JavaScript

Contents

- Javascript introduction
- Features
- Embedding JavaScript
- JavaScript Data Types :
  - Integer, String,Boolean
- JavaScript Operators:
  - Arithmetic,
  - Relational,
  - Logical,
  - Conditonal,
  - Assignment,
  - Type Of
- Control Flow Statements : Branching , Looping ,Loop control
- Functions : User Defined , Built In (Math,String,Date)

Cont...

- .Event Handling :
  - Mouse Event
  - Keyboard Event
  - Drag and Drop Event
  - Media Event
  - Touch Event
  - Clipboard Event
- Drag and Drop Event
- Dialog Boxes : Alert Confrim, Propmpt
- DOM :
- User Defined Objects
  - DOM : Windows , Document , Location , History
- Working with Array
- 12 . Forms with Javascript

Introduction

- JavaScript is a lightweight, interpreted programming language.
- Designed for creating network-centric applications.
- Complementary to and integrated with Java.
- Complementary to and integrated with HTML.
- Open and cross-platform

Advantages

- **Less server interaction** – You can validate user input before sending the page off to the server. This saves server traffic, which means less load on your server.
- **Immediate feedback to the visitors** – They don't have to wait for a page reload to see if they have forgotten to enter something.
- **Increased interactivity** – You can create interfaces that react when the user hovers over them with a mouse or activates them via the keyboard.
- **Richer interfaces** – You can use JavaScript to include such items as drag-and-drop components and sliders to give a Rich Interface to your site visitors.

Placement of JavaScript in HTML

- There is a flexibility given to include JavaScript code anywhere in an HTML document. However the most preferred ways to include JavaScript in an HTML file are as follows −
- Script in <head>...</head> section.
- Script in <body>...</body> section.
- Script in <body>...</body> and <head>...</head> sections.
- Script in an external file and then include in <head>...</head> section.

JavaScript DataTypes

- JavaScript allows you to work with three primitive data types
  - **Numbers,** eg. 123, 120.50 etc.
  - **Strings** of text e.g. "This text string" etc.
  - **Boolean** e.g. true or false.
- JavaScript also defines two trivial data types, **null** and **undefined,** each of which defines only a single value.

JavaScript Variables

- Variables are declared with the **var** keyword
- Storing a value in a variable is called **variable initialization**. You can do variable initialization at the time of variable creation or at a later point in time when you need that variable.
- The scope of a variable is the region of your program in which it is defined. JavaScript variables have only two scopes.
  - **Global Variables** – A global variable has global scope which means it can be defined anywhere in your JavaScript code.
  - **Local Variables** – A local variable will be visible only within a function where it is defined. Function parameters are always local to that function.

JavaScript Operators

- JavaScript supports the following types of operators.
- Arithmetic Operators
- Comparison Operators
- Logical (or Relational) Operators
- Assignment Operators
- Conditional (or ternary) Operators
- typeof Operator

Control Flow Statements

- Branching
  - if statement
  - if…else statement
  - if…else if… statement.
  - Switch case
- Looping
  - While
  - For loop
  - Do…while
  - For…in

Looping Statements

- The for/in statement loops through the properties of an object.
- The block of code inside the loop will be executed once for each property.
- JavaScript supports different kinds of loops:
- **for** - loops through a block of code a number of times
- **for/in** - loops through the properties of an object
- **while** - loops through a block of code while a specified condition is true
- **do/while** - loops through a block of code once, and then repeats the loop while a specified condition is true

Loop Control

- Break
- Continue

JavaScript Function

- A function is a group of reusable code which can be called anywhere in your program.
- This eliminates the need of writing the same code again and again.
- It helps programmers in writing modular codes.
- Functions allow a programmer to divide a big program into a number of small and manageable functions.

