Web Programming

Dr. D. Natarajasivan
Lecturer
TNPT

Topics

- Need of Scripting Languages
- Variables and Data Types
  - Declaring Variables
  - Lifespan of variables
  - Data Types

What is Scripting Language?

- A Scripting Language is executed at runtime.
- Scripts are interpreted one by one at runtime.
- Two categories of scripting language
  - Server Side Scripting Language
  - Client Side Scripting Language

Introduction

- JavaScript is a client side scripting language
- It is lightweight and it can be integrated with HTML.
- JavaScript created by Brendan Eich in 1995

- It was called as Mocha, later renamed to LiveScript by Netscape
- LiveScript is renamed as JavaScript
- In 1997, JavaScript 1.1 was standardized and renamed as ECMAScript

Need of Scripting Language

- Reduce web server overhead
- Make website responsive
- Execute code faster
- Create interactive user interface
- Create Open Source Tools

Basics

- JavaScript implementation syntax

  <script>

  </script>

- It is recommended to place it within <head> tag.
- Two attributes
  - type : text/javascript
  - src : link to external file

- 2 ways of writing JavaScript Code
  - Internal Scripting
  - External Scripting
- Internal Scripting
  - Uses <script> tag in <head> or <body> of HTML page

Example 1

- **External Scripting**
  - Placed in external file with extension ".js"
  - Inserted into HTML file using "scr" attribute in <script> tag

```
<script src=""></script>
```

Example 2

Identifiers

- Identifiers: name of variable, function or class
- Rules
  - First character alphabet, underscore(_) or dollar sign($)
  - Other characters alphabet, numbers, underscore(_) and dollar sign($)
- JavaScript is case sensitive

Variable Declaration

- **Variables**
  - Variables are loosely typed in JavaScript
  - Declared using *var* keyword

  Syntax

  ```
  var varname;
  varname = value;

  var  varname = value
  ```

Constants

- Used to create a read-only value.
- Syntax

```
const const_name = value;
```

Example 3

Lifespan of variables (Scope)

- Scope of a variable specify where the variable can be accessed
- Two Scopes
  - Global Scope: Accessed anywhere in the whole script
  - Local Scope: Accessed only within the function where it is defined

Example 4

Data Types

- Numbers
- String
- Boolean
- Null
- Undefined

Example 5

Unit 3

Section 3.1

Topics

- Need of Scripting Languages
- Variables and Data Types
  - Declaring Variables
  - Life span of variables
  - Data Types

Web Programming

Dr. D. Natarajasivan
Lecturer
TNPT

Topics

- Operators
- Control Structure
  - Conditional Statements

Operators

- An operator is a symbol used to perform operations on operands

  - Arithmetic Operators (computational)
  - Assignment Operators
  - Comparison Operators
  - Logical Operators
  - Special Operators

Arithmetic Operators

Operator	Description
+	Binary Addition, Unary convert to number
-	Binary Subtraction, Unary convert to number
*	Binary Multiplication
/	Binary Division
**	Binary Exponential
%	Binary Modulus
++, --	Unary Increment/Decrement

Example 6

Assignment Operator

Operator	Description
=	Assign right to left
Shorthand assignment	
+=	
-=	
*=	
/=	
%=	

Comparison Operator

Operator	Description
==	Is equal to
===	Identical
!=	Not equal to
!==	Not identical
>, <	Greater than Less than
<=	Less than or equal to
>=	Greater that or equal to

Example 7

Logical Operators

Operator	Description
&&	Logical AND
||	Logical OR
!	Logical NOT

Special Operators

Operator	Description
?:	Conditional Operator
new	Create new Object
typeof	Check the type of object

Control Structure

- Used to control the program flow
- Two types
  - Conditional Statement
  - Looping Statement

Conditional Statements

- Based on some condition a specific action is performed
- Types of conditional statements
  - if statement
  - if..else statement
  - if...else if statement
  - switch statement

if Statement

- When the condition is true the statements are executed.

