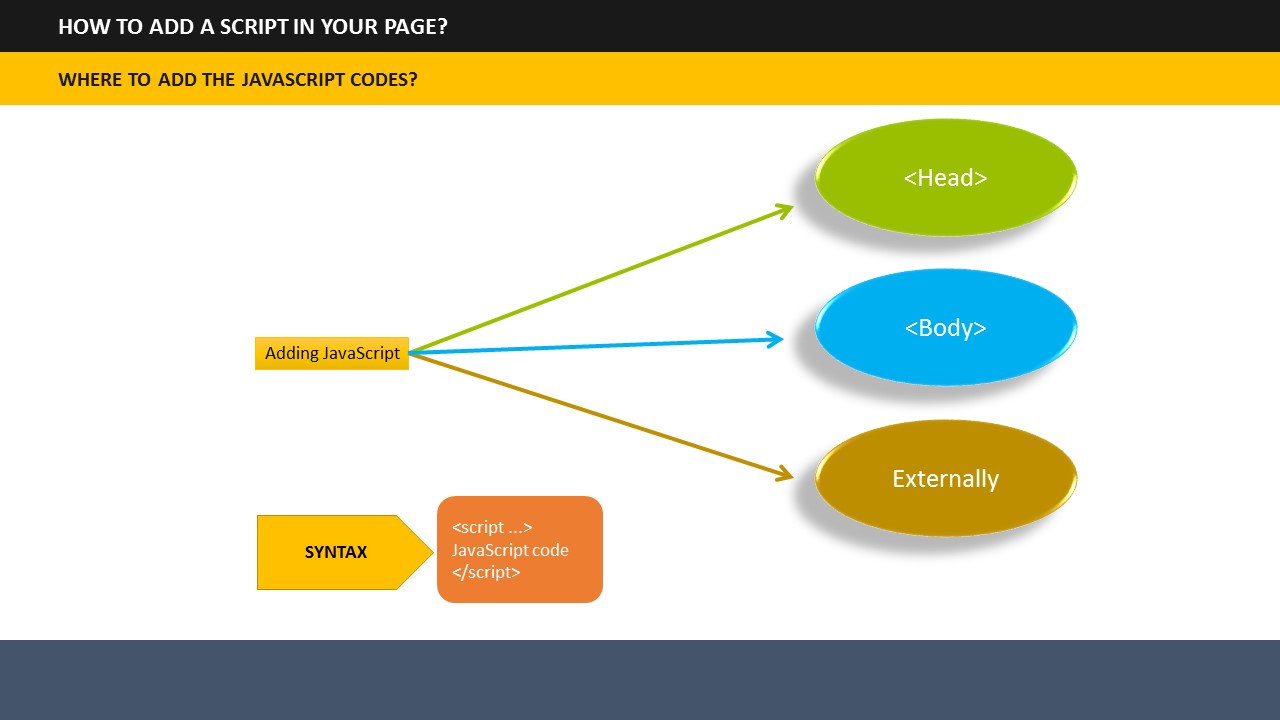


|  |
| --- |
| Hello Everybody! My name is Sam |

**Do:** Show the resultant text on the screen.

**Say**: 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.   
  
**Do:** Ask participants to share their thoughts on programming and what it’s about?  
**Say**: You have already learned about HTML and CSS which are also used to do programming.   
**Say**: Another language that is used to do programming is JavaScript. It is extremely popular amongst programmers and web designers.   
**Say**: Do you know that JavaScript is also called as an Object Oriented Programming or OOP language. This is because JavaScript involves working with objects. **Say**: JavaScript can change HTML content, its attributes and style. It can also be used to validate data inputs

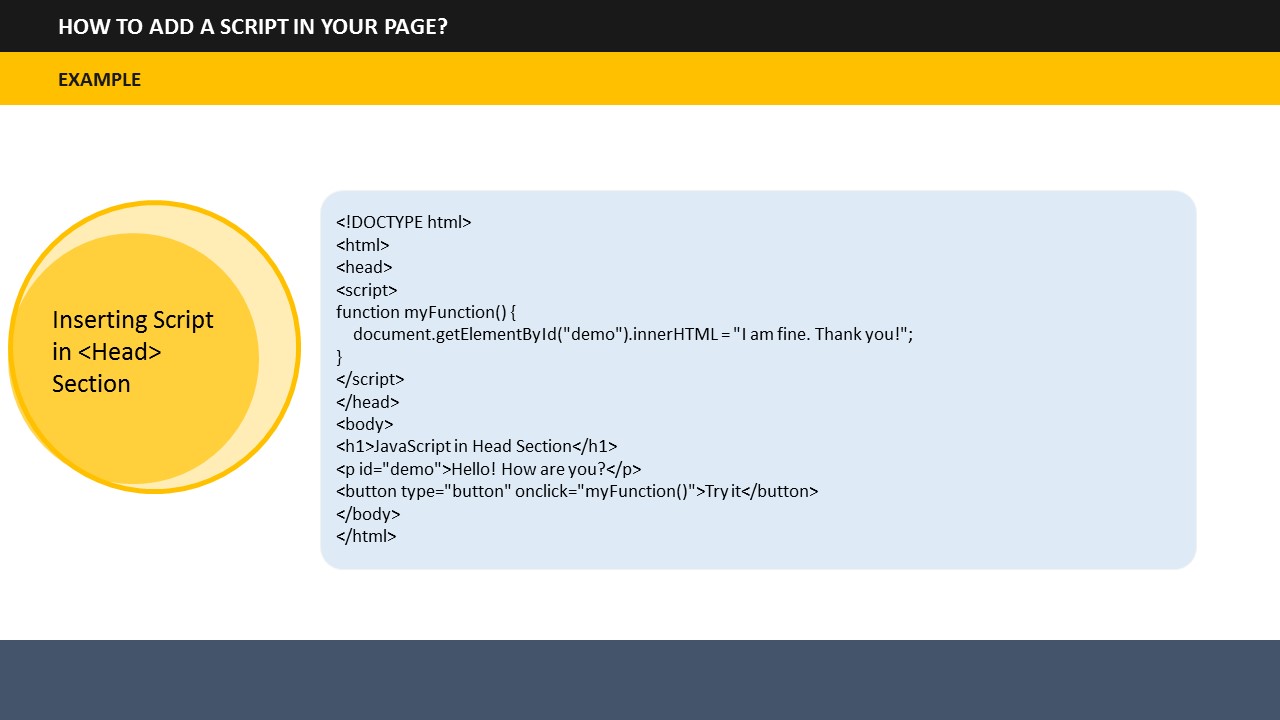
**TOPIC 3 – WHAT IS PROGRAMING ALL ABOUT?**



**SUBTOPIC 1: WHERE TO ADD THE JAVASCRIPT CODES?**

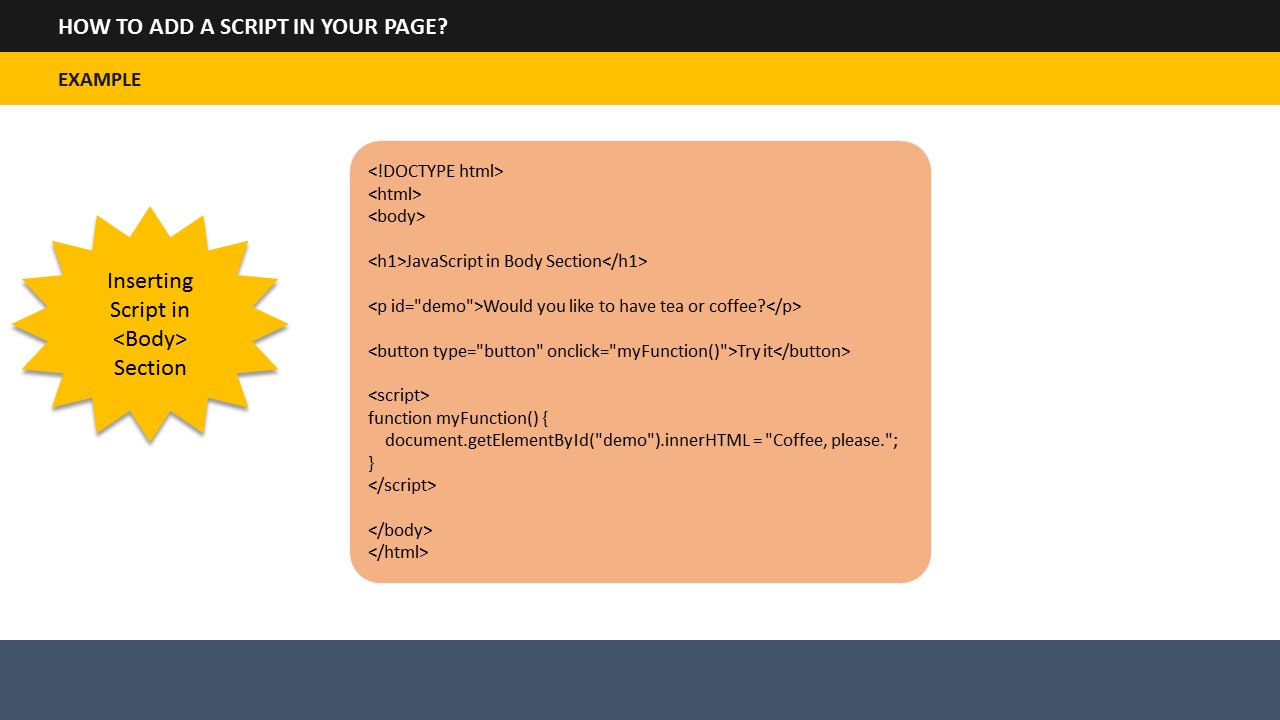
**TOPIC 4: HOW TO ADD A SCRIPT IN YOUR PAGE?**

1. **Say**: Now, let’s look at an example that shows how you can add JavaScript codes to HTML pages.
2. **Say**: The example show on the screen demonstrates how JavaScript can be placed in the <head> section of an HTML page.  
     
   **Do:** Show the example on the screen and highlight the <head> section.

**Say**: The example code inserted into the Head section will give us an output as shown on the screen.

**Say**: If you click the Try It button, the function is invoked in this script and it will give a result.  
  
  
  
**Do:** Highlight the following result shown in the example: “I am fine. Thank you!”

.



***Inserting Script in <Body> Section***

1. **Say**: The example code inserted into the <body> section will give us an output as shown on the screen.
2. **Say**: Just like the <head> example, this code will also give result on clicking the Try It button.  
     
   **Do:** Highlight the following result shown in the example: “Coffee, please”

## 

**TOPIC 5: STARTING WITH PROGRAMMING**

**SUBTOPIC 1: VARIABLES**

1. **Say**: Like many other programming languages, JavaScript uses variables, operators and data types. Let’s go through each of them.
2. **Say**: 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.



1. **Say**: Let’s see examples of variables.

