***ASSIGNMENT***

**1). What 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.

**2). What is use of is NaN function ?**

**Ans.** The NaN is isNaN() function means ‘Not-a-Number’. The function is used to determine whether an input value is a Number, that is, it’s type is a Number type or not.

Example :-

1. console.log(isNaN(10)); // returns "false"

2. console.log(isNaN('10')); // still returns "false"

3. console.log(isNaN('sample')); // returns "true"

In the 2nd example, the input is ‘10’ and the result is false. This is because here is an implicit conversion before checking if the given input is a NaN .

**3). 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.

Note: JavaScript shows the NEGATIVE\_INFINITY value as -Infinity.

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.

**4). Which company developed javascript ?**

**Ans.** JavaScript was invented by Brendan Eich in 1995. It was developed for Netscape 2, and became the ECMA-262 standard in 1997. After Netscape handed JavaScript over to ECMA, the Mozilla foundation continued to develop JavaScript for the Firefox browser. Mozilla's latest version was 1.8.

**5). What are undeclared and undefined variables ?**

**Ans. 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:**

**Undefined:** var geek;

undefined

console.log(geek)

**Undeclared:** //ReferenceError: myVariable is not defined

console.log(myVariable)

**6). Write the code for adding new elements dynamically ?**

**Ans.** New elements can be dynamically created in JavaScript with the help of createElement() method. The attributes of the created element can be set using the setAttribute() method. The examples given below would demonstrate this approach.

**7). What is the difference between viewstate and sessionstate ?**

**Ans.** It is known that the web is stateless, which means every time a specific webpage is requested. It is recreated each time and posted to the server. Also, HTTP is a stateless protocol, i.e. it cannot hold client information on the webpage. So it need to maintain the state of a page and the server-side as well, State management is done. ViewState and SessionState are used for client-side state management and server-side state management respectively. The basic difference between these two is that the ViewState is to manage state at the client’s end, making state management easy for end-user while SessionState manages state at the server’s end, making it easy to manage content from this end too.

**ViewState:** It is maintained at only one level that is page-level. Changes made on a single page is not visible on other pages. Information that is gathered in view state is stored for the clients only and cannot be transferred to any other place. View state is synonymous with serializable data only.

ViewState has a tendency for the persistence of page-instance-specific data. When view state is used, the values posted of a particular page persist in the browse area that the client is using and post back only when the entire operation is done. The data of the previous page is no longer available when another page is loaded. Also, Data is not secure in this case because it is exposed to clients. Encryption can be used for data security.

**SessionState:** It is maintained at session-level and data can be accessed across all pages in the web application. The information is stored within the server and can be accessed by any person that has access to the server where the information is stored.

SessionState has the tendency for the persistence of user-specific data and is maintained on the server-side. This data remains available until the time that the session is completed or the browser is closed by the user. The session state is only valid for type objects.

**Differences between ViewState and SessionState:**

|  |  |
| --- | --- |
| **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 session state, user data remains in the server. Data is available to user until the browser is closed or there is session expiration. |
| Information is stored on the client’s end only. | Information is stored on the server. |
| used to allow the persistence of page-instance-specific data. | used for the persistence of user-specific data on the server’s end. |
| ViewState values are lost/cleared when new page is loaded. | SessionState can be cleared by programmer or user or in case of timeouts. |

**8). What is === operator ?**

**Ans.** === (Triple equals) is a strict equality comparison operator in JavaScript, which returns false for the values which are not of a similar type. This operator performs type casting for equality. If we compare 2 with “2” using ===, then it will return a false value.

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

**Ans.** There are three ways to change the style of an element.

1. **External CSS :** It is used with a <link> tag and imports CSS from outside of the HTML document.

**For example :**

<link href=”style.css” type=”text/css” rel=”stylesheet”>

And then we define the stylesheet named style.css with an extension .css where we can define the styles by using tags, ids, classes and attributes.

Tag :

img {

display : none;

height : 200px;

width : 200px;

}

Id:

#thisimage {

display : none;

height : 200px;

width : 200px;

}

or img#thisimage

Class :

.thisimage {

display : none;

height : 200px;

width : 200px;

}

or img.thisimage

Attribute :

img[:type] {

display : none;

height : 200px;

width : 200px;

}

or [:type]

**2. Embedded :**  It uses a <style> tag to style the HTML and is basically same as the external (using ids, classes, tags, attributes) but here we define these into HTML itself with a tag <style> inside the <head> tag usually. As,

<html>

<head>

...