  Syntax

```
if(condition)
{
    statements
}
```

Example 8

If..else Statement

- **Based on the condition either true part or false part is executed**

Syntax

```
if(condition)
{
    True
}
else
{
    False
}
```

Example 9

Unit 3

Section 3.1

Topics

- Operators
- Conditional Statements
  - If Statement
  - If else Statement

Web Programming

Dr. D. Natarajasivan
Lecturer
TNPT

Topics

- **Conditional Statements**
  - if statement
  - if..else statement
  - if...else if statement
  - switch statement
- **Looping Statements**

else..if Statement

- **Multiple conditions are checked**

Syntax

```
if(condition1)
{
    True
}
else if(condition2)
{
    False
}
else
{
    False
}
```

Example 10

Switch Statement

- **Value based branching statement**

Syntax

```
switch(value)
{
    case val1:

            break;
    case val2:

            break;
    default:

            break;
}
```

Example 10.1

Looping Statements

- Used to execute same block of code repeatedly
- Types of Loops
  - while loop
  - do...while loop
  - for loop
  - for/in loop
  - for/of loop

While loop

- Entry controlled loop
- Loop executed when the condition is true

Syntax

```
while(condition)
{
    Code to execute
}
```

Example 11

Do…while loop

- Exit controlled loop
- Block of code is executed once and then the condition is checked

Syntax

```
do
{
    Block if code
}while(condition);
```

Example 12

For loop

- Used to execute code block a specified number of times

Syntax

```
for(initialization;condition;inc/dec)
{
    Code block
}
```

Example 13

Unit 3

Section 3.1

Topics

- if...else if statement
- switch statement
- Loops
  - while loop
  - do...while loop
  - for loop

Web Programming

Dr. D. Natarajasivan
Lecturer
TNPT

Topics

- **Looping Statements**
  - while loop
  - do...while loop
  - for loop
  - for/in loop
  - for/of loop

For/in Loop

- Special kind of loop
- Iterates over the Properties of an object

Syntax

```
for(var in object)
{
    Code block
}
```

Example 14

For/of Loop

- Loops through the values of an array

Syntax

```
for(var of object)
{
    Code block
}
```

Example 15

Break and continue

- **Break statement is used to exit from a loop/function**

  Syntax

  ```
  break [label];
  ```

- **Continue statement is used to skip the current loop and continue**

  Syntax

  ```
  continue [loop];
  ```

Example 16

Section 3.2 - Topics

- **Functions**
  - **Basic function**
  - **Function Literals**

Basic Function

- It is a group of code which can be called anywhere in the program.
- Function allows a big program to be divided into smaller parts.
- In java script “function” keyword is used to define a function

- **Syntax**

```
function name(par1, par2,...)
{
//Code to be executed
[return variable;]
}
```

Function Call

- When an event occurs
- When it is invoked as function call
- Automatically

  Syntax

  ```
  (function fun_name(){ //Code})();
  ```

- Functions Example

Example 17

Example 18

Function Literals

- A function literal is an expression that defines an unnamed function.

Syntax

```
var varname = function (arg List){
//Function Body
};
```

Example 19

Unit 3

Section 3.1 & Section 3.2

Topics

- Looping Statements
  - For/in loop
  - For/of loop
- Functions
  - Basic Function
  - Function Literal

Web Programming

Dr. D. Natarajasivan
Lecturer
TNPT

Topics

- Executing Deferred Script
- Objects
  - Document Object Model
  - Predefined Object
    - Array Object
    - History Object
    - Location Object

Executing Deferred Script

- HTML parser is paused when an external script is loaded.
- External script is executed and then the HTML is parsed

Syntax

```
<script defer src=“”></script>
```

Example 20

Objects

- Object is a collection of properties
- Properties are “key=value” pair,
- Key: property name
- Value: any value

Example 21

Object Creation in JavaScript

- Using NEW Keyword
- Using class

Using new keyword

- Using new keyword and Object() constructor

Syntax

```
var obj = new Object();
obj.prop1 = value1;
obj.prop2 = value 2;
```

contd...