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

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

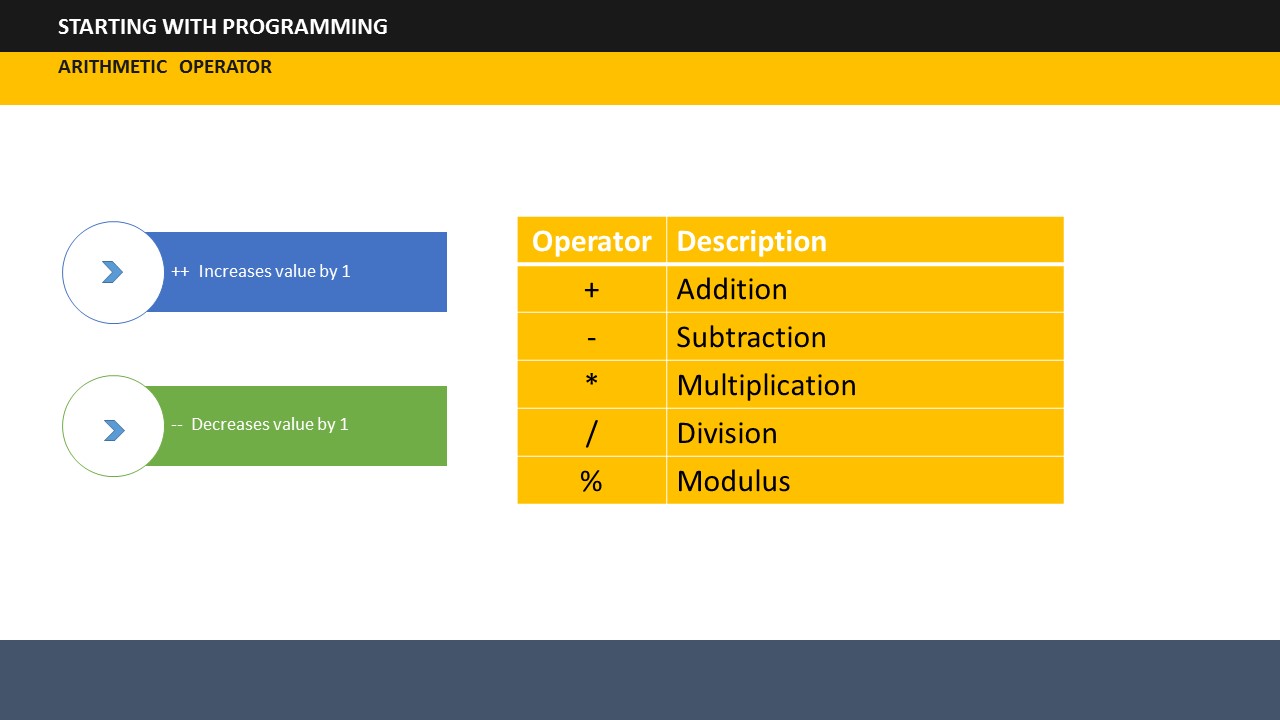
## 

**SUBTOPIC 2: OPERATORS**

1. **Say**: JavaScript uses the following types of operators:

* Arithmetic Operators
* Assignment Operators
* Comparison Operators
* Logical Operators

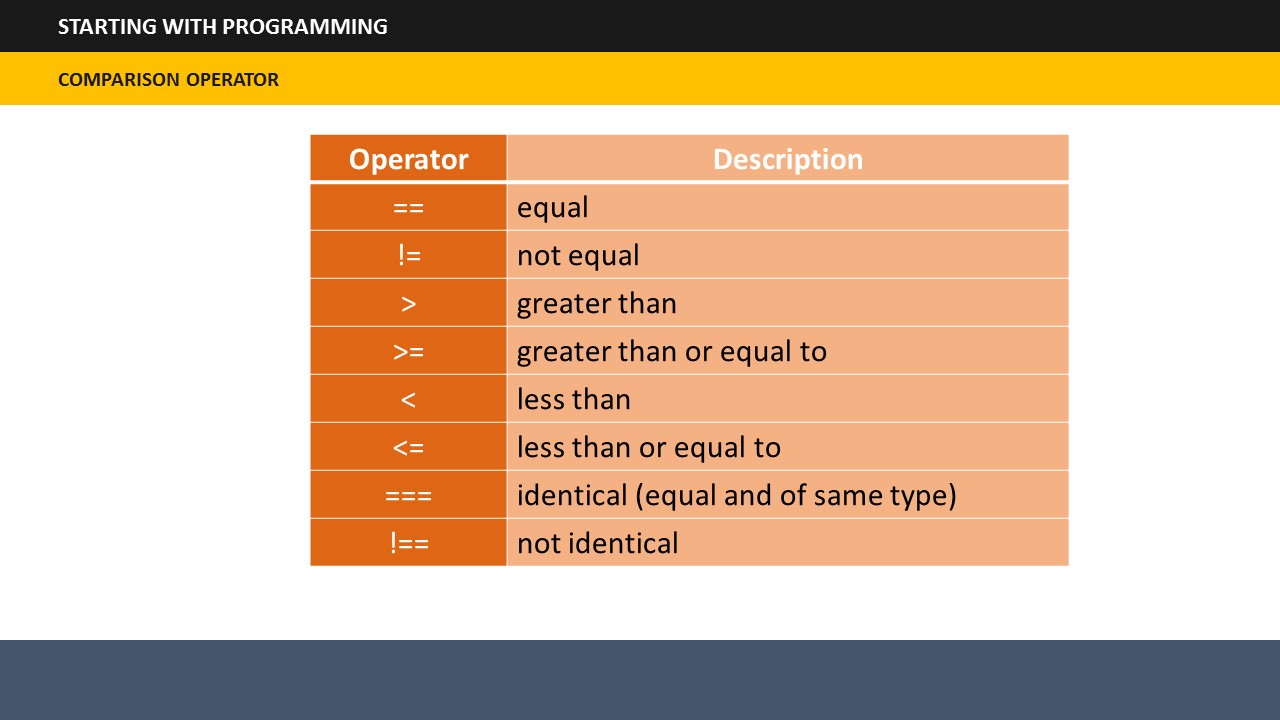
1. **Say**: Let’s see details and examples of each operator.



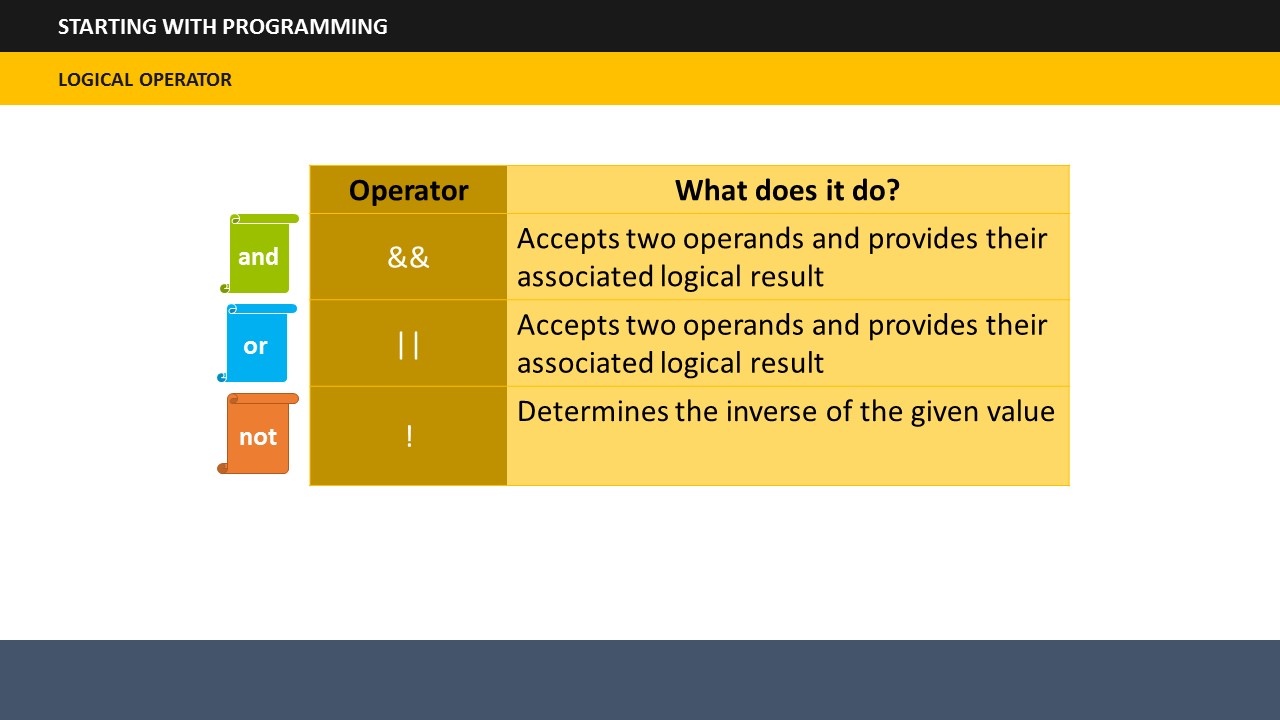
1. **Say**: The arithmetic operators function as the addition, subtraction, multiplication, division, and modulus operators, and operate very similarly to other languages.
2. **Say**: In addition to these basic arithmetic operators, there are two more operators which increase or decrease the value of variable by 1.



1. **Say**: The assignment operators assign a value to a variable.



1. **Say**: The comparison operators determine if the two operands or numbers meet the given condition

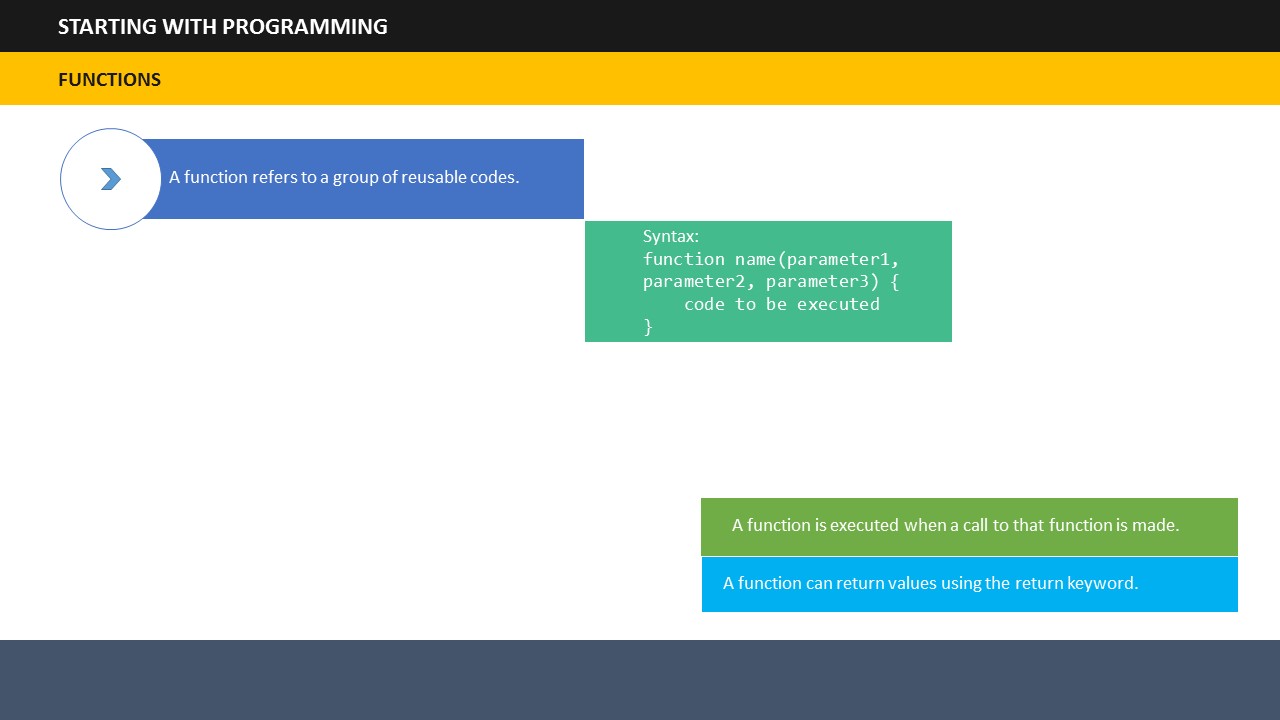


1. **Say**: The logical operators are *and*, *or*, and *not*. The and operator along with or operator accepts two operands or numbers and provides their associated logical result, while the third operator determines the inverse of the given value, and returns the boolean: true values become false, or false values become true.

