**Web Designing**

**MODULE: 4 (JavaScript Basic & DOM)**

**Q1. What is JavaScript?**

**A1.** JavaScript is the **programming language** for the Web which was developed in 1995 by Brendan Eich while he was an engineer at Netscape. It can update and change both HTML & CSS. It is used to calculate, manipulate & validate data. It is also a **scripting language** that enables us to create dynamically updating content, control multimedia, animate images, etc.

**Q2. What is the use of the isNaN function?**

**A2.** The **isNaN()** function is used to determine whether a value is NaN **(Not-a-Number)** or not. It returns true if the value is NaN, and false otherwise. It’s commonly used to validate whether user input is a valid number.

**Q3. What is negative Infinity?**

**A3. NEGATIVE\_INFINITY** is a special number value that is returned when an arithmetic operation or mathematical function generates a negative value greater than the largest representable number in JavaScript. It’s typically the result of mathematical operations that lead to values beyond the representable range.

**Q4. Which company developed JavaScript?**

**A4.** JavaScript was created at **Netscape Communications** by **Brendan Eich** in **1995**. Netscape and Eich designed JavaScript as a scripting language for use with the company’s flagship web browser, Netscape Navigator.

**Q5. What are undeclared and undefined variables?**

**A5. Undeclared variables** are those that have been referenced in the code without being declared using the var, let, or const keywords. It occurs when we try to access any variable that is not initialized or declared earlier.

**Undefined variables** are those that have been declared but have not been assigned any value, or they are declared but not accessible in the current scope. Undefined is not a keyword.

**Q6. Write the code to add new elements dynamically.**

**A6.** // Example: Adding a new paragraph dynamically to a div with id “container”

// Get container element by its ID

var container = document.getElementByID(“container”);

// Create a new paragraph element

var newParagraph = document.getElementByID(“p”);

// Create a text node with the content for the paragraph

var textnode = document.createTextNode(“This is new Paragraph”);

// Append the text node to the new paragraph

newParagraph.appendChild(textNode);

// Append the new paragraph to the container

container.appendChild(newParagraph);

**Q7. What is the difference between ViewState and SessionState?**

**A7.**

| **ViewState** | **SessionState** |
| --- | --- |
| Maintained at page level only. | Maintained at session level. |
| View State can only be visible from a single page and not multiple pages. | Session State value availability is across all pages available in a user session. |
| It will retain values in the event of a postback operation occurring. | In the session state, user data remains on the server. Data is available to the user until the browser is closed or there is a session expiration. |
| Information is stored on the client side only. | Information is stored on the server. |
| ViewState values are lost/cleared when a new page is loaded. | SessionState can be cleared by the user or in case of timeouts. |
| It can be used to store information that you wish to access from the same web page. | It can be used to store information that you wish to access on different web pages. |

**Q8. What is === operator?**

**A8.** In JavaScript, === is a comparison operator that checks the equality of two values without performing any type of conversion. If the values being compared have different data types, === will return false.

Eg.: console.log(7===7); *//true*

console.log(7===’7’); *//false*

console.log(true===1); *//false*

console.log(false===0); *//false*

**Q9. How can the style/class of an element be changed?**

**A9.** We can change, add, or remove any CSS property from an HTML element on the occurrence of any event with the help of JavaScript. There are two ways to change the style/class of an element:

1. style.property

Syntax: document.getElementById(“id”).style.property=new\_style;

1. Changing the class itself

Syntax: document.getElementById(“id”).classList

**Q10. How to read and write a file using JavaScript?**

**A10.** In a browser environment, JavaScript doesn’t have direct access to the file system for security reasons. However, you can read files using the File Reader API for files selected by the user through **<input type=”file”>**. Writing files is typically done using server JavaScript in environments like Node.js, where you have access to file system APIs.

**Q11. What are all the looping structures in JavaScript?**

**A11.** JavaScript loops are essential for efficiently handling repetitive tasks. They execute a block of code repeatedly as long as a specified condition remains true. These loops are powerful tools for automating tasks and streamlining your code.

There are four types of loops in JavaScript:

1. **For Loop**

JavaScript for loop iterates the elements for a fixed number of times. It should be used if some iterations are known.

**Syntax**: for(initialization,condition,increment)

{

code to be executed;

};

1. **While Loop**

JavaScript while loop iterates the elements an infinite number of times. It should be used if some iterations have yet to be discovered.

**Syntax**:while(condition)

{

Code to be executed;

};

1. **Do-While Loop**

JavaScript do-while loop iterates the elements an infinite number of times like a while loop. But, code is executed at least once whether the condition is true or false.

**Syntax**: do

{

Code to be executed;

}while(condition);

1. **For-in Loop**

JavaScript for-in loop is used to iterate the properties of an object.

**Q12. How can you convert the string of any base to an integer in**

**JavaScript?**

**A12.** To convert a string into an integer we can use either the **parseInt()** method which accepts the string and radix parameter and converts it into an integer or we can use the **Number()** method which is used to convert primitive data to a number, if it is not convertible it returns NaN.

**Q13. What is the function of the delete operator?**

**A13.** The delete operator in JavaScript is used to remove a property from an object. It works for both properties owned by the object and those inherited from prototypes. When used on an array item, it creates a ‘hole’ in the array. This operator returns true if it removes a property. While deleting an object property that doesn’t exist will return true but it will not affect the object. However, trying to delete a variable or a function will return a false.

