1. **What is JavaScript?**

JavaScript is a scripting language that enables you to create dynamically updating content, control multimedia, animate images, and pretty much everything else.

1. **What is the use of is NaN function?**

**In javaScript Nan is short for “Not-a-Number”.**

**The is Nan() method returns true if a value is NaN.**

**The is NaN() method converts the value to a number before testing it.**

1. **What is negative Infinity?**

**The negative infinity in JavaScript is a constant value which is used to represent a value which 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**

**1, Negative infinity results in -0(different from 0 ) when divided by any other number.**

**2, When divided by itself or positive infinity, negative infinity return NaN**

**3, Negative infinity, when divided by any positive number (apart from positive infinity) is negative infinity.**

**4, Negative infinity, divided by any negative number (apart from negative infinity) is positive infinity.**

**5, If we multiply negative infinity with NaN, we will get NaN as a result.**

**6, The product of 0 and negative infinity is Nan.**

1. **Which company developed JavaScript?**

**JavaScript is a scripting language developed by Netscape. It can be used to program web browser or even servers. It can dynamically update the contents of the webpage, which is the beauty of JavaScript**

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

**•Undefined variable means a variable has been declared but**

**does not have a value.**

**•Undeclared variable means that the variable does not exist in**

**the program at all.**

1. **Write the code for adding new elements dynamically?**

<!DOCTYPE html>

<html>

<head>

<meta charset=”UTF-8”>

<meta Http-equiv=”X-UA-Compatible” content=”IE=edge”>

<meta name=”viewport” content=”width-device-width, initial-scale=1.0”>

<title>Adding New Elements</title>

</head>

<script type="text/javascript">

function addNode() { var newP = document.createElement("p");

var textNode = document.createTextNode(" This is a new text node");

newP.appendChild(textNode); document.getElementById("firstP").appendChild (newP); }

</script>

</head>

<body> <p id="firstP">firstP<p> </body>

</html>

**7, What is the difference 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?

. 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

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

1: Changing CSS with the help of the style property:

Syntax:

document.getElementById("id").style.property = new\_style

Example: In this example, we have built a PAN number validator. First, we will take the input value and match it with a regex pattern. If it matches then using JavaScript add an inline style on the <p> tag. Otherwise, add a different style on the <p> tag.

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset=”UTF-8”>

<meta Http-equiv=”X-UA-Compatible” content=”IE=edge”>

<meta name=”viewport” content=”width-device-width, initial-scale=1.0”>

<title>document</title>

</head>

<body>

<h1 style="color: green;">

GeeksforGeeks

</h1>

<h2>

How can the style/class of

an element be changed?

</h2>

<b>Validate Pan Number</b>

<input type="text" id="pan" />

<p></p>

<button id="submit">Validate</button>

<script>

const btn = document.getElementById("submit");

btn.addEventListener("click", function () {

const pan = document.getElementById("pan").value;

const para = document.querySelector("p");

let regex = /([A-Z]){5}([0-9]){4}([A-Z]){1}$/;

if (regex.test(pan.toUpperCase())) {

para.innerHTML = "Hurrey It's correct";

// Inline style

para.style.color = "green";

} else {

para.innerHTML = "OOps It's wrong!";

// Inline style

para.style.color = "red";

}

});

</script>

</body>

</html>

2. The className Property: This property is used to set the current class of the element to the specified class.

Syntax:

document.getElementById("id").className = class

Example:

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset=”UTF-8”>

<meta Http-equiv=”X-UA-Compatible” content=”IE=edge”>

<meta name=”viewport” content=”width-device-width, initial-scale=1.0”>

<title>document</title>

<style>

.colorBlue {

color: blue;

}

.colorRed {

color: red;

}

</style>

</head>

<body>

<h1 style="color: green;">

GeeksforGeeks

</h1>

<h2>

How can the style/class of

an element be changed?

</h2>

<h3>className Example</h3>

<p class="colorBlue">

GeeksforGeeks is a computer science portal

for geeks.This platform has been designed

for every geek wishing to expand their

knowledge, share their knowledge and is

ready to grab their dream job. GFG have

millions of articles, live as well

as online courses, thousands of tutorials

and much more just for the geek inside you.

