web desining

TOPSTECHNOLOGIES
Training | Outsourcing | Placement | Study Abroad

Course: Frontend Assignment

MODULE : 4 [JAVASCRIPT BASIC & DOM]

Name: Yashrajsinh Parmar

QUE:01What is JavaScript?

Ans)

JavaScript is a scripting language for creating dynamic web page content. It creates elements for improving site visitors' interaction with web pages, such as dropdown menus, animated graphics, and dynamic background colors.

JavaScript is a lightweight programming language that web developers commonly use to create more dynamic interactions when developing web pages, applications, servers, and or even games.

QUE:02→ What is the use of isNaN function?

Ans)

The JavaScript **isNaN()** Function is used to check whether a given value is an illegal number or not. It returns true if the value is a NaN else returns false. It is different from the Number.isNaN() Method.

Syntax:

```
isNaN( value )
```

- **Parameter Values:** This method accepts a single parameter as mentioned above and described below:

- **value:** It is a required value passed in the isNaN() function.
-
- **Return Value:** It returns a Boolean value i.e. returns true if the value is NaN else returns false.

QUE:03→ What is negative Infinity?

Ans)

- The **negative infinity** in JavaScript is a constant value that is used to represent a value that is the lowest available. This means that no other number is lesser than this value. It can be generated using a self-made function or by an arithmetic operation.

- **Negative infinity** is different from mathematical infinity in the following ways:

- Negative infinity results in **-0**(different from 0) when divided by any other number.
- When divided by itself or positive infinity, negative infinity return NaN
- Negative infinity, when divided by any positive number (apart from positive infinity) is negative infinity.
- Negative infinity, divided by any negative number (apart from negative infinity) is positive infinity.
- If we multiply negative infinity with NaN, we will get NaN as a result.

- The product of 0 and negative infinity is Nan.
- The product of two negative infinities is always a positive infinity.
- The product of both positive and negative infinity is always negative infinity.

QUE:04 → **Which company developed JavaScript?**

Ans)

History of JavaScript

- JavaScript language comes from the times when early web browsers were being developed. Netscape Communications company in 1994 created Netscape Navigator that became the most popular web browser in the 90s.

- Company's board quickly realized that browsers should allow create more dynamic websites and do some activities that do server-side languages, like input validation.

- First Netscape Communications cooperate with Sun Microsystems to use in Netscape Navigator Sun's programming language Java. Then they wanted adopting and embedding a existing programming language like Scheme, Perl or Python.

- Eventually they decided to create scripting language that would complement Java and has a similar syntax.

- In 1995 Netscape Communications employed Brendan Eich to develop scripting language for web browser. Eich prepared it in a very short time.

- First version of new language had Mocha name, whereas official version used in Netscape Navigator 2 beta version was called LiveScript. In the same 1995 year new developed scripting language was renamed to JavaScript and used in next beta version of Netscape Navigator 2. LiveScript followed a lot of Java features.

- This, but above all the desire to use the growing popularity of Java to call positive associations with a new language where reasons that it was finally called JavaScript. Also implementation of the language for server-side was introduced.

QUE:05 → What are undeclared and undefined variables?

Undefined: It occurs when a variable has been declared but has not been assigned any value. Undefined is not a keyword.

Undeclared: It occurs when we try to access any variable that is not initialized or declared earlier using the *var* or *const keyword*. If we use *'typeof'* operator to get the value of an undeclared variable, we will face the *runtime error* with the return value as **"undefined"**. The scope of the undeclared variables is always global.

Example 01: Undefined Variable

To see an example of an undefined value, declare a variable but do not assign it a value:

```
var dog;

console.log(dog);
```

Output:

```
Undefined
```

This is what is meant by an undefined variable in JavaScript. It's been declared but doesn't hold value.

Example 02: Undeclared Variable

An example of an undeclared variable is when there is no such variable in the program.

For example, let's try to print a variable called **cat** without having such a variable in the program:

```
console.log(cat);
```

Output:

```
ReferenceError: cat is not defined
```

QUE:06 **Write the code for adding new elements dynamically?**

Ans)

Javascript is a very important language when it comes to learning how the browser works. Often there are times we would like to add dynamic elements/content to our web pages. This post deals with all of that.

Creation of new element: New elements can be created in JS by using the **createElement()** method.

Syntax: document. createElement("tagName");

QUE:07 → **What is the difference between ViewState and SessionState?**

Ans)

ViewState	SessionState
1. Maintained at page level only.	1. Maintained at session level.
2. View state can only be visible from a single page and not multiple pages.	2. Session state value availability is across all pages available in a user session.

ViewState	SessionState
3. It will retain values in the event of a postback operation occurring.	3. In session state, user data remains in the server. Data is available to user until the browser is closed or there is session expiration.
4. Information is stored on the client's end only.	4. Information is stored on the server.
5. used to allow the persistence of page-instance-specific data.	5. used for the persistence of user-specific data on the server's end.
6. ViewState values are lost/cleared when new page is loaded.	6. SessionState can be cleared by programmer or user or in case of timeouts.

QUE:08 → What is === operator?

Ans)

- The strict equality (===) operator checks whether its two operands are equal, returning a Boolean result.
- Unlike the equality operator, the strict equality operator always considers operands of different types to be different.
-
- Syntax: x === y Description The strict equality operators (=== and !==) provide the IsStrictlyEqual semantic.
-
- The **strict equality (===)** operator checks whether its two operands are equal, returning a Boolean result. Unlike the equality operator, the strict equality operator always considers operands of different types to be different.