Syntax: delete object

//or

delete object.property

//or

delete object[‘property’]

**Q14. What are all the types of Pop-up boxes available in JavaScript?**

**A14.** JavaScript has three kinds of popup boxes:

1. **Alert box**

An alert box is often used if you want to make sure information comes through to the user. When an alert box pops up, the user will have to click ‘OK’ to proceed.

**Syntax**: window.alert(“some text”);

1. **Confirm Box**

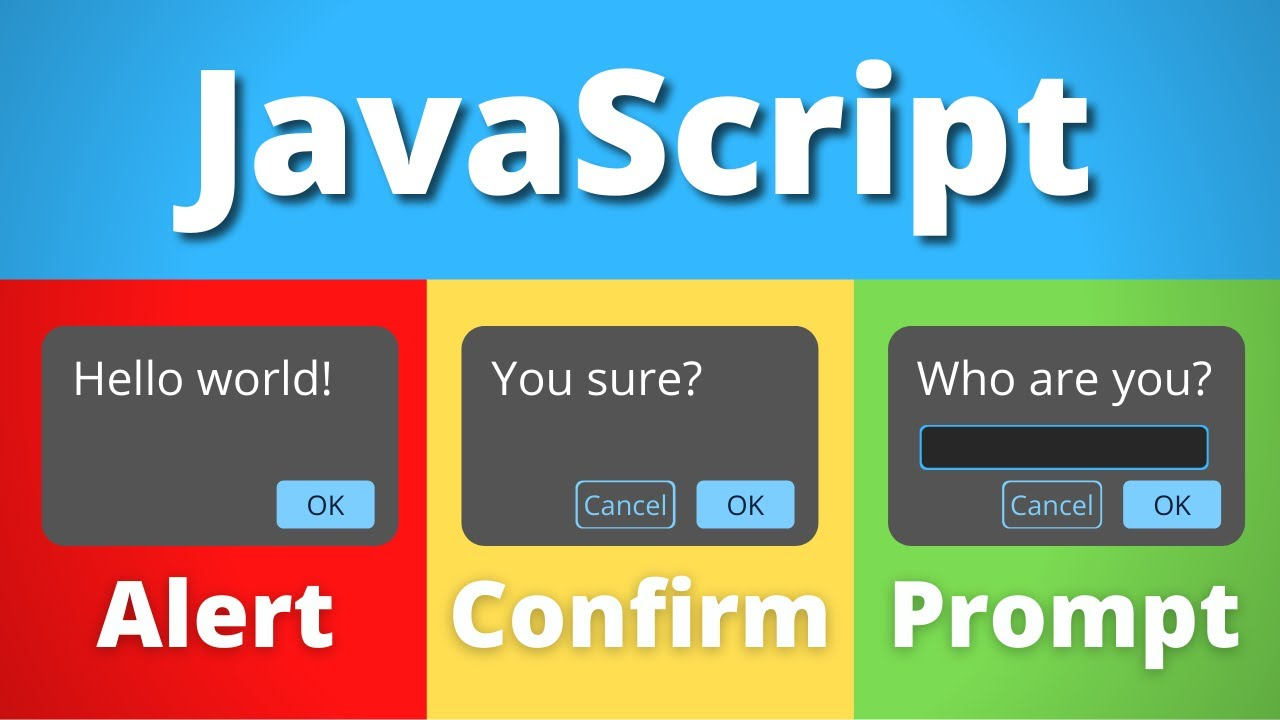
A confirm box is often used if you want the user to verify or accept something. When a confirm box pops up, the user will have to click either ‘OK’ or ‘Cancel’ to proceed. If the user clicks ‘OK’, the box returns true. If the user clicks ‘Cancel’, the box returns false.

**Syntax**: window.confirm(“some text”);

1. **Prompt Box**

A prompt box is often used if you want the user to input a value before entering a page. When a prompt box pops up, the user will have to click either ‘OK’ or ‘Cancel’ to proceed after entering an input value. If the user clicks ‘OK’, the box returns true. If the user clicks ‘Cancel’, the box returns null.

**Syntax**: window.prompt(“some text”, “default Text”);

****

**Q15. What is the use of Void (0)?**

**A15.** JavaScript void(0) means returning undefined(void) as a primitive value. It prevents the browser from taking any action when a link is clicked. It’s commonly used as the href attribute of a link to prevent the page from navigating to a new URL.

**Q16. How can a page be forced to load another page in JavaScript?**

**A16.** To force a page to load another page in JavaScript, you can use **window.location** object. You can either use the **window.location.href** property which is the most straightforward way as it changes the URL of the current page to the specified URL or uses methods like **window.location.replace()** which replaces the current document with the new one.

**Q17. What are the disadvantages of using innerHTML in JavaScript?**

**A17.** Disadvantages of innerHTML in JavaScript:

* The use of innerHTML is very slow
* Preserves event handlers attached to any DOM elements
* Content is replaced everywhere
* Appending to inner HTML is not supported
* Old content replaced issues
* Can break the document
* Can also be used for Cross-site Scripting (XSS)