Function Syntax

- Syntax:

```
<script type="text/javascript">
    function functionname(parameter-list)
    {
      statements
    }
</script>
```

Event

- JavaScript's interaction with HTML is handled through events that occur when the user or the browser manipulates a page.
- examples include events like pressing any key, closing a window, resizing a window, etc.
- HTML DOM events allow JavaScript to register different event handlers on elements in an HTML document.
- Events are normally used in combination with functions, and the function will not be executed before the event occurs

Touch Events

Event	Description	DOM
ontouchcancel	The event occurs when the touch is interrupted	
ontouchend	The event occurs when a finger is removed from a touch screen	
ontouchmove	The event occurs when a finger is dragged across the screen	
ontouchstart	The event occurs when a finger is placed on a touch screen	

Clipboard Events

Event	Description	DOM
oncopy	The event occurs when the user copies the content of an element	
oncut	The event occurs when the user cuts the content of an element	
onpaste	The event occurs when the user pastes some content in an element	

Form Event

Event	Description	DOM
onblur	The event occurs when an element loses focus	2
onchange	The event occurs when the content of a form element, the selection, or the checked state have changed (for <input>, <select>, and <textarea>)	2
onfocus	The event occurs when an element gets focus	2
onfocusin	The event occurs when an element is about to get focus	2
onfocusout	The event occurs when an element is about to lose focus	2
oninput	The event occurs when an element gets user input	3
oninvalid	The event occurs when an element is invalid	3
onreset	The event occurs when a form is reset	2
onsearch	The event occurs when the user writes something in a search field (for <input="search">)	3
onselect	The event occurs after the user selects some text (for <input> and <textarea>)	2
onsubmit	The event occurs when a form is submitted	2

Keyboard Events

Event	Description	DOM
onkeydown	The event occurs when the user is pressing a key	2
onkeypress	The event occurs when the user presses a key	2
onkeyup	The event occurs when the user releases a key	2

Mouse Events

Event	Description	DOM
onclick	The event occurs when the user clicks on an element	2
oncontextmenu	The event occurs when the user right-clicks on an element to open a context menu	3
ondblclick	The event occurs when the user double-clicks on an element	2
onmousedown	The event occurs when the user presses a mouse button over an element	2
onmouseenter	The event occurs when the pointer is moved onto an element	2
onmouseleave	The event occurs when the pointer is moved out of an element	2
onmousemove	The event occurs when the pointer is moving while it is over an element	2
onmouseover	The event occurs when the pointer is moved onto an element, or onto one of its children	2
onmouseout	The event occurs when a user moves the mouse pointer out of an element, or out of one of its children	2
onmouseup	The event occurs when a user releases a mouse button over an element	2

Drag Events

Event	Description	DOM
ondrag	The event occurs when an element is being dragged	3
ondragend	The event occurs when the user has finished dragging an element	3
ondragenter	The event occurs when the dragged element enters the drop target	3
ondragleave	The event occurs when the dragged element leaves the drop target	3
ondragover	The event occurs when the dragged element is over the drop target	3
ondragstart	The event occurs when the user starts to drag an element	3
ondrop	The event occurs when the dragged element is dropped on the drop target	3

Dialog boxes

- Alert
- An alert dialog box is mostly used to give a warning message to the users.
- Alert box gives only one button "OK" to select and proceed.

Cont...

- Prompt
- The prompt dialog box is very useful when you want to pop-up a text box to get user input. Thus, it enables you to interact with the user. The user needs to fill in the field and then click OK.
- This dialog box has two buttons: **OK** and **Cancel**. If the user clicks the OK button,