## 

**SUBTOPIC 3: FUNCTIONS**

1. **Say**: 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. **Say**: You can write a function using the syntax shown on the screen.
3. **Say**: 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.
4. **Say**: Another point to note is that functions can return values using the return keyword.



1. **Say**: Let’s take a look at an example to understand the concepts of functions better.
2. **Say**: The name of the function in this example is say hello. And it is followed by an opening and closing parenthesis. The say hello at the bottom acts as the alert for function call. If you will type this example code and preview your page in a browser, you will see hello displayed.
3. **Say**: We have another example on the screen that displays the return value. Here instead of displaying an alert, the return value will show the calculated distance on entering speed and time values.

## 

**SUBTOPIC 4: CONDITIONAL STATEMENTS**

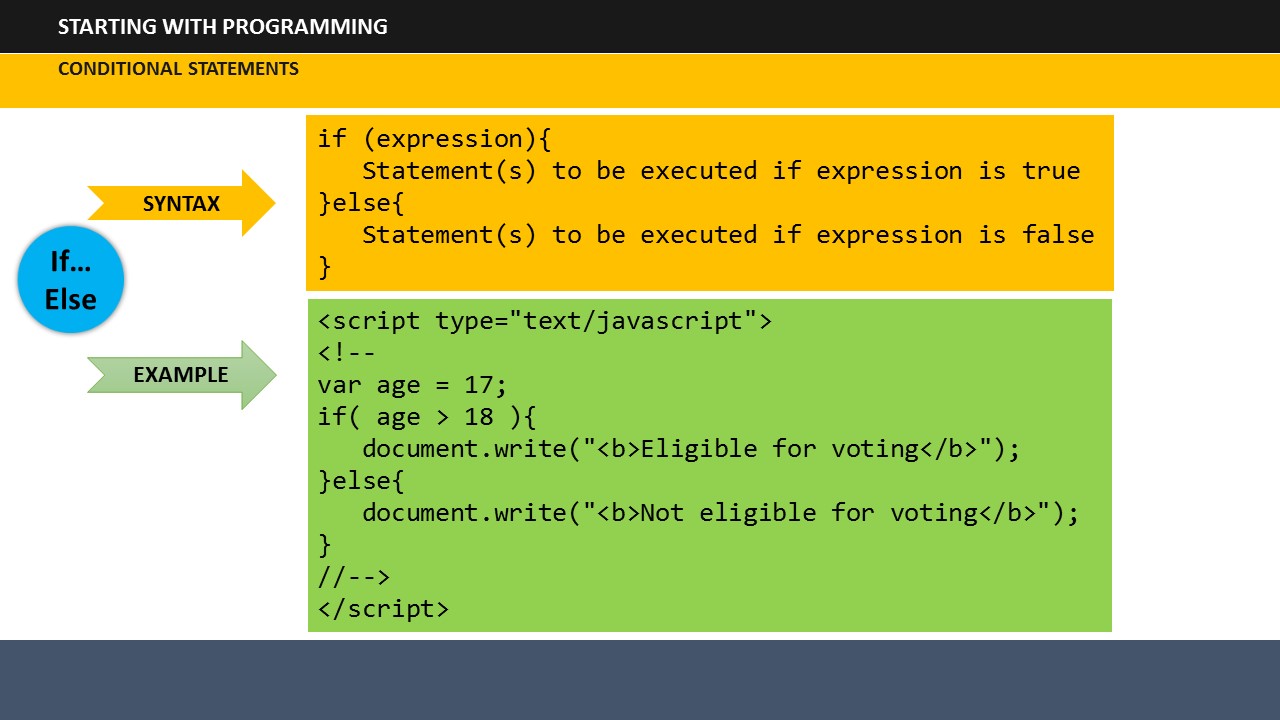
1. **Say**: The JavaScript conditional statements help in changing the flow of programs. In other words, they allow different actions to be taken for different decisions.
2. **Say**: JavaScript supports the following conditional statements:

* **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.

1. **Say**: Let’s look at the examples showing how to use these statements.
2. **Say**: 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 obviously is more than 18, so the result would say, Eligible for voting.



1. **Say**: Next, we have the syntax for if…then statement. 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”.



**Say**: Lastly, take a look at the syntax of If…Else…If statement. Here 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.

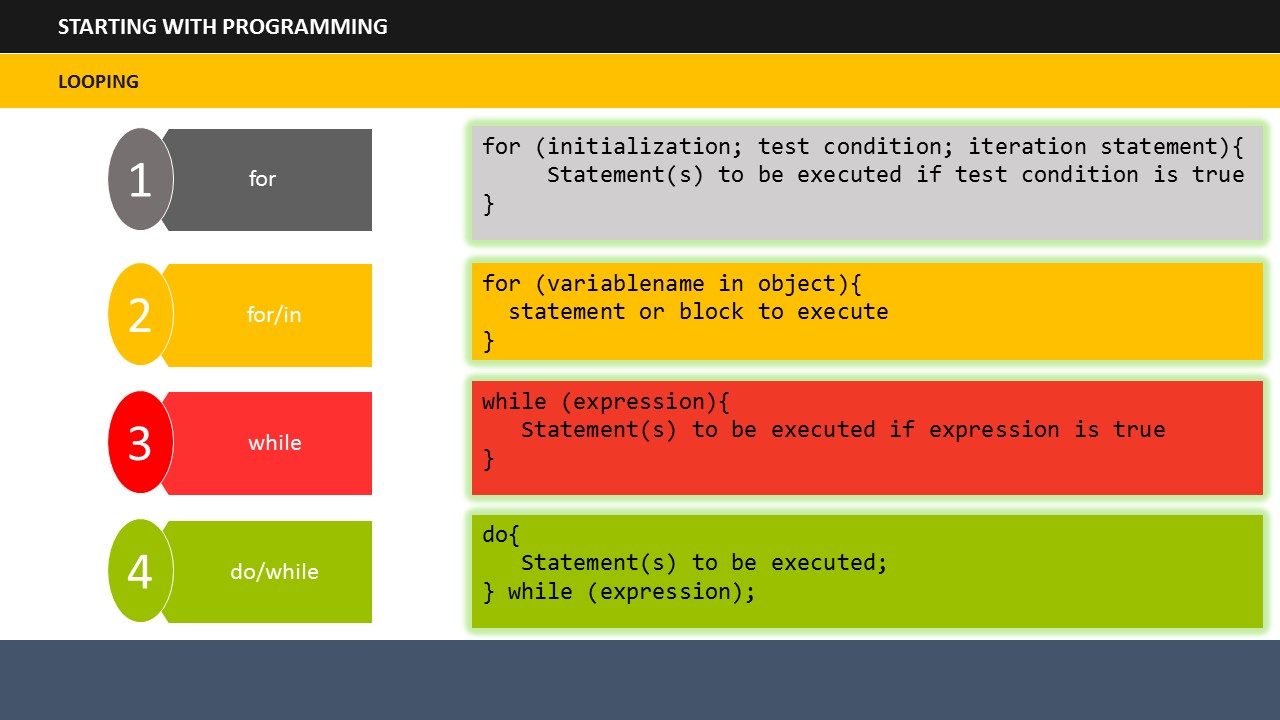
## 

**SUBTOPIC 5: LOOPING**

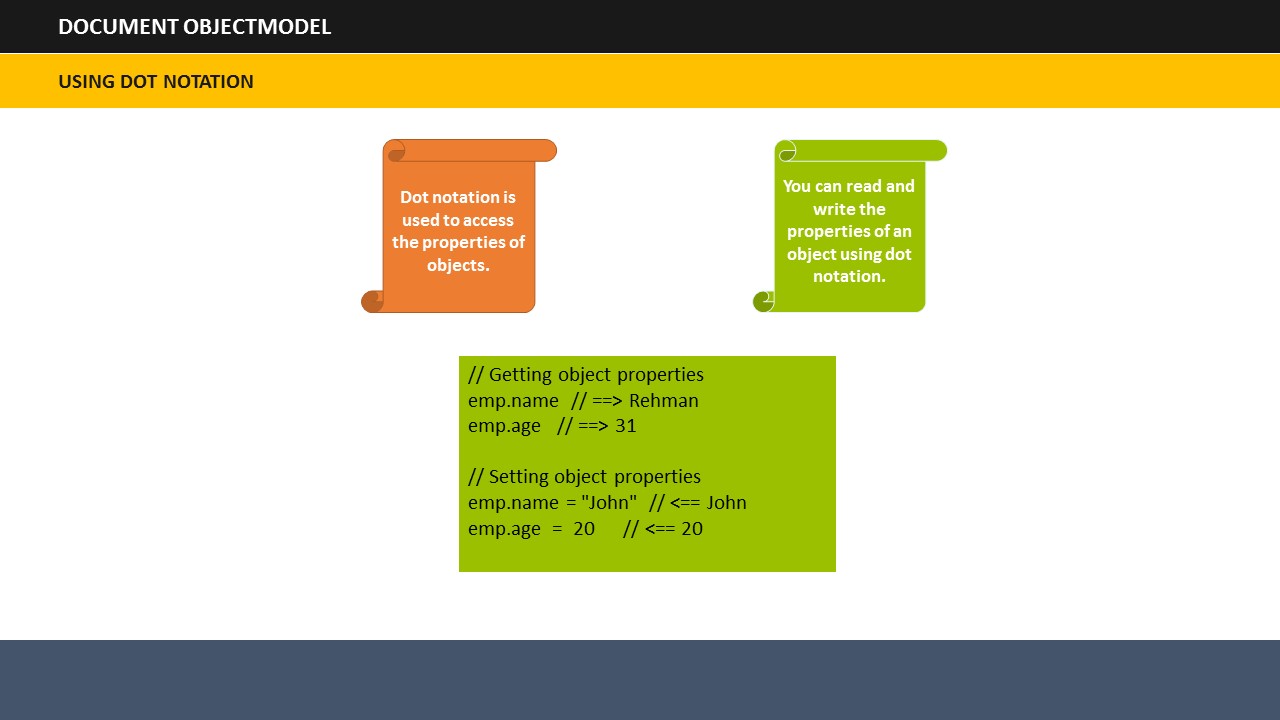
1. **Say**: Loops are used to execute a code again and again, but every time with a different value.
2. **Say**: JavaScript supports the following types of loops:

* For
* For/in
* While
* Do/while

1. **Say**: The for loop loops through a block of code multiple times.
2. **Say**: The for/in loop loops through the properties of an object.
3. **Say**: The while loop loops through a block of code as long as a specified condition is true.
4. **Say**: 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. **Say**: Take a look at the syntax given on the screen for each of these loops.



**TOPIC 6: DOCUMENT OBJECTMODEL**

**SUBTOPIC 1: USING DOT NOTATION**

1. **Say**: Dot notation is used to access the properties of objects. You can read and write the properties of an object using dot notation.
2. **Say**: Take a look at the example given on the screen.