</p>

<button id="submit">Change Color</button>

<script>

const btn = document.getElementById("submit");

const para = document.querySelector("p");

btn.addEventListener("click", function () {

para.className = "colorRed";

});

</script>

</body>

</html>

10. How to read and write a file using JavaScript?

There are two ways to read and write a file using JavaScript

Using JavaScript extensions

Using a web page and Active X objects

11. What are all the looping structures in JavaScript? 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.

12. How can you convert the string of any base to an integer in JavaScript? The parseInt function available in JavaScript has the following signature −

parseInt(string, radix);

Where, the paramters are the following −

String − The value to parse. If this argument is not a string, then it is converted to one using the ToString method. Leading whitespace in this argument is ignored.

Radix − An integer between 2 and 36 that represents the radix (the base in mathematical numeral systems) of the string.

So we can pass the string and the radix and convert any numbner with base from 2 to 36 to integer using this method.

Example

console.log(parseInt("100", 10))

console.log(parseInt("10", 8))

console.log(parseInt("101", 2))

console.log(parseInt("2FF3", 16))

console.log(parseInt("ZZ", 36))

Output

100

8

5

12275

1295

13. What is the function of the delete operator?

The delete operator in JavaScript is used to delete an object’s

property.

If it is used to delete an object property that already exists, it returns true and removes the property from the object. However, deleting an object property that doesn’t exist will not affect the object, but will still return true.

The only time false will be returned is when the delete operator is used to delete a variable or a function.

14 .What are all the types of Pop up boxes available in JavaScript?

In Javascript, popup boxes are used to display the message or notification to the user. There are three types of pop-up boxes in JavaScript namely Alert Box, Confirm Box and Prompt Box.

Alert Box: It is used when a warning message is needed to be produced. When the alert box is displayed to the user, the user needs to press ok and proceed.

Example.



2.prompt



15. What is the use of Void (0)?

Using “javascript:void(0);” in anchor tag: Writing “javascript:void(0);” in anchor tag can prevent the page to reload and JavaScript functions can be called on single or double clicks easily.

Example:

<!DOCTYPE html>

<html lang=”en”>

<head>

<meta charset=”UTF-8”>

<meta Http-equiv=”X-UA-Compatible” content=”IE=edge”>

<meta name=”viewport” content=”width-device-width, initial-scale=1.0”>

<title>JavaScript:void(0)</title>

</head>

<body>

<center>

<h1 style="color:green">GeeksforGeeks</h1>

<h3>JavaScript:void(0)</h3>

<a href="javascript:void(0);"

ondblclick="alert('Welcome to Geeks for Geeks')">

Double click on me </a>

</center>

</body>

</html>

16. 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>

17. What are the disadvantages of using innerHTML in JavaScript?

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.

Example:

<p id="geek">Geeks</p>

title = document.getElementById('#geek')

// The whole "geek" tag is reparsed

title.innerHTML += '<p> forGeeks </p>'

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.

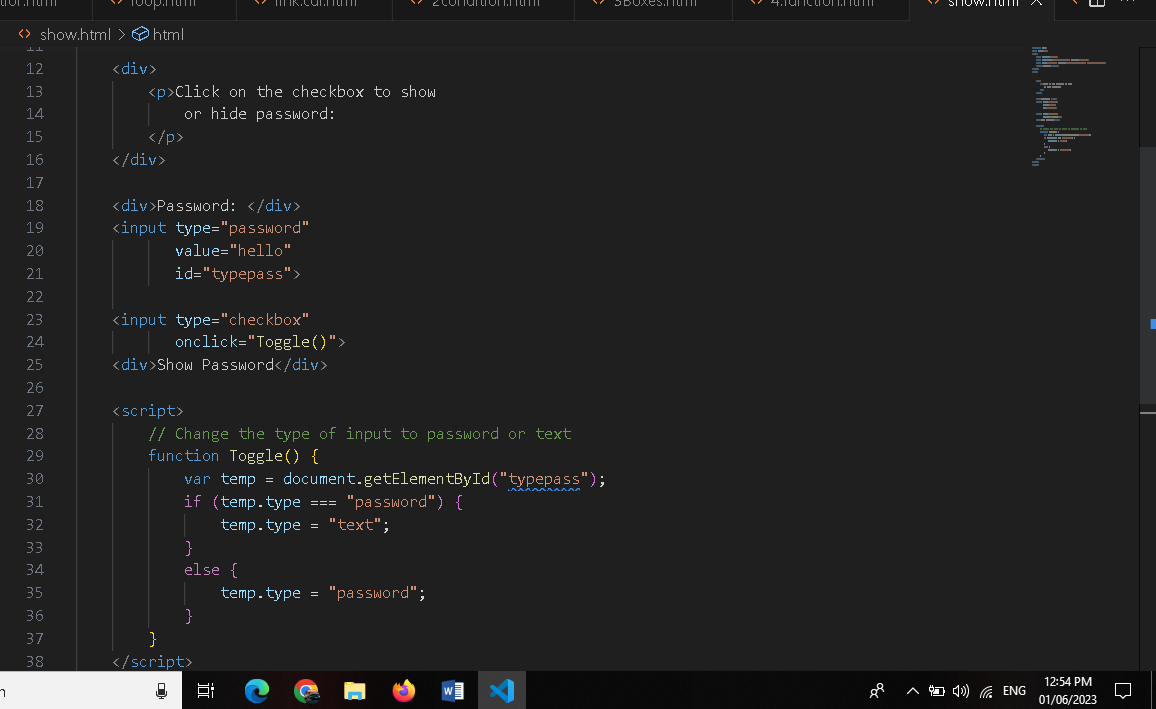
Can also be used for Cross-site Scripting(XSS): The fact that innerHTML can add text and elements to the webpage, can easily be used by malicious users to manipulate and display undesirable or harmful elements within other HTML element tags. Cross-site Scripting may also lead to loss, leak and change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

