JS ASSIGMENT

JAVASCRIPT
1.What is JavaScript
RAJ’S ASSIGNMENT 13
JavaScript is a versatile and widely-used programming language primarily known for its role in web development. It allows
developers to add interactivity, dynamic behavior, and advanced functionality to websites and web applications. JavaScript is a
scripting language, which means it's executed by the browser (or other runtime environments) on the client side, directly in response
to user interactions or other events.
Web Development: JavaScript is a fundamental part of front-end web development. It's used to manipulate and modify the content
of web pages in real-time, create interactive user interfaces, handle form submissions,
Dynamic Content: JavaScript allows developers to change and update the content of a webpage without requiring the user to
navigate away.
2.What is the use of isNaN function?
The isNaN() function in JavaScript is used to determine whether a value is "Not-a-Number" (NaN). NaN is a special value in
JavaScript that represents an undefined or unrepresentable value resulting from arithmetic operations.
For Example
isNaN(value);
3.What is negative Infinity?
In JavaScript, negative Infinity is a special numeric value that represents negative infinity. It is one of the values in the IEEE 754
floating-point standard, which JavaScript uses to represent numbers. Infinity, whether positive or negative, is used to represent
values that are beyond the limits of representable numbers.
const negative_value= -Infinity;
4.Which company developed JavaScript?

JavaScript was developed by Netscape Communications Corporation, which is now known as Netscape Communications or simply
Netscape. It was created by Brendan Eich, who was an engineer at Netscape. The language was originally developed under the
name "Mocha," which was later renamed to "LiveScript," and finally to "JavaScript" to capitalize on the popularity of Java at the time.

5.What are undeclared and undefined variables?

"Undeclared" and "undefined" are terms used to describe different states of variables in programming languages like JavaScript.

1. Undeclared Variables:

An undeclared variable is a variable that has been used in the code without being explicitly declared using a var , let , or const
statement.

2.

```
x = 10;
console.log(x);
```

result 10;

Undefined Variables:

An undefined variable is a variable that has been declared but has not been assigned a value. In JavaScript, when a variable is
declared but not assigned a value, it automatically takes the value undefined .

```
let demo;
console.log(demo); // Outputs undefined
```

6.Write the code for adding new elements dynamically

RAJ’S ASSIGNMENT 14

- Apple
- banan
- Graps

Add Item

7.What is the difference between ViewState and SessionState?

ViewState is a feature in ASP.NET that allows developers to persist state information across postbacks (when a form is submitted to
the server and the page is reloaded).

ViewState data is encoded and included in the HTML output of the page. When the

page is submitted back to the server, the
ViewState data is sent along with the request.
SessionState is another feature in ASP.NET that allows you to store and manage user-specific data across multiple requests. Unlike
ViewState , SessionState data is stored on the server, and a unique session ID is associated with each user's session.
8.What is === operator?
The === operator in JavaScript is called the "strict equality operator." It is used to compare two values for equality without
performing type coercion.
9.How can the style/class of an element be changed?

RAJ’S ASSIGNMENT 15

Lorem ipsum dolor sit amet consectetur adipisicing elit. A dignissimos aliquam vero aut ad voluptatem enim quos? Hic d

10.How to read and write a file using JavaScript?
JavaScript has limited access to the local file system due to security concerns.
However, you can interact with files using the File
API when working with files that are selected by the user through input elements like . Additionally, you can use
JavaScript on the server side (e.g., with Node.js) to perform more extensive file operations.
11.What are all the looping structures in JavaScript?
JavaScript provides several looping structures that allow you to repeat a block of code multiple times. Here are the main looping

structures in JavaScript:
for (initialization; condition; increment/decrement) {
// code to be executed
}
12.How can you convert the string of any base to an integer in JavaScript?
Keep in mind that parseInt() will attempt to convert the provided string to an integer in the specified base. If the string contains nonnumeric characters that are not valid for the specified base, parseInt() will stop parsing and return the converted part of the string.
const string = "42";
const int = parseInt(int);
console.log(int);
13.What is the function of the delete operator?
let demo = {
firstName: "Raj",
lastName: "Vishwakarma",
salary: 15000
}
console.log(delete emp.salary);
console.log(demo);
console.log(delete emp.salary);
console.log(demo);
14.What are all the types of Pop up boxes available in JavaScript?
alert

RAJ’S ASSIGNMENT 16

Enter ME

confirm

Hit ME

prompt

Hit Me

RAJ’S ASSIGNMENT 17

15.What is the use of Void (0)?
The usage of void(0) in JavaScript is primarily related to preventing the navigation of a page when clicking on a hyperlink or a
button. This technique is used to create "javascript:void(0)" links or buttons that perform JavaScript actions without causing the
browser to navigate to a new page or trigger a page refresh.

- Product
- Customers
- Pricing
- Resources
- Sign in
- Sign up

16.How can a page be forced to load another page in JavaScript?
// Navigate to a new page
window.location.href = "https://www.amazon.in/";
17.What are the disadvantages of using innerHTML in JavaScript?
While the innerHTML property in JavaScript is a convenient way to manipulate the content of HTML elements.
Using innerHTML to directly insert or modify HTML content from untrusted sources can lead to security vulnerabilities like Cross-Site
Scripting (XSS).
Changing an element's innerHTML can potentially remove or overwrite existing event handlers attached to child elements