## 

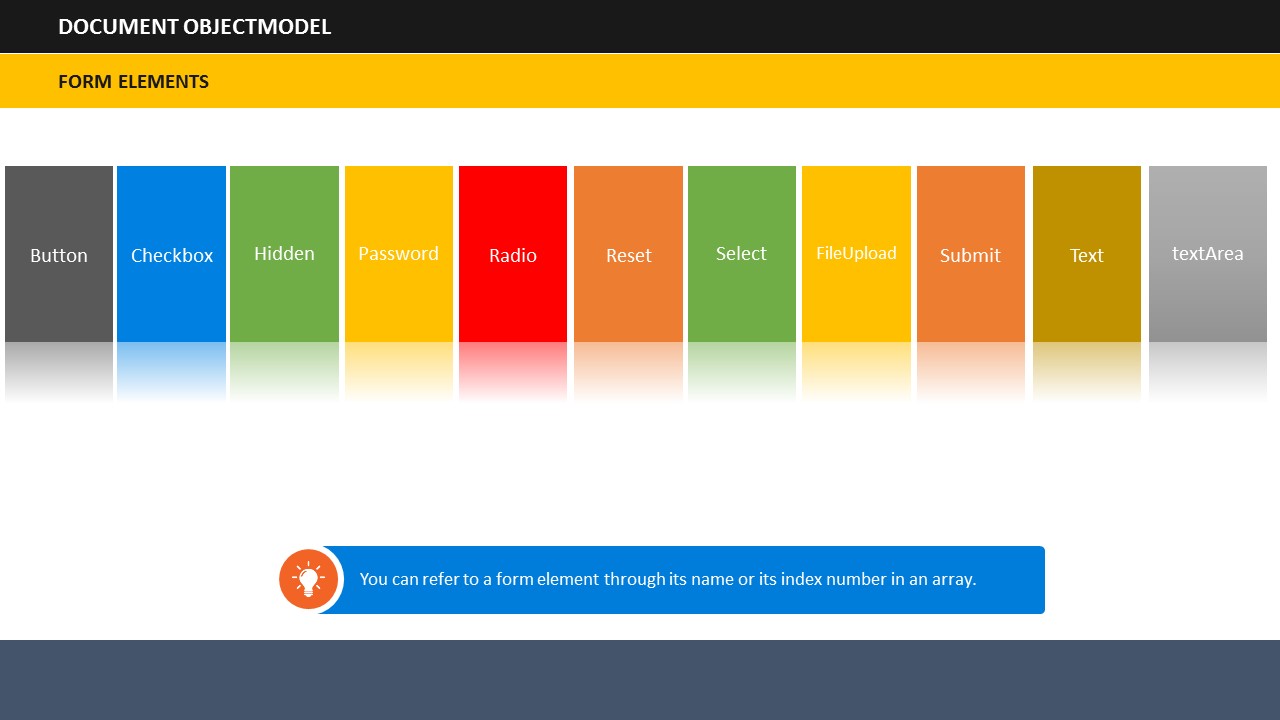
**SUBTOPIC 2: DOCUMENT OBJECT**

1. **Say**: 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. **Say**: The objects are generally organized in a hierarchy in an HTML document.
3. **Say**: Let’s take a look the hierarchy of these objects.
4. **Say**:First is the window object which is at the top level in the hierarchy.
5. **Say**: Next is the document object. It contains the contents of a page. Each HTML document loaded into a browser window becomes a document object.
6. **Say**:Then comes the form object. Any object that is enclosed in the <form> tags falls into the category of form objects.
7. **Say**:The last are the form control elements. They contain all the elements that define a form object. For example, radio buttons, text fields, checkboxes etc.

## 

**SUBTOPIC 3: FORM COLLECTION**

1. **Say**: Document objects have certain properties.
2. **Say**: One of such properties is form elements collection. The form collection returns a collection of all elements in a form.
3. **Say**: The syntax for forms collection command is document.forms.length

**TYPES OF FORM ELEMENTS**

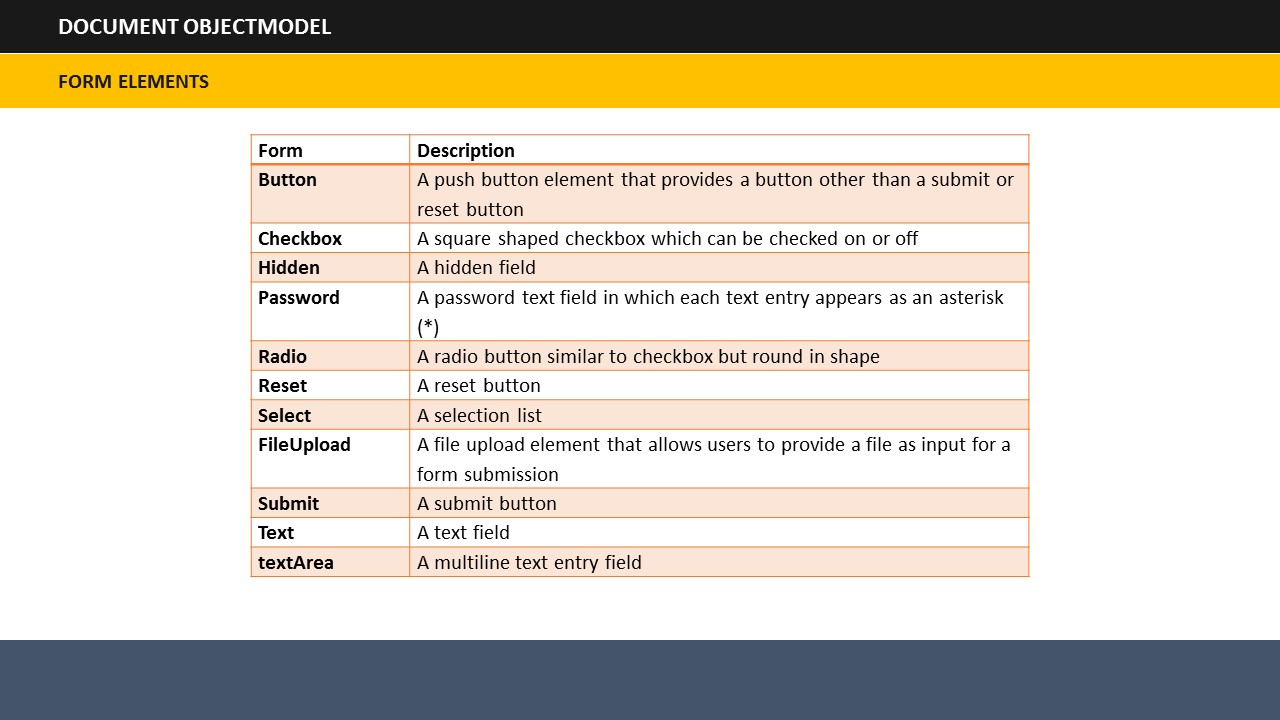
## Say: Form elements can be of several types such as:

**SUBTOPIC 4: FORM ELEMENTS**

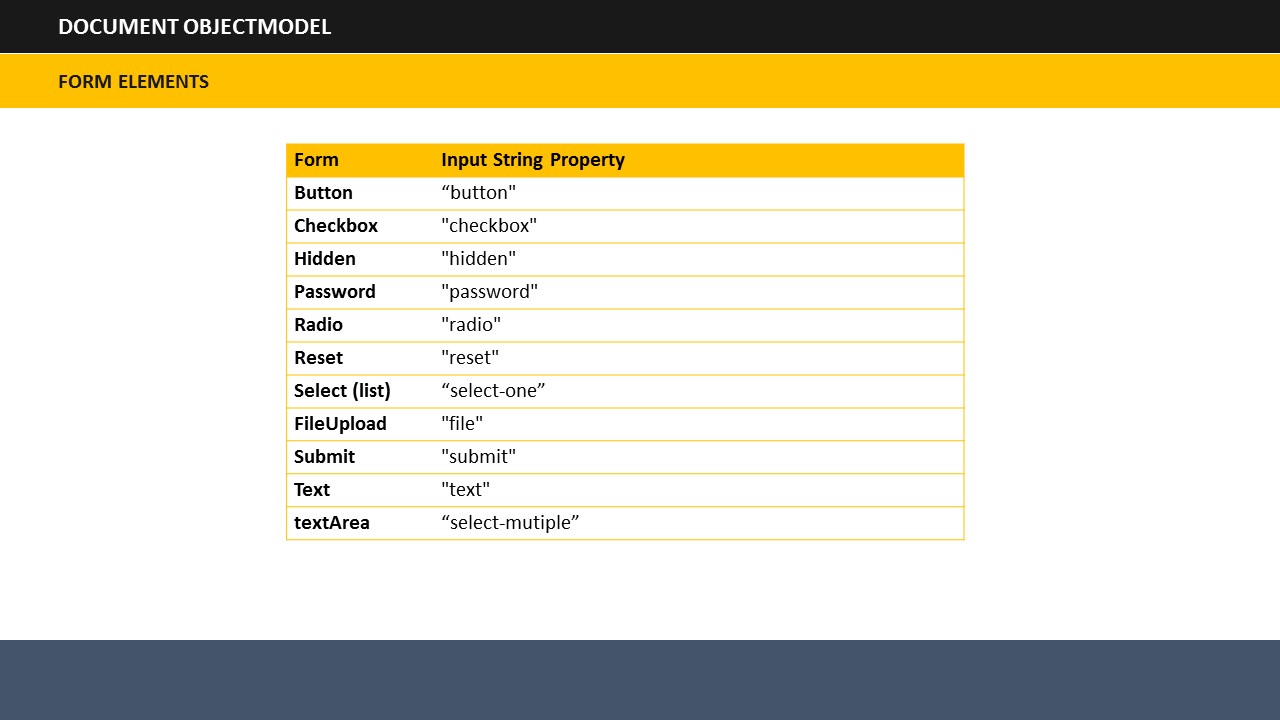
* Button
* Checkbox
* Hidden
* Password
* Radio
* Reset
* Select
* FileUpload
* Submit
* Text
* textArea

**Say**: You can refer to a form element through its name or its index number in an array.

**Say**: Each of these elements has its own properties in JavaScript.

**DESCRIPTION OF FORM ELEMENTS**

|  |
| --- |
| 1. **Say**: Button -> This provides a push button that provides functionality other than submit or reset button. |
| 1. **Say**: Checkbox -> This provides a square shaped checkbox which can be checked on or off. |
| 1. **Say**: Hidden -> This provides a hidden field. |
| 1. **Say**: Password -> This provides a password text field in which each text entry appears as an asterisk. |
| 1. **Say**: Radio -> This provides a radio button similar to checkbox but round in shape. |
| 1. **Say**: Reset -> This provides a reset button. |
| 1. **Say**: Select -> This provides a selection list. |
| 1. **Say**: FileUpload -> This provides a file upload element that allows users to provide a file as input for a form submission. |
| 1. **Say**: Submit -> This provides a submit button. |
| 1. **Say**: Text -> This provides a text field. |
| 1. **Say**: textArea -> This provides a multiline text entry field. |

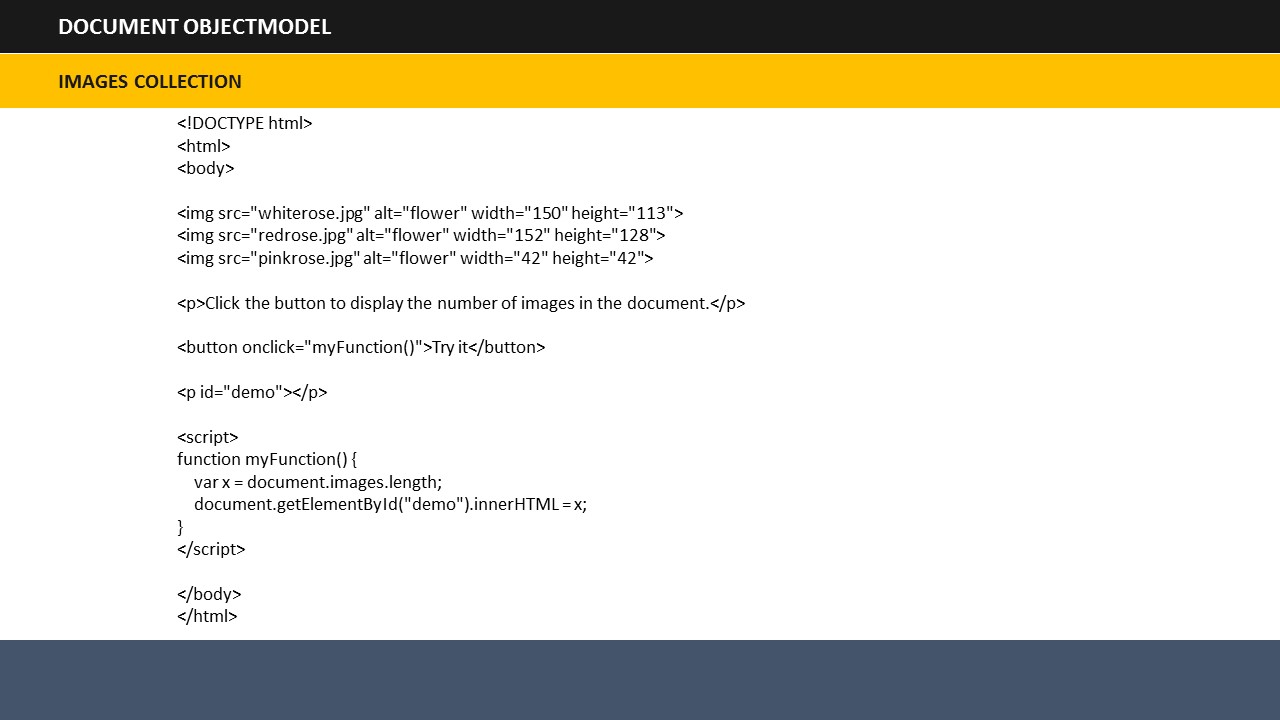
**  
  
TYPE PROPERTIES**

1. **Say**: Each of the form elements has a type property. This property is a string value that shows the type of input element.
2. **Say**: The strings reflected by the various type properties for each form element are shown on the screen.