- Using new keyword with constructor function
- Syntax

```
function fun_name(parameter list)
{
    this.prop1 = value;
    this.prop2 = value;
}
var obj = new fun_name(parameter list);
```

Example 22

Using class

- Using class keyword similar to that of Java language

Syntax

```
class class_name{
constructor(parameter list){
        ..
        ..
    }
}
```

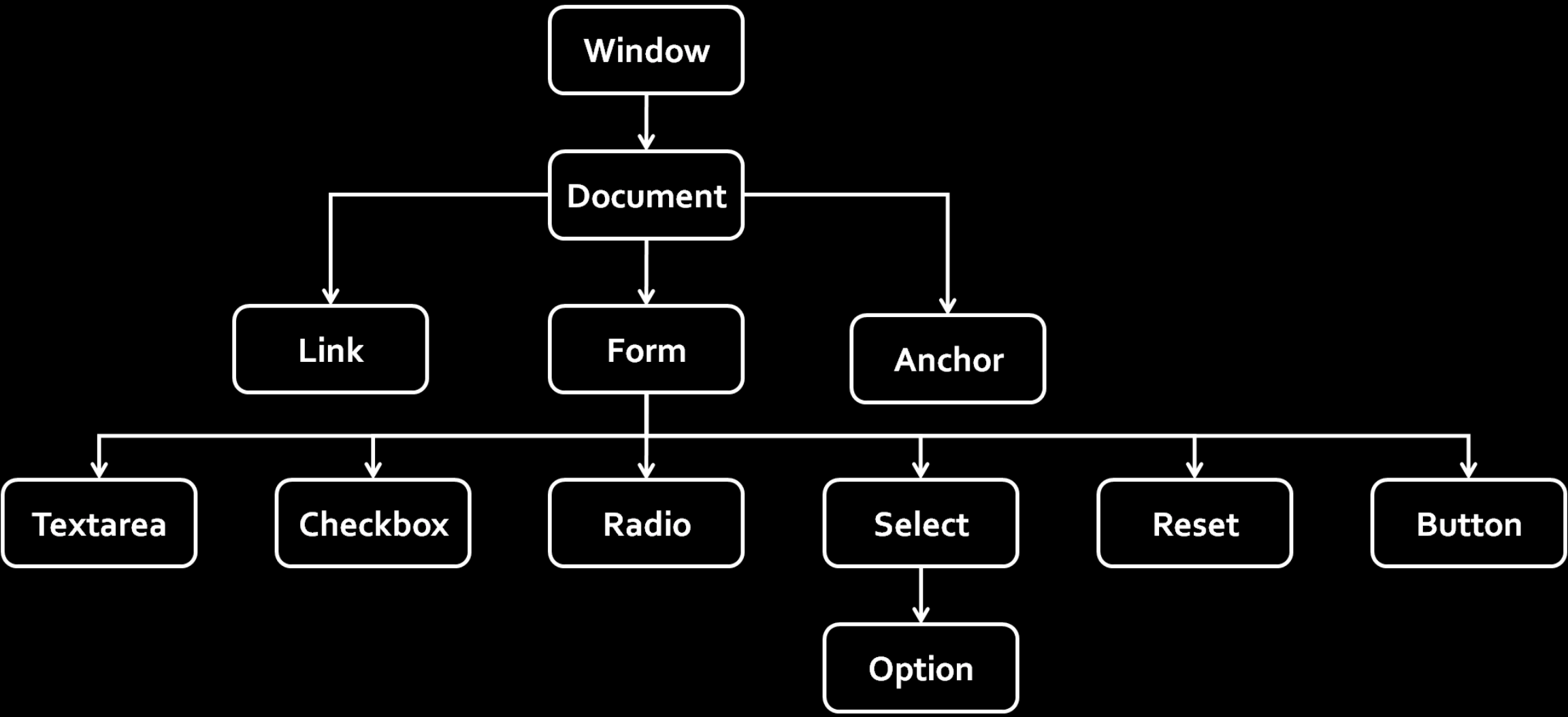
Example 23

Document Object Model

- Document Object Model(DOM) is a programming interface for HTML.
- It specifies the logical structure of the webpage and the way it can be manipulated.
- Browser creates a Document Object Model of the page loaded.