<style>

.thisimage {

height : 200px;

width : 200px;

}

</style>

</head>

<body>

...

</body>

</html>

**3. Inline CSS :** In this type of CSS coding we directly add styles in the tags itself with a style attribute. For example,

<img class=”thisimage” id=”thisimage” style=”height : 200px; width : 200px;”>

or

<div style=”display : none;”></div>

**10). How to read and write a file using javascript ?**

**Ans.**

**Reading from the file**

After the File System module is imported, the reading of the file in JavaScript can be done by using the readFile() function.

Syntax

The syntax to read from a file is as follows −

readFile(path, format, callBackFunc)

The readFile() function accepts three parameters including one optional parameter.

**Path −** The first parameter is the path of the test file from which the contents are to read. If the current location or directory is the same directory where the file which is to be opened and read is located then, only the file name has to be given.

**Format −** The second parameter is the optional parameter which is the format of the text file. The format can be ASCII, utf-8 etc.

**CallBackFunc −** The third parameter is the call back function which takes the error as the parameter and displays the fault is any raised due to the error.

**Write operation on a file**

After the File System file is imported then, the writeFile() operation is called. The writeFile() method is used to write into the file in JavaScript. The syntax of this method is as follows −

writeFile (path , inputData , callBackFunction)

The writeFile() function accepts three parameters −

**Path −** The first parameter is the path of the file or the name of the file into which the input data is to be written.

If there is a file already, then the contents in the file are deleted and the input which is given by the user will get updated or if the file is not present, then the file with that will be created in the given path and the input information is written into it.

**inputData −** The second parameter is the input data which contains the data to be written in the file that is opened.

**callBackFuntion −** The third parameter is the function which is the call back function which takes the error as the parameter and shows the fault if the write operation fails.

**11). What are all the looping structures in javascript ?**

**Ans.** 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
* for/of - loops through the values of an iterable object
* while - loops through a block of code while a specified condition is true
* do/while - also loops through a block of code while a specified condition is true

**12). How can you convert the string of any base to an integer in javascript ?**

**Ans.** parseInt() function is used to convert the string of any base to an integer

in JavaScript. This function returns an integer of the base which is specified in

the second argument of the parseInt() function.

**parseInt(string, radix)**

A radix parameter specifies the number system to use:

2 = binary, 8 = octal, 10 = decimal, 16 = hexadecimal.

If radix is omitted, JavaScript assumes radix 10. If the value begins with "0x",

JavaScript assumes radix 16

**13). What is the function of the delete operator ?**

**Ans.** The delete function is used on object properties. It deletes both value of

property and property itself.

**Example code:**

var person = {

firstName:"Parth",

lastName:"Shinde",

age:23,

};

delete person.age; // or delete person["age"];

it will shows person age is undefined.

**14). What are all the types of pop up boxes available in javascript ?**

**Ans.**  JavaScript has three types of pop-up boxes:

• **Alert Box:** An alert box is used if you want to make sure information is

comes through to the user. When an alert box pop up, user will have to

press “OK” to proceed. (window.alert();).

• **Confirm Box:** A confirm box is used if you want the user to verify or

accept something. When confirm box pop up, user will have to press

either “OK” or “CANCEL” to proceed. If the user press “OK” it returns

true or if the user press “CANCEL” it returns false.(window.confirm();).

• **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. (window.prompt();).

**15). What is the use of the void (0) ?**

**Ans.** Void means nothing. In a JavaScript Void(0) also similar to its meaning. It

means returns undefined as a primitive value. We use this to prevent any

negative effects on a webpage when we insert some expression.

For example, in the case of URL hyperlinks. Hyperlinks open by reloading the

page when the user clicks on the link. When you need to run some other code

in such cases, you can use javascript: void(0).

**16). How can a page be forces to load another page in javascript ?**

**Ans.** 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.

**17). What are the disadvantages of using innerHTML in javascript ?**

**Ans.**

* 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.

• 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.

• 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.