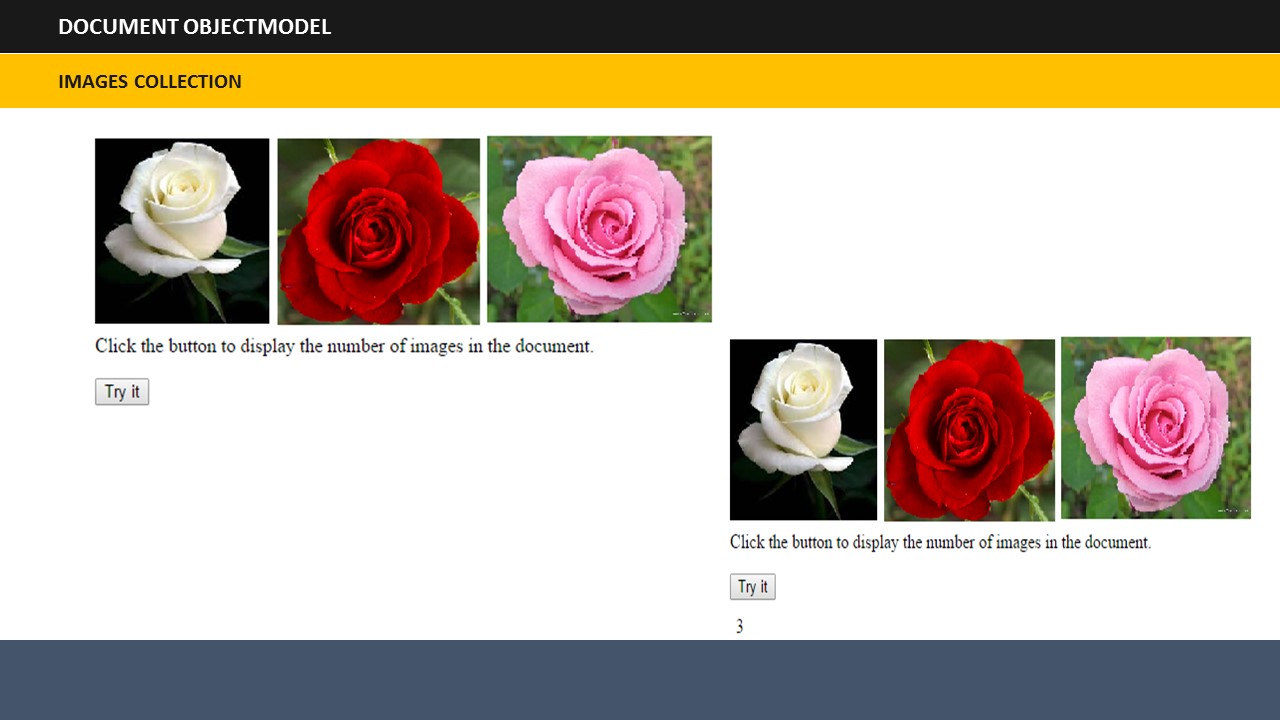
## 

**SUBTOPIC 5: IMAGES COLLECTION**

1. **Say**: The images collection returns a collection of all <img> elements in the document.
2. **Say**: The syntax for image collection is document.images.length



1. **Say**: For example, take a look at the script shown on the screen. It shows that the page has three images.



1. **Say**: This script will display the result as shown on the screen and when you will click the Try It button, the image collection code will show you the result as 3 which is the actual number of images in the HTML page.

## 

**SUBTOPIC 6: DIFFERENT TYPES OF OBJECTS**

1. **Say**: JavaScript has two types of objects: User-Defined Objects and Built-In Objects
2. **Say**: 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. **Say**: The built-in objects can be accessed from anywhere in your program. They will function in the same way irrespective of the browser used.
4. **Say**: The available built-in objects are:

* Number Object
* Boolean Object
* String Object
* Array Object
* Date Object
* Math Object
* Regular Expression Object
* Regular Expression Object

# 

**TOPIC 7 – OBJECT ORIENTED PROGRAMMING**

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

**Say:** Let us begin by outlining the main learnings of this session. This will help you understand what tasks you will be able to perform after completing this session.

1. **Say**: In this session, you will learn about…
2. **Say**: … building objects in JavaScript and…
3. **Say**: … working of JavaScript and HTML.
4. **Say**: You will also learn about Data Types, Variables, Operators and Control Structures in JavaScript.
5. **Say**: In addition, you will also learn about using loops....
6. **Say**: … creating and manipulating arrays and…
7. **Say**: … using object hierarchy.
8. **Say**: Finally, you will learn about managing events.

**Do:**

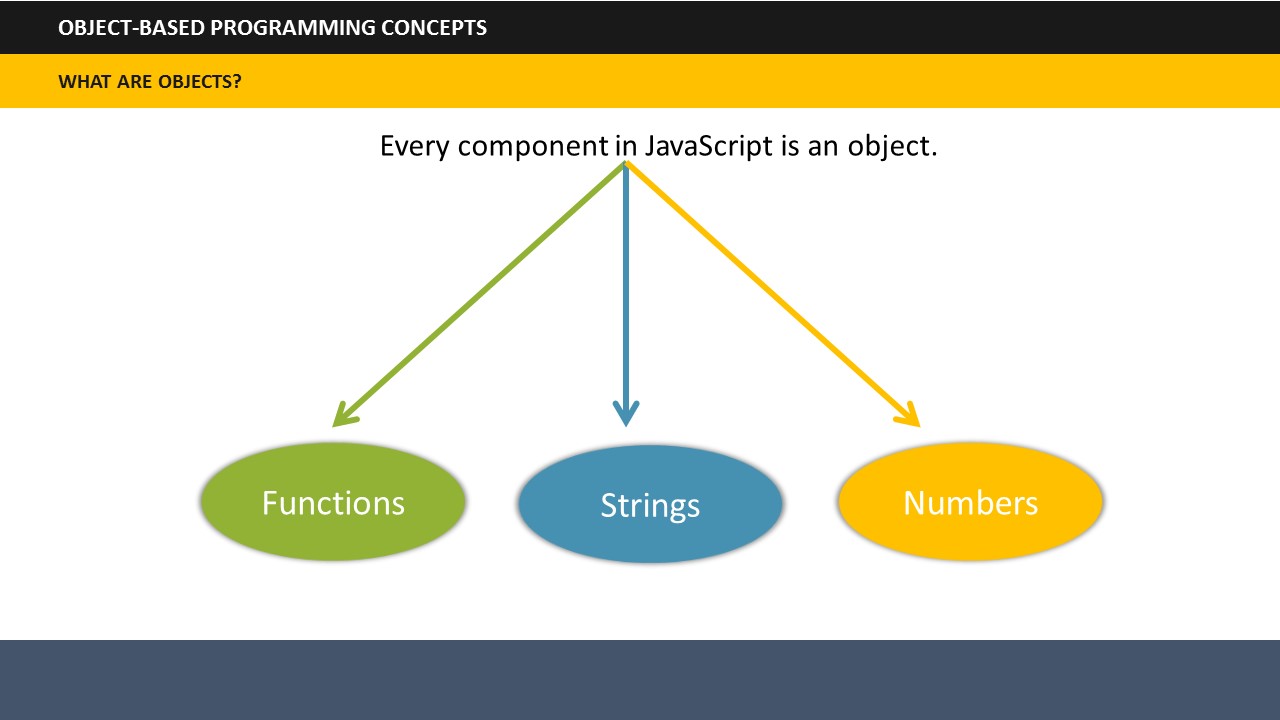
* Invite participant’s inputs on the objectives – asking them to state their own objectives/problem areas. In case such objectives/problem areas are relevant, write them on the flipchart. Remember to cover such points at the appropriate time in the training session. Also, remember to discuss them again during summarization.
* You can refer to these objectives during the summarization session to know whether the objectives have actually been met or not.

# 

**SUBTOPIC 1: WHAT ARE OBJECTS?**

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

1. **Say**: By now, you already have a basic understanding of JavaScript. In this session, we will build up on this understanding by going through a few key concepts in detail. One of these concepts is significance of objects in JavaScript. **Do:** Ask participants if they know how objects are related to JavaScript. Encourage them to share their understanding even if they don’t know much about it.
2. **Say:** Did you know that JavaScript is called as an Object Oriented Programming or OOP language. In other words, we can say that using JavaScript involves working with what are known as objects.



1. **Say**: So, what are these objects? 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.