Example 24

Properties of DOM



- **Window Object: Top of hierarchy**
- **Document object: HTML loaded in a window**
- **Form Object: Represented by *form* tags.**
- **Link Objects: Represented by *link* tags.**
- **Anchor Objects: Represented by *a href* tags.**
- **Form Control Elements: HTML form elements**

Methods of DOM

- wite(“string”): write on document
- getElementById(): return element with given ID
- getElementByName(): return element with given Name
- getElementsByTagName(): return element with HTML tag name

Manipulating DOM

- innerHTML
- textContent
- value

Example 25
Example 26
Example 27

Unit 3

Section 3.2

Topics

- Executing Deferred Script
- Objects
  - Document Object Model

Web Programming

Dr. D. Natarajasivan
Lecturer
TNPT

Topics

- Objects
  - Document Object Model
  - **Predefined Object**
    - **Array Object**
    - **History Object**
    - **Location Object**

Array Object

- Array object is used to store multiple values in a single variable
- Store data of same type

Syntax

```
var array_name = new Array(values);
                    (or)
var array_name = [values]
```

Example 28

Array Methods

- length: return array length

  Syntax

  len = array.length;

- concat(): combine two array

  Syntax

  new_array = array.concat(array1);

Example 29

- **filter(): filter the element that pass the test.**

  Syntax

  ```
  new_array = array.filter(e=>(condition));
  ```

- **forEach(): Calls a function for each element in the array.**

  Syntax

  ```
  array.forEach(e=>function(e));
  ```

Example 30

- **join(): Join all elements into a String**

  Syntax

  string = array.join(separator);

- **sort(): Sorts the elements of an array.**

  Syntax

  new_array = array.sort();

Example 31

- **reverse(): reverse element of an array**

  Syntax

  new_array = array.reverse();

- **indexOf(): return first index of the element searched**

  Syntax

  array.indexOf(search);

Example 32

- **push(): add at the last of array**

  Syntax

  array.push(value);

- **pop(): remove the last element from the array**
  - **Syntax**
  - **array.pop();**

Example 33

History Object

- This represents the pages visited in a tab where the current page is loaded.
- This is a property of the window object

Syntax

window.history

(or)

history

Properties and Methods

- length: return the number of pages visited in the tab
- forward(): load next page
- backward(): load Previous Page
- go(): load the given page number

Example 34

Location Object

- It represent the current URL of the document being displayed.
- This object can be used to get the different parts of the URL

Location Object Properties

- host: access hostname and port number
- hostname: access hostname of URL
- href: access entire URL
- pathname: access path name of URL
- protocol: access protocol of URL

Example 35

Location Object Methods

- assign(): load new document
- reload(): reload current document
- replace(): replace current document with new document

Example 36

Unit 3

Section 3.2

Topics

- Array Object
- History Object
- Location Object

Web Programming

Dr. D. Natarajasivan
Lecturer
TNPT

Topics

- Dialog Boxes
  - Alert Dialog Box
  - Confirmation Dialog Box
  - Prompt Dialog Box

Alert Dialog Box

- Simple dialog box used to show a short message.
- It contains OK button to close the box and continue

Syntax

```
alert("message");
```

Example 37

Confirmation Dialog Box

- Used to get opinion from the user
- It has OK and CANCEL buttons.
- OK return true, CANCEL return false

Syntax

```
var  res = confirm("message");
```

Example 38

Prompt Dialog Box

- Used to get input from the user.
- It has OK and CANCEL
- OK return entered text, CANCEL return null

Syntax

```
var val = prompt("message");
```

Example 39

Section 3.3

- Topics
  - Events
  - Event Handlers
  - JavaScript Events

Events

- An event is a signal that something has happened.
- This can be used to execute JavaScript code.
- Every HTML element contains a set of events

Event Handler

- It is the function that runs in case of an event.
- Two types of event handler
  - Interactive event handler
  - Non-interactive event handler