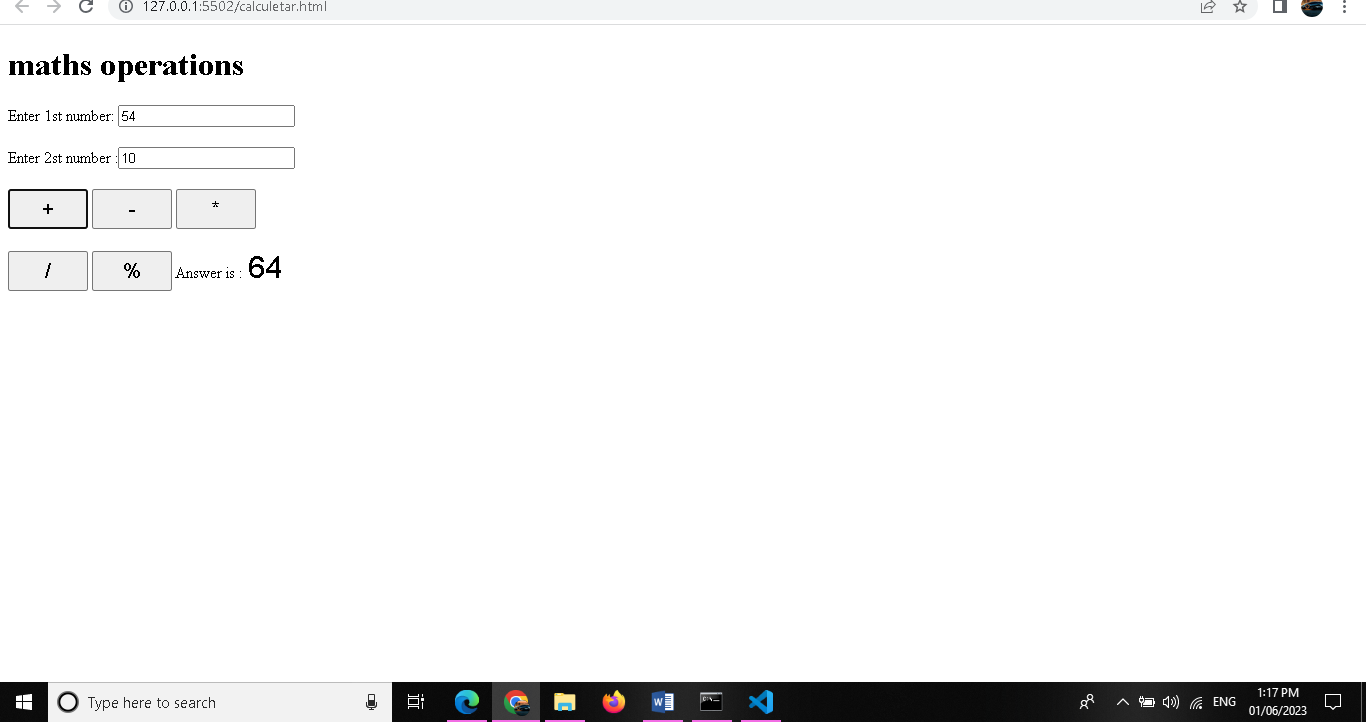


**18.Create password field with show hide functionalities**



Example.



19. Create basic math operation in JS

Example.