# 

**TOPIC 9: BUILDING OBJECTS IN JAVASCRIPT**

**SUBTOPIC 1: CREATE NEW OBJECTS**

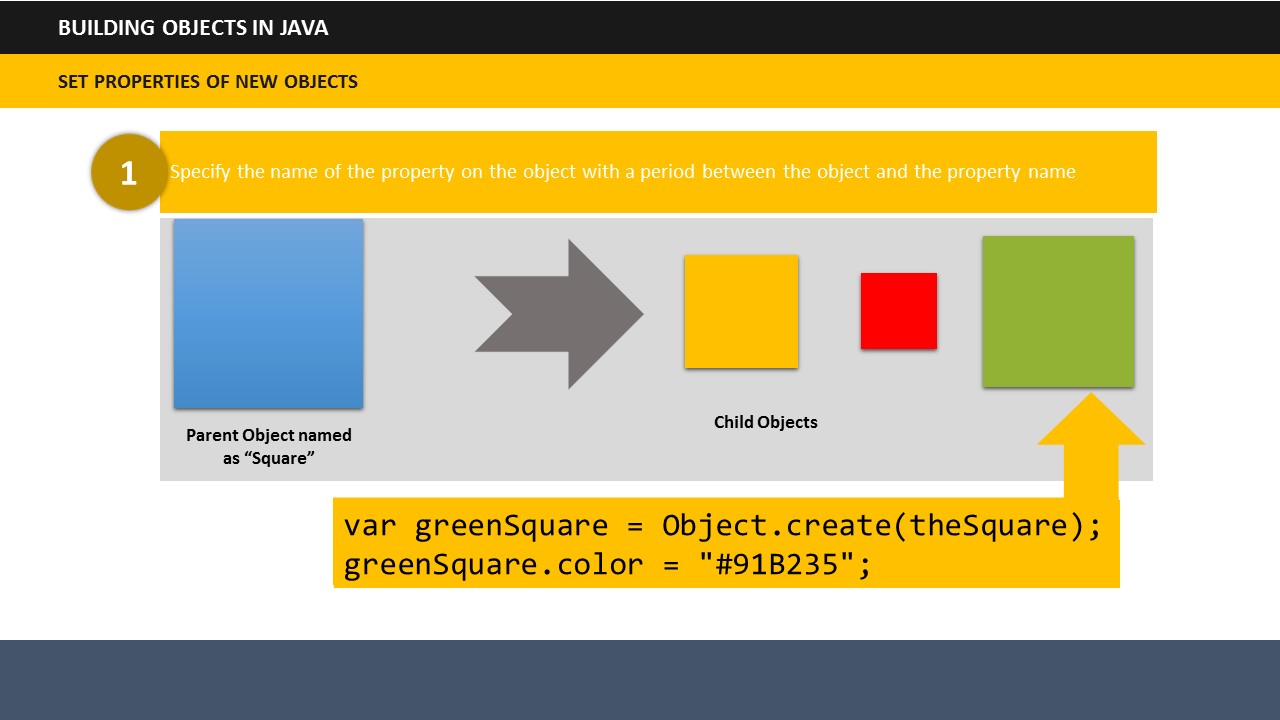
1. **Say**: Now that you have an understanding of objects, let’s see how we can **create** objects in JavaScript. This is quite simple.
2. **Say**: The key to creating objects is the ***Object.create*** function. This is like creating child objects from a parent object as show on the screen. You can easily link the function to the name of the object from which you want to create your new object.
3. **Say**: In the example shown on the screen, we can link the function with the parent blue square, which will define the child objects that are the orange, red and green squares.

## 

## 

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

1. **Say**: The next step is to set the properties of the newly created object.
2. **Say**: The first method involves specifying the name of the property on the object with a period between the object and the property name.
3. **Say**: The other method involve setting the property using bracket syntax. Let’s understand these two methods using examples.

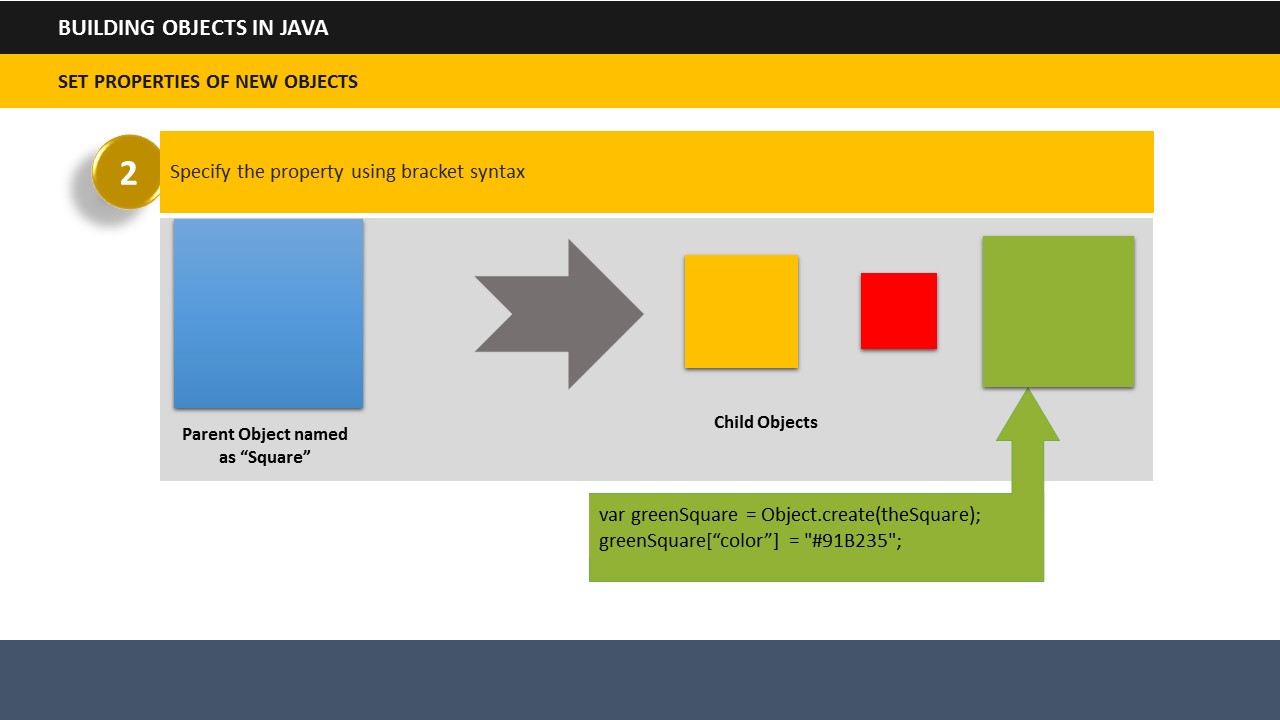


1. **Say**: Let’s take the example of parent and child objects that we just saw. So, if you want to set the color property of the green square using the first method, the script will look like this:

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

**Do:** Show the codes on the screen.

1. **Say**: Notice that we have set the color property on our green circle object to a value of #91B235.

****

1. **Say**: Next, let’s see how we can set this green color using the second method of bracket syntax.
2. **Say**: Notice that in the code the word “color” has been placed in square brackets. The end result will be the same irrespective of the method that is the square will be assigned a green color.

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

**Do:** Show the codes on the screen.

# 

**TOPIC 10: JAVASCRIPT AND HTML**

1. **Do:** Invite participants to share their knowledge. Ask them if they know how JavaScript and HTML are linked together.
2. **Say**: JavaScript can change HTML attributes.
3. **Say**: You can include JavaScript code anywhere in an HTML document. But there are many ways to do it such as:

* 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.

# 

**SUBTOPIC 1: VARIABLES**

**TOPIC 11: VARIABLES & DATA TYPES**

1. **Say**: Like many other programming languages, JavaScript uses variables, operators and data types. Let’s go through each of them.
2. **Say**: 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. **Say**: Let’s see examples of variables.

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

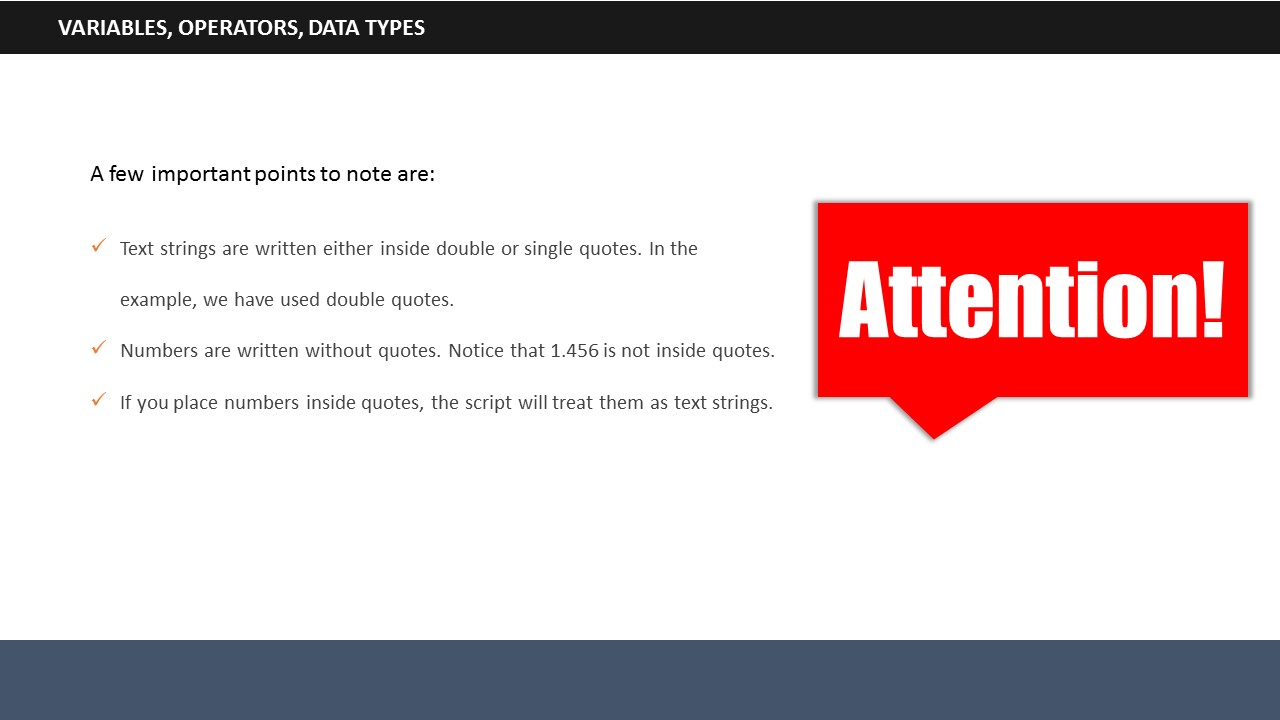
1. **Say**: Here a, b, and c are the containers which are storing their corresponding values 101, 102, 103.

## 

**SUBTOPIC 2: DATA TYPES**

1. **Say**: Several types of data can be used in JavaScript, but for now we will only focus on two types: Numbers and Text Strings.
2. **Say**: Text string here refers to text values.
3. **Say**: For example:

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



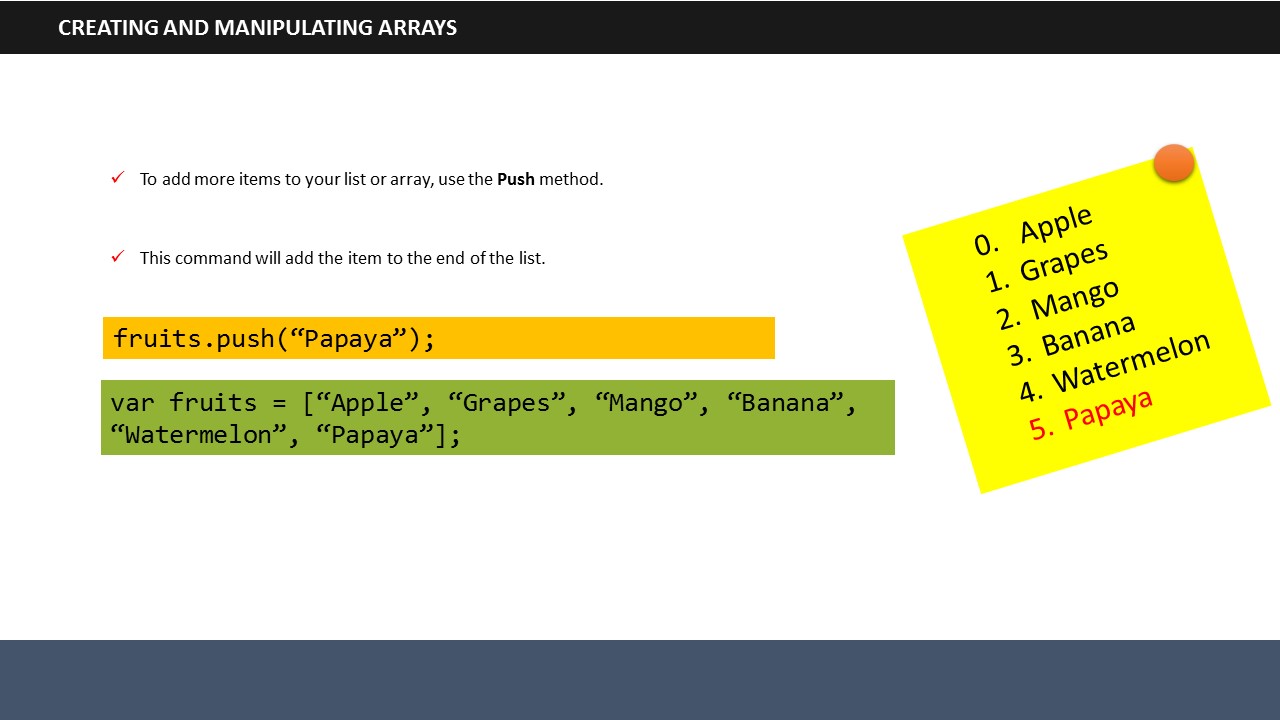
1. **Say**: A few important points to note are:

* 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.