JavaScript Events

- Window events
- Mouse events
- Keyboard events
- Form events

Window events

- onload – called when browser finishes loading the page
- onunload – called when the webpage is unloaded
- resize – called when the browser is resized

Example 40

Mouse Event

- onclick – on mouse click
- onmouseover – mouse over an element
- onmousedown – mouse button is down
- onmouseup – mouse button is up

Example 41

Keyboard Events

- onkeydown – when key is pressed
- onkeypress – when key is pressed and released
- onkeyup – when pressed key is released
- event.key – get the pressed key

Example 42

Form events

- onchange – when an element changes
- onselect – when an element is selected

Example 43

- onfocus – when an element is focused
- onblur – when an element is loses focus

Example 44

- onsubmit – when the form is submitted
- onreset – when form is reset

Example 45

- onerror – when an error is occurred. Used in image, link and script tags

Example 46

Unit 3

Section 3.2

Topics

- Dialog Boxes
  - Alert Box
  - Confirmation Box
  - Prompt Box
- JavaScript Events

Web Programming

Dr. D. Natarajasivan
Lecturer
TNPT

Topics

- **Forms**
  - **Forms Array**
  - **Form Element Property**

Forms Array

- Forms in a document can be accessed using JavaScript
- Using document object and “forms” property
- It is used to access all the forms in a document as an array

Syntax

document.forms

- It has the properties
  - length: gives number of forms
  - id: return the form id value
  - name: return the form name
  - elements: gets all the elements in the form

- **Methods**
  - **[index]: Return form using index value**
  - **item(index): Return form using index value**
  - **namedItem(id): Return form with given id**

Example 47

Form Element Property

- This property returns all the elements in the selected from
- It can be used to get values of individual elements in array format

- **Property**
  - value: get the value present in the element
  - name: get the name of the element
  - id: get the ID of the element
  - Use element name to select a particular element

Example 48

Section 3.4

- Client side Image Maps
- Server side Image Maps
- Status Bar
- Cookies

Client Side Image Maps

- These are clickable images.
- Different areas of an image are mapped to a particular link
- Tags used
  - img
  - map
  - area

- **img tag**
  - usemap attribute is used to give the map name

    <img src="img_src" usemap="#map_name">
- **map tag**
  - name attribute is used to give the name of the mpa

    <map name="map_name">

    </map

- area tag
  - shape : poly, rect, circle
  - coords
  - href
  - target
  - [events]

Example 49

Server Side Image Maps

- Image Map is processed in the server
- ismap attribute is added to the img tag
- This sends the x, y coordinate of the image click to the server
- CGI script or any other script is used to manipulate it.

Example 50

Status Bar

- Status bar can be modified using JavaScript
- It is a property of the window object

  Syntax

  window.status="text"

Example 51

Cookies

- Data stored in small text files
- Cookies can be created, read and deleted using JavaScript

  Syntax

  document.cookie = “value”;

- **Values**
  - Name=value: Key-value pair
  - expires: set cookie expiry date (delete cookie)
  - path: web page that set the cookie
  - domain: domain name of site
  - secure: if “secure” is given only a secure server can access the cookie

Example 52

Unit 3

Section 3.3

Topics

- Form Array
- Form Element Property

Section 3.4

Topics

- Client Side Image Maps
- Server Side Image Maps
- Status Bar & Cookies

Web Programming

Dr. D. Natarajasivan
Lecturer
TNPT

Topics

- Live Connect

Live Connect

- Mechanism that allows JavaScript and Java to work together.
- This feature was first introduced in Netscape browser
  - Java Console
  - JavaScript to Java
  - Java to JavaScript

Java Console

- It is a Navigator window that is used to display Java messages.
- This can be enabled by choosing “Show Java Console” from Options menu.

JavaScript to Java

- JavaScript can communicate with Java using
  - Call Java methods directly
  - Control Java applets
  - Control Java plug-ins

Java to JavaScript

- To access JavaScript methods from Java applet, import Netscape javascript package.

```
import netscape.javascript.*;
```

- JSObject class: used to access JavaScript elements
- JSException object: used to handle script errors