QUE:09 → **How can the style/class of an element be changed?**

Ans)

getElementById() **method is used to return the element in the document with the "id" attribute and the "className" attribute can be used to change/append the class of the element.**

QUE:10 → **How to read and write a file using JavaScript?**

Ans)

Files can be read and written by using java script functions – fopen(),fread() and fwrite().

The function fopen() takes two parameters – 1. Path and 2. Mode (0 for reading and 3 for writing).

The fopen() function returns -1, if the file is successfully opened.

Example:

```
file=fopen(getScriptPath(),0);
```

The function fread() is used for reading the file content.

Example:

```
str = fread(file,flength(file) ;
```

The function fwrite() is used to write the contents to the file.

Example:

```
file = fopen("c:\MyFile.txt", 3);// opens the file for writing
fwrite(file, str);// str is the content that is to be written into the
file.
```

read and write a file using javascript

There are two ways to do it:

1. Using JavaScript extensions (runs from JavaScript Editor), or

2. Using a web page and ActiveX objects (Internet Explorer only)

QUE:11 → **What are all the looping structures in JavaScript?**

Ans)

The JavaScript loops are used *to iterate the piece of code* using for, while, do while or for-in loops. It makes the code compact. It is mostly used in array.

There are four types of loops in JavaScript.

- for loop
- while loop
- do-while loop
- for-in loop

QUE:12 **How can you convert the string of any base to an integer in JavaScript?**

Ans)

- How can you convert the string of any base to integer in JavaScript? The parseInt () function is used to convert numbers between different bases. It takes the string to be converted as its first parameter, and the second parameter is the base of the given string.

QUE:13 → **What is the function of the delete operator?**

Ans)

Delete is an ***operator*** that is used to **destroy array** and **non-array**(pointer) **objects** which are created by new expression.

- Delete can be used by either using ***Delete operator*** or ***Delete [ ] operator***

- New operator is used for dynamic memory allocation which puts variables on heap memory.
- Which means Delete operator deallocates memory from heap.
- Pointer to object is not destroyed, value or memory block pointed by pointer is destroyed.
- The delete operator has **void** return type does not return a value.

QUE:14→ **What are all the types of Pop up boxes available in JavaScript?**

Ans)

JavaScript has three kind of popup boxes: Alert box, Confirm box, and Prompt box.

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("sometext");
```

The `window.alert()` method can be written without the window prefix.

Example

```
alert("I am an alert box!");
```

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("sometext");
```

The `window.confirm()` method can be written without the window prefix.

Example

```
if (confirm("Press a button!")) {
  txt = "You pressed OK!";
} else {
  txt = "You pressed Cancel!";
}
```

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 the input value. If the user clicks "Cancel" the box returns null.

Syntax

```
window.prompt("sometext","defaultText");
```

The `window.prompt()` method can be written without the window prefix.

Example

```
let person = prompt("Please enter your name", "Harry Potter");
let text;
if (person == null || person == "") {
  text = "User cancelled the prompt.";
} else {
  text = "Hello " + person + "! How are you today?";
}
```

QUE:15→ **What is the use of Void (0)?**

Ans)

In a programming language, void means return nothing. “javascript: void(0)” is similar to void. javascript: void(0) means return undefined as a primitive value. We use this to prevent any negative effects on a webpage when we insert some expression.

QUE:16→ **How can a page be forced to load another page in JavaScript?**

Ans)

In this article, we will see how can a page be forced to load another page in JavaScript.

Approach: We can use ***window.location*** property inside the *script* tag to forcefully load another page in Javascript. It is a reference to a Location object that is it represents the current location of the document. We can change the URL of a window by accessing it.
Syntax:

```
<script>
```

```
    window.location = <Path / URL>
</script>
```

Example:

```
<script>
    window.location =
"https://www.geeksforgeeks.org/"
</script>
```

QUE:17→ **What are the disadvantages of using innerHTML in JavaScript?**

Ans)

Disadvantages of using innerHTML property in JavaScript:

- **The use of innerHTML very slow:** The process of using innerHTML is much slower as its contents as slowly built, also already parsed contents and elements are also re-parsed which takes time.

- **Preserves event handlers attached to any DOM elements:** The event handlers do not get attached to the new elements created by setting innerHTML automatically. To do so one has to keep track of the event handlers and attach it to new elements manually. This may cause a memory leak on some browsers.

- **Content is replaced everywhere:** Either you add, append, delete or modify contents on a webpage using innerHTML, all contents is replaced, also all the DOM nodes inside that element are reparsed and recreated.

- **Appending to innerHTML is not supported:** Usually, += is used for appending in JavaScript. But on appending to an Html tag using innerHTML, the whole tag is re-parsed.

- **Old content replaced issue:** The old content is replaced even if object.innerHTML = object.innerHTML + ‘html’ is used instead of object.innerHTML += ‘html’. There is no way of appending without reparsing the whole innerHTML. Therefore, working with innerHTML becomes very slow. String concatenation just does not scale when dynamic DOM elements need to be created as the plus’ and quote openings and closings becomes difficult to track.

- **Can break the document:** There is no proper validation provided by innerHTML, so any valid HTML code can be used. This may break the document of JavaScript. Even broken HTML can be used, which may lead to unexpected problems.

- **Can also be used for Cross-site Scripting(XSS):** The fact that innerHTML can add text and elements to the webpage, can easily be used by malicious users to manipulate and display undesirable or harmful elements within other HTML element tags.

- Cross-site Scripting may also lead to loss, leak and change of sensitive information.