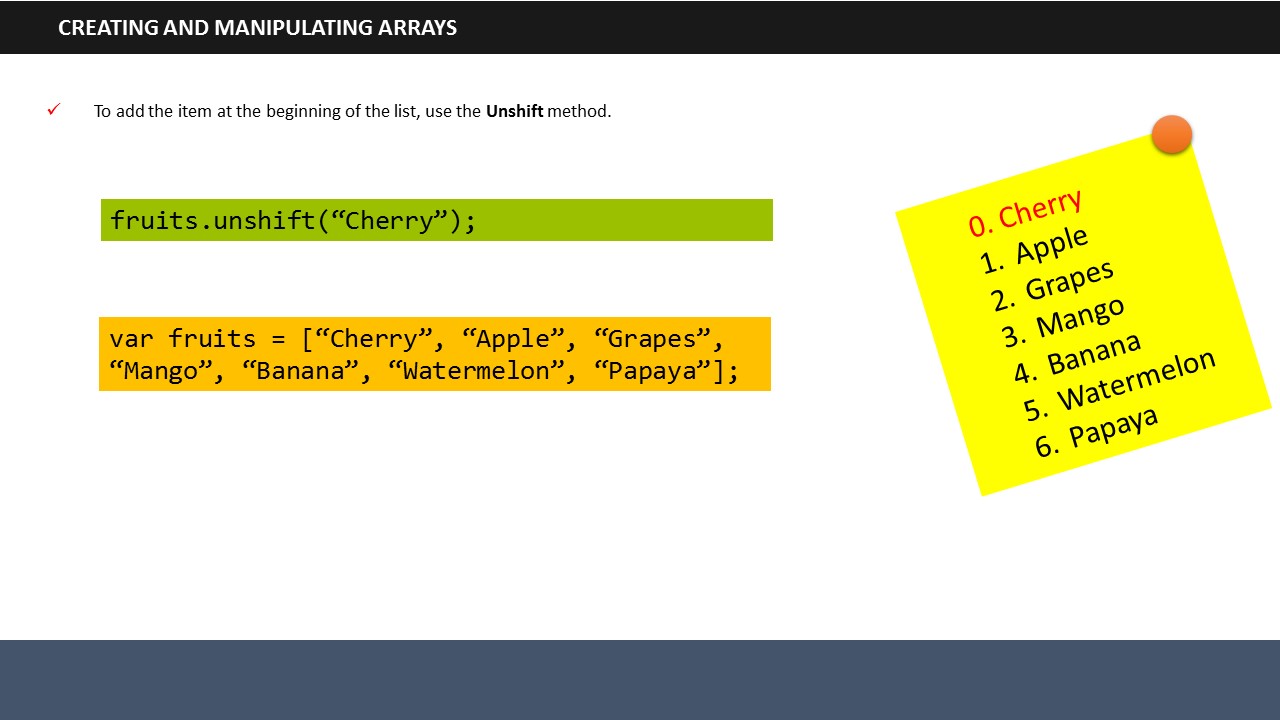
# 

**TOPIC 12: CREATING AND MANIPULATING ARRAYS**

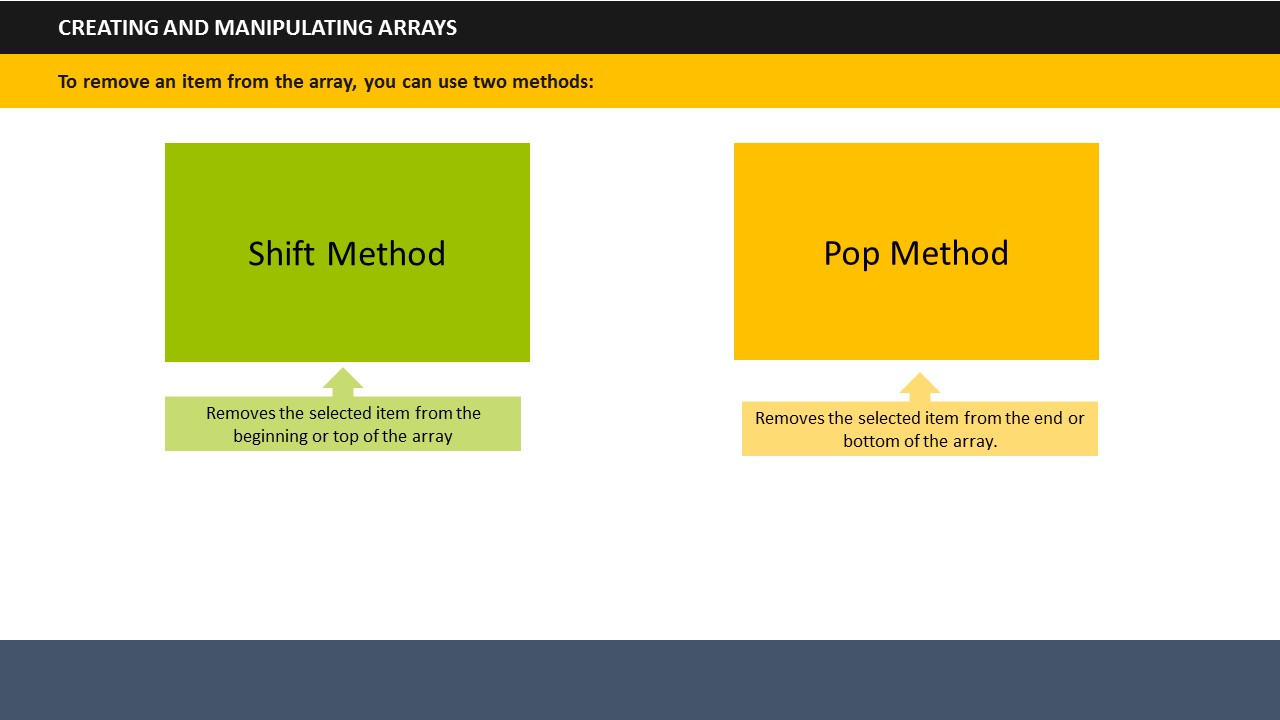
1. **Say**: An array can be defined as a list/sequence of values. The value in array can be a number, object, function and so on. You can store any form of data in an array.
2. **Say**: If you wish to create an array, as shown in the example you need to place the variable name on the left, and the variable values on right in a pair of brackets that initialize this variable as an empty array. In this example, we have made an array of fruits.



1. **Say:** If you want to add more fruits to the array, we use a method called as the push method. This command adds the item to the end of the list.
2. **Say**: As shown on the screen, we have added Papaya in the list using the push command and it gets added right at the end of the list.



1. **Say**: On the contrary, if you want to add a fruit, say Cherry at the beginning of the list, use the unshift command.

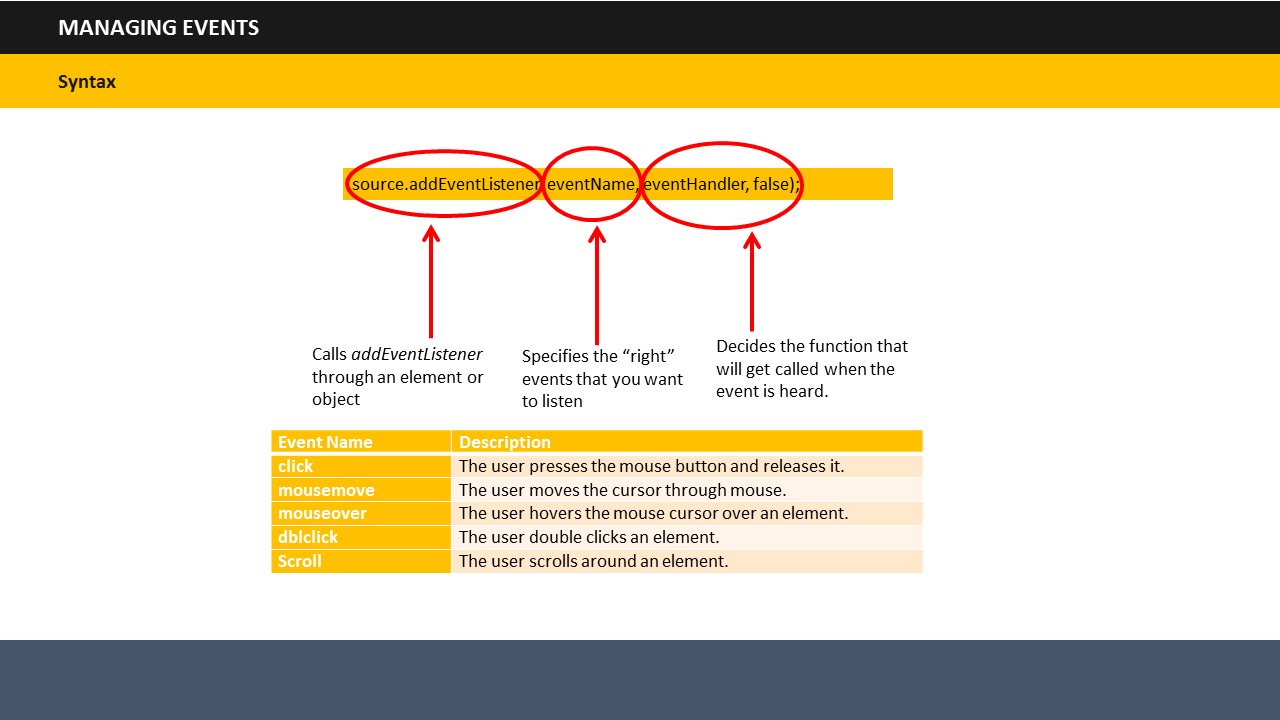


1. **Say**: To remove an item from the array, you can use two methods: Shift and Pop.
2. **Say**: The shift method removes the selected item from the beginning or top of the array.
3. **Say**: The pop method removes the selected item from the end or bottom of the array.

# 

**TOPIC 13: MANAGING EVENTS**

1. **Say**: Events in JavaScript are the signals that happen due to some action. For example, pressing a keyboard button, clicking the mouse, loading of HTML are all examples of events.
2. **Say**: Events define what action should be taken when an action transpires.
3. **Say**: To manage events, you need to listen for events and then react accordingly.
4. **Say**: 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.



1. **Say**: The syntax used for managing events is shown on the screen. It includes a source, an event name and handler.
2. **Say**: The Source helps to call *addEventListener* through an element or object.
3. **Say**: The *Event Name* specifies the “right” events that you want to listen.
4. **Say**: A few examples of event names that can be written in place of event name in the syntax have been shown in the table.
5. **Say**: Lastly, the *Event Handler* decides the function that will get called when the event is heard.

# 

**TOPIC 14: SUMMARY**

***SUMMARIZATION ACTIVITY***

**Duration:** 10 Minutes

**Say:** It’s time to review what we have learned so far. I want you all to write in the space given in your participant’s guide, what you have learned today.