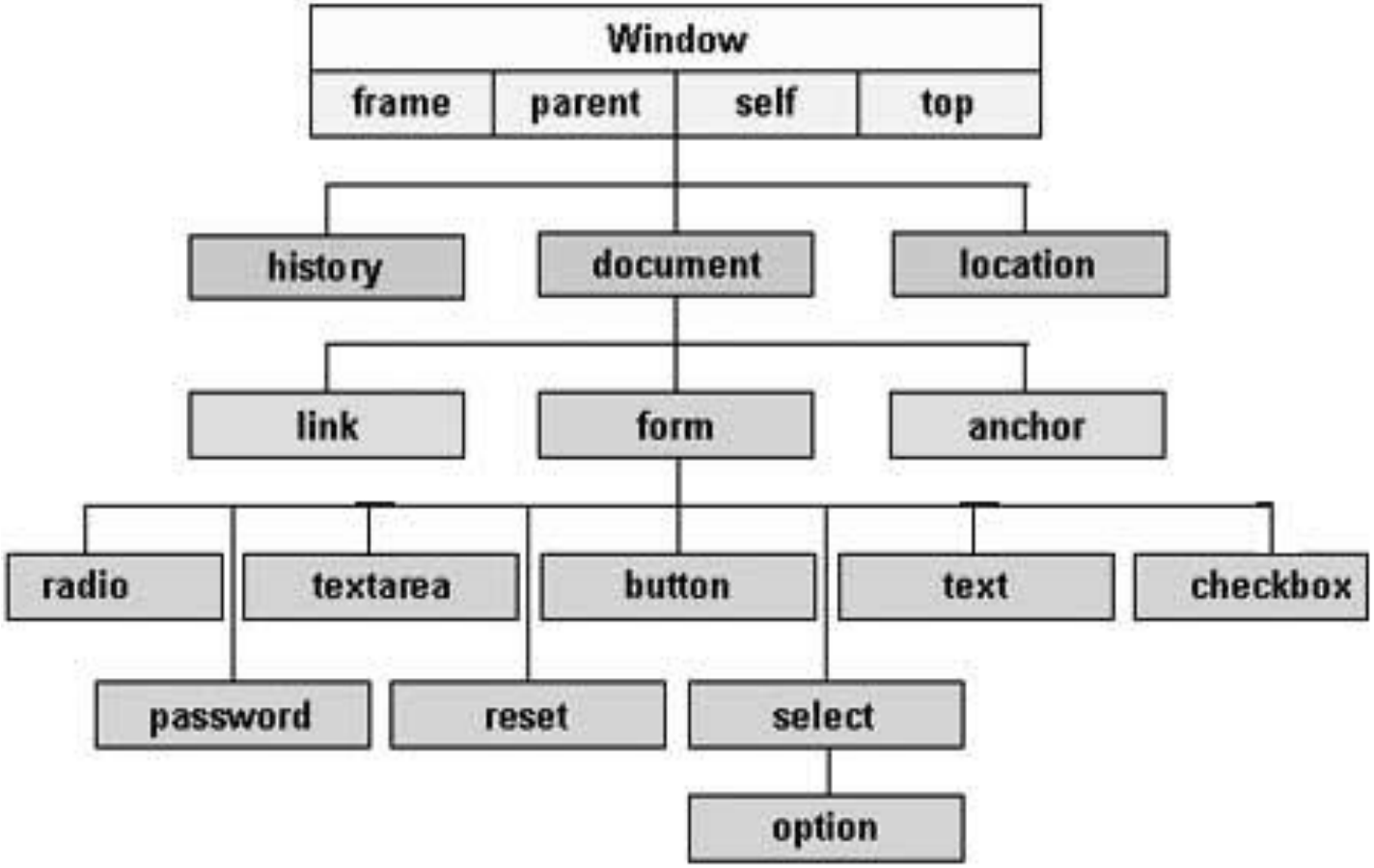
Comfirm

- A confirmation dialog box is mostly used to take user's consent on any option. It displays a dialog box with two buttons: **Cancel**
- If the user clicks on the OK button, the window method **confirm()** will return true. If the user clicks on the Cancel button, then **confirm()** returns false.

Use of Void Keyword

- **void** is an important keyword in JavaScript which can be used as a unary operator that appears before its single operand, which may be of any type. This operator specifies an expression to be evaluated without returning a value.
- Another use of **void** is to purposely generate the **undefined**

Document object model



Cont...

