Advance JavaScript for Front-End

Introduction and Code Quality

● Write a program to Show an alert.

To create an alert JavaScript, you can use the alert() method.

Here's an example :

=> alert ("hello,world") ;

When you run the above code, you'll see a pop-up alert box that says "Hello, world!". You can change the message inside the parentheses to display whatever message you want to alert the user with.

● What will be the result for these expressions?

1. 5 > 4 = This is true because 5 is greater than 4.
2. `"apple" > "pineapple"` - This is false because in lexicographical order, "apple" comes before "pineapple".
3. `"2" > "12"` - This is true because when comparing strings, it compares character by character from left to right. "2" is greater than "1" in the second position.
4. `undefined == null` - This is true in JavaScript because `undefined` and `null` are both falsy values, and they are considered equal with loose equality (`==`).
5. `undefined === null` - This is false because `===` is the strict equality operator, which checks both value and type. `undefined` and `null` are of different types, so they are not strictly equal.
6. `null == "\n0\n"` - This is false because `null` is not equal to any string, even if the string represents a falsy value.

7. `null === +"\n0\n"` - This is false because `+"\n0\n"` coerces the string "\n0\n" to a number, which is 0. `null` and `0` are of different types, so they are not strictly equal.

- **Will alert be shown?**

```
if ("0") { alert( 'Hello'); }
```

Yes, an alert will be shown. The condition `("0")` evaluates to `true` because the string `"0"` is considered a truthy value in JavaScript. Therefore, the code inside the curly braces `{}` will be executed, resulting in the alert displaying "Hello".

- **What is the code below going to output? alert(null || 2 || undefined);**

This code will output `2`.

In JavaScript, the logical OR (`||`) operator returns the first truthy operand or the last operand if all operands are falsy. In this case, `null` is falsy, `2` is truthy, so the expression `null || 2` evaluates to `2`. Therefore, the `alert` will output `2`.Top of Form.

New Request

- **What is JSON ?**

JSON stands for JavaScript Object Notation. It's a lightweight data interchange format that is easy for humans to read and write and easy for machines to parse and generate. JSON is often used to transmit data between a server and a web application, but it's also a common format for storing and exchanging data in many other contexts. JSON is built on two structures:

1. A collection of key/value pairs. In various programming languages, this is realized as an object, record, struct, dictionary, hash table, keyed list, or associative array.
2. An ordered list of values. In most programming languages, this is realized as an array, vector, list, or sequence.

JSON is language-independent, but it uses conventions that are familiar to programmers of the C family of languages (C, C++, C#, Java, JavaScript, and so on), making it easy to work with in many programming environments.

● What is promises ?

Promises in programming, particularly in languages like JavaScript, are objects used to represent the eventual completion or failure of an asynchronous operation. They provide a way to deal with asynchronous code in a more manageable and organized manner.

Here's a breakdown of how promises work:

1. **Creation**: You create a promise by using the `new Promise()` constructor. This constructor takes a function with two parameters, `resolve` and `reject`.
2. **Execution**: Inside this function, you perform an asynchronous operation, such as making an HTTP request or reading a file. When the operation is completed successfully, you call the `resolve` function with the result. If an error occurs, you call the `reject` function with an error object.
3. **Consumption**: Once you have a promise, you can use methods like `.then()` and `.catch()` to handle the result or the error respectively. These methods allow you to chain asynchronous operations in a readable way.

JavaScript Essentials

• What is JavaScript Output method?

Ans :

JavaScript provides several methods for outputting content to the browser. The most common ones are:

---

1. **Using `console.log()`:** This method is primarily used for debugging purposes. It outputs the content to the browser's console, which can be accessed through developer tools. It's useful for logging variable values, messages, or any other information during development.

    - ***console.log("Hello, world!");***

2. **Using `document.write()`:** This method writes content directly to the HTML document. However, it's not commonly used in production code because it can overwrite the entire document if called after the document has finished loading. It's more suitable for simple testing or educational purposes.

    - ***document.write("Hello, world!");***

3. **Manipulating the DOM:** JavaScript can manipulate the Document Object Model (DOM) to dynamically update the content of HTML elements. This is the preferred way to output content in web applications because it provides more control and flexibility.

    - ***// Assuming there's an element with id "output" in the HTML***

document.getElementById("output").innerHTML = "Hello, world!";

• How to used JavaScript Output method?

JavaScript provides several methods for outputting content to the user. The most commonly used methods include:

1.console.log(): This method is primarily used for debugging purposes. It outputs messages to the browser console, which can be viewed using the developer tools of the

browser. It's useful for displaying variables, objects, or any other information that helps in understanding the flow of the program.

- ***console.log("Hello, world!");***

2.alert(): This method displays a dialog box with a message and an OK button. It's commonly used for simple notifications or alerts to the user.

- ***alert("Hello, world!");***

3.document.write(): This method writes HTML expressions or JavaScript code to a document. It's not commonly used in modern web development due to its potential to overwrite existing content or cause issues with document parsing, but it can be useful for quick testing or demos.

- ***document.write("Hello, world!");***

4.innerHTML property: This property can be used to change the HTML content of an element. It's powerful but should be used with caution, especially when dealing with user-generated content, to avoid security vulnerabilities like cross-site scripting (XSS) attacks.

- *** ***

<script>

document.getElementById("output").innerHTML = "Hello, world!";

</script>

• How to used JavaScript Events to do all examples?

Sure, let's revise the examples using JavaScript events. We'll use the `addEventListener()` method to attach event listeners to elements and trigger the desired actions when those events occur.