**Say:** Let’s outline the learnings from today’s session. (After giving 5 minutes time for writing summary points)

**Do:** Invite participants to raise their hands and share their take-away and learnings from each topic.

***Facilitator’s Notes:***

**Do:** Check if all the points given here are covered in the summarization activity.

**SUBTOPIC 1: RECAP**

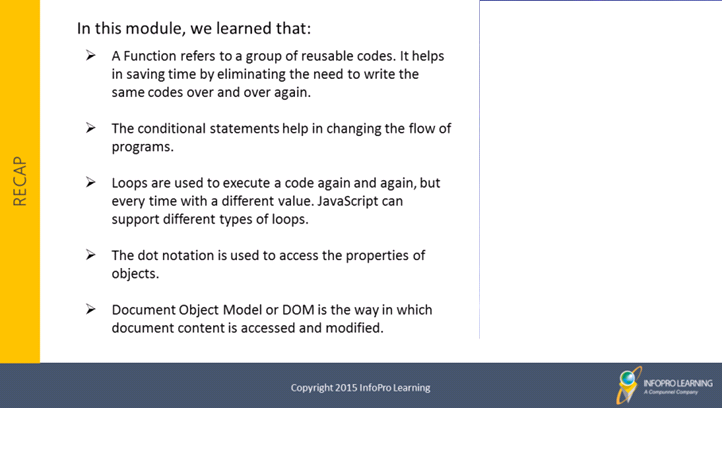


**Say:** In this module, we learned that:

* JavaScript is an object based language. Every component in JavaScript is an object.
* You can insert JavaScript in <head> section / <body> section or both the section of an HTML page.
* JavaScript uses variables, operators, and data types.
  + Variables store data values. You can place data into these containers and name these containers.
  + Various types of operators are used to perform different functions:
    - * Arithmetic Operators
      * Assignment Operators
      * Comparison Operators
      * Logical Operators

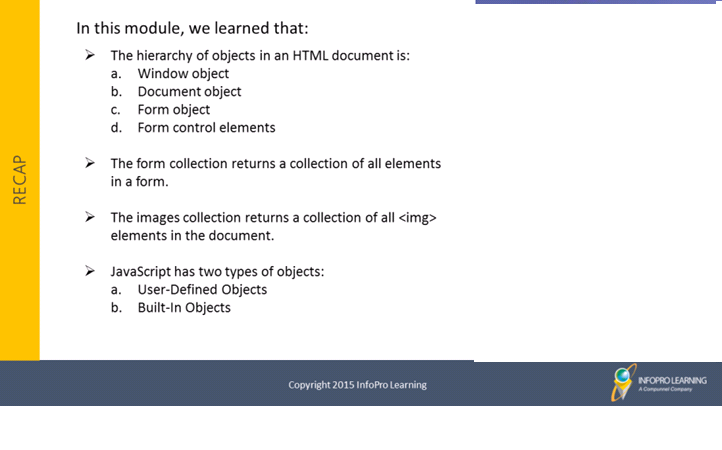
|  |
| --- |
|  |

|  |
| --- |
|  |



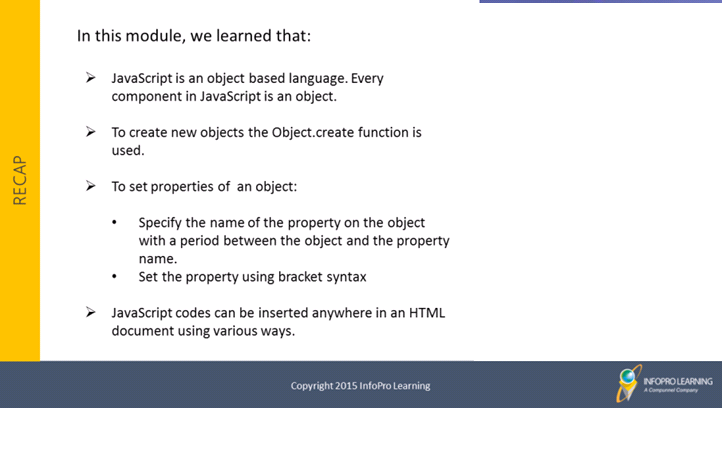
**Say:** We also learned that:

* A Function refers to a group of reusable codes. It helps in saving time by eliminating the need to write the same codes over and over again.
* The conditional statements help in changing the flow of programs.
* Loops are used to execute a code again and again, but every time with a different value. JavaScript can support different types of loops.
* The dot notation is used to access the properties of objects.
* Document Object Model or DOM is the way in which document content is accessed and modified.



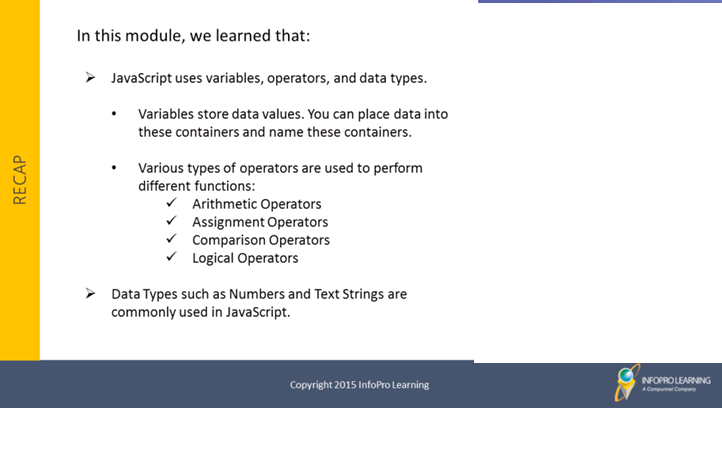
**Say:** Finally we learned that:

* The hierarchy of objects in an HTML document is:
  1. Window object
  2. Document object
  3. Form object
  4. Form control elements
  5. The form collection returns a collection of all elements in a form.
* The images collection returns a collection of all <img> elements in the document.
* JavaScript has two types of objects:
  1. User-Defined Objects
  2. Built-In Objects



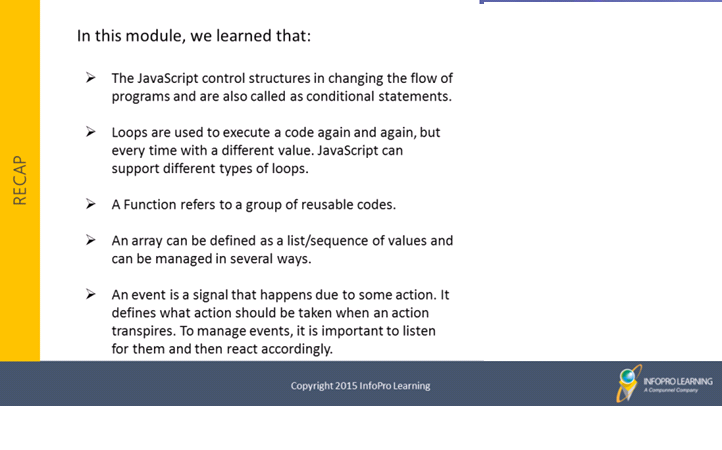
**Say**: In this module, we learned that:

* JavaScript is an object based language. Every component in JavaScript is an object.
* To create new objects the Object.create function is used.
* To set properties of an object:
  + Specify the name of the property on the object with a period between the object and the property name.
  + Set the property using bracket syntax
* JavaScript codes can be inserted anywhere in an HTML document using various ways.



**Say**: In this module, we also learned that:

* JavaScript uses variables, operators, and data types.
  + Variables store data values. You can place data into these containers and name these containers.
  + Various types of operators are used to perform different functions:
* Arithmetic Operators
* Assignment Operators
* Comparison Operators
* Logical Operators
* Data Types such as Numbers and Text Strings are commonly used in JavaScript.



**Say**: Finally we learned that:

* The JavaScript control structures in changing the flow of programs and are also called as conditional statements.
* Loops are used to execute a code again and again, but every time with a different value. JavaScript can support different types of loops.
* A Function refers to a group of reusable codes.
* An array can be defined as a list/sequence of values and can be managed in several ways.
* An event is a signal that happens due to some action. It defines what action should be taken when an action transpires. To manage events, it is important to listen for them and then react accordingly.

## 

**SUBTOPIC 2: FINAL TIPS AND SUMMARIZATION**

**Say**: This is the end of session. I hope you understood the concepts that we covered today and would be able to apply them while working.

**Do:** Use this moment to clarify any doubts and provide any tips to the learners.

# 

**TOPIC 15: CHECK YOUR UNDERSTANDING:**

1. You can include JavaScript code anywhere in an HTML document.
   1. **True**
   2. False
2. Match the following:

|  |  |  |
| --- | --- | --- |
| Type of Loop |  | Function |
| For | A | Loops through a block of code as long as a specified condition is true. |
| For/in | B | Executes the code block once before checking if the condition is true and then repeats the loop as long as the condition is true. |
| While | C | Loops through the properties of an object |
| Do/while | D | Loops through a block of code multiple times |

Answer:

* For -> D
* For/in -> C
* While -> A
* Do/while -> B

1. A function does not use reusable codes.  
   1. True
   2. **False**
2. Which of the following conditional statements is used to specify execution of a block of code when the same condition is false?  
   1. If
   2. **If…Else**
   3. If…Else…If
3. Which of the following loops through a block of code multiple times?  
   1. **For**
   2. For/in
   3. While
   4. Do/while
4. Arrange the following objects in the correct hierarchy as accepted in an HTML document.

|  |  |  |  |
| --- | --- | --- | --- |
| Document object | Form control elements | Form object | Window object |

Answer: The correct order is:

1. Window object
2. Document object
3. Form object
4. Form control elements
5. Which of the following can be classified as objects in JavaScript?  
   1. Functions
   2. Numbers
   3. Text Inputs
   4. **All of the above**
6. The function used to create a new object is:  
   1. **Object.create**
   2. Create.object
7. You can include JavaScript code anywhere in an HTML document.
   1. **True**
   2. False
8. Which of the following data types is placed in double quotes in the syntax?
   1. **Text strings**
   2. Numbers
   3. Both Text Strings and Numbers
   4. None of the above
9. If you place numbers inside quotes, the script will treat them as text strings.  
   1. **True**
   2. False
10. Match the following:

|  |  |  |
| --- | --- | --- |
| Type of Loop |  | Function |
| For | A | Loops through a block of code as long as a specified condition is true. |
| For/in | B | Executes the code block once before checking if the condition is true and then repeats the loop as long as the condition is true. |
| While | C | Loops through the properties of an object |
| Do/while | D | Loops through a block of code multiple times |

Answer:

* For -> D
* For/in -> C
* While -> A
* Do/while -> B

1. The push method is used to:
   1. Add items to the beginning of a list
   2. **Add items to the end of a list**
   3. Remove items from the beginning of a list
   4. Remove items from the end of a list



**ITES Functional Skills Training**

# LAB EXERCISES



* You may want to bring your FacilitatorGuide to labs to refer to the content and examples.
* If the participants have a question, tell them that they can ask you for help, or look at their ParticipantGuide or lecture slides.
* Encourage the participants to talk to their classmates; tell them it is okay to share code and ideas during lab.
* Tell the participants that they are not expected to finish all the exercises. They simply need to do as much as they can in the allotted time. They don't need to finish the rest after they leave the lab.
* Before they leave, tell them to check in with you to get credit for their work.



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

**Hint**: Create a function and then call it by passing arguments.

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

**Hint**: Use the “+” operator to concatenate.

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

**Hint**: Use join() method.

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

**Hint**: Use whichever loop you want to use to make this program.

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

**Hint**: use a drop down list, a button, and make use of selected index to perform the action**.**

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

**Hint**: Use document.forms[index].name.

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

**Hint**: Use image.src to perform the action.

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

**Hint**: Use onmouseover and onmouseevent.