- **Window object** – Top of the hierarchy. It is the outmost element of the object hierarchy.
- **Document object** – Each HTML document that gets loaded into a window becomes a document object. The document contains the contents of the page.
- **Form object** – Everything enclosed in the <form>…</form> tags sets the form object.
- **Form control elements** – The form object contains all the elements defined for that object such as text fields, buttons, radio buttons, and checkboxes.

Window

- The **window** object is supported by all browsers. It represents the browser's window.
- All global JavaScript objects, functions, and variables automatically become members of the window object.
- Global variables are properties of the window object.
- Global functions are methods of the window object.
- Even the document object (of the HTML DOM) is a property of the window object:

Location

- The **window.location** object can be written without the window prefix.
- Some examples:
- window.location.href returns the href (URL) of the current page
- window.location.hostname returns the domain name of the web host
- window.location.pathname returns the path and filename of the current page
- window.location.protocol returns the web protocol used (http: or https:)
- window.location.assign loads a new document

History

- The **window.history** object can be written without the window prefix.
- To protect the privacy of the users, there are limitations to how JavaScript can access this object.
- Some methods:
- history.back() - same as clicking back in the browser
- history.forward() - same as clicking forward in the browser

Javascript Objects

- Objects are variables too. But objects can contain many values.
- The values are written as **name:value** pairs (name and value separated by a colon).
- JavaScript objects are containers for **named values**.
- **Object Properties**
  - The name:values pairs (in JavaScript objects) are called **properties**.
- **Object Methods:**
  - Methods are **actions** that can be performed on objects.
  - Methods are stored in properties as **function definitions**.

Accessing Object Properties

- *objectName.propertyName*

OR

- *objectName["propertyName"]*
- var student = {
  Name:“Rohan",
  age:15
  };
- Student.name
- Student[“name”]

Accessing Object Methods

- *objectName.methodName()*
- *Example:*

```
<script>
var student = {
    Name: "Rohan",
     Age:15,
     getName : function() {
       return this.Name + " " + this.Age;
     }
};

document.getElementById("demo").innerHTML = student.getName();
</script>
```

Methods

- String handling
- Math
- Array
- Date

Math Object

- Math.round()
  - Math.round(x) returns the value of x rounded to its nearest integer:
- Math.pow()
  - Math.pow(x, y) returns the value of x to the power of y:
- Math.sqrt()
  - Math.sqrt(x) returns the square root of x:
- Math.abs()
  - Math.abs(x) returns the absolute (positive) value of x:
- Math.ceil()
  - Math.ceil(x) returns the value of x rounded **up** to its nearest integer:
- Math.floor()
  - Math.floor(x) returns the value of x rounded **down** to its nearest integer:

Cont...

- Math.min() and Math.max()
  - Math.min() and Math.max() can be used to find the lowest or highest value in a list of arguments:
- Math.random()
  - Math.random() returns a random number between 0 (inclusive), and 1 (exclusive):
  - To Return Random integer :
    - Math.random() used with Math.floor() can be used to return random integers.

Date Object

- To get the current date :
- Date();
- To specify the exact date format:
- Using new Date(**date string**), creates a new date object from the **specified date and time**:

Date Get Methods

Get methods are used for getting a part of a date. Here are the most common (alphabetically):

Method	Description
getDate()	Get the day as a number (1-31)
getDay()	Get the weekday as a number (0-6)
getFullYear()	Get the four digit year (yyyy)
getHours()	Get the hour (0-23)
getMilliseconds()	Get the milliseconds (0-999)
getMinutes()	Get the minutes (0-59)
getMonth()	Get the month (0-11)
getSeconds()	Get the seconds (0-59)
Time()	Get the time (milliseconds since January 1, 1970)

String Methods

- toUpperCase()
- Trim()
- Substring()
- Split()
- Search()
- Replace()
- Concat()
- charAt()

Array

- The JavaScript Array object is a global object that is used in the construction of arrays;
- var array_Name= [‘v1', ‘v2'];
- Methods:
- Join()
- Sort()
- Reverse()