Full Stack Web Development JavaScript – Question 3

Student Name: Jegapalini Jegatheeswaran
Student ID : UKI STU 860

Student Name: Kayani Thampirasa

Student ID : UKI STU 881

1.What is a variable in JavaScript?

Variable is a named storage for data, which gold values and can be manipulated by the program.

2.How to declare a variable in JavaScript?

To declare a variable in JavaScript, use var, let, or const followed by the variable name and an optional initial value.

Example:
var x = 5;
let name = "Ram";
const y = 6;

3.What are the differences between var, let, and const?

var: Function-scoped, hoisted, can be re-declared and updated.

let: Block-scoped, not hoisted for initialization, cannot be re-declared within the same

can be updated.

const: Block-scoped, not hoisted for initialization, cannot be re-declared or updated.

4.Explain variable hoisting in JavaScript.

Variable hoisting in JavaScript is a behavior in which variable and function declarations are moved to the top of their containing scope (global or function) during the compile phase before the code execution. This means that you can use a variable or a function before it is declared in the code.

5.What are the scoping rules for var, let, and const?

Var : Functional or Global scope.

Let : Block scope.

Const : Block scope

6.How can you use template literals in JavaScript?

Template literals are enclosed by backticks (`) to create strings that are easier to read, write, and maintain compared to traditional string concatenation methods

List the primitive data types in JavaScript.

String

Number

Boolean

Null

Undefined

Symbol

7.What is the different between null and undefined?

undefined means a variable has been declared but has not yet been assigned a value, whereas null is an assignment value, meaning that a variable has been declared and given the value of null .

8.How do you check the type of a variable in JavaScript?

You can use the type of operator to check the type of operator returns a string that indicates the type of the operand.

8.Explain the difference between primitive and reference data types

Primitive Data Types

Primitive data types are the most basic types of data. They are immutable, meaning their values cannot be changed once created. When you manipulate a primitive value, you are working directly with the value itself.

Reference Data Types

Reference data types, on the other hand, are more complex. They include objects, arrays, and functions. Unlike primitives, reference data types are mutable and are not stored directly in the variable but rather as references to the memory location where the data is stored.

11.**Explain the difference between primitive and reference data types.**

Primitive data types are the most basic data types provided by a programming language. They are predefined by the language and represent single values.

(e.g.. Number, String, Boolean)

Reference data types are more complex and can hold collections of values or more sophisticated data structures.

(e.g.. Object, Array, Function)

Storage:

- Primitive: Stores the actual value.
- Reference: Stores the memory address of the object.

Memory Allocation:

- Primitive: Fixed size and stored in stack memory.
- Reference: Dynamic size and stored in heap memory, with the reference in the stack.

Mutability:

- Primitive: Immutable; cannot change the value directly.
- Reference: Mutable; the contents can be changed.

Performance:

- Primitive: Faster access because they are stored directly.
- Reference: Slower access due to the indirection of references.

11.How does type coercion work in JavaScript?

JavaScript attempts to make the data types compatible to complete the operation or comparison.

12.What are the type of operator and the instance of operator used for?

In JavaScript, the typeof and instanceof operators are used to determine the type of a value or object. They serve different purposes and are used in different contexts. Here's a detailed

typeof Operator

The typeof operator is used to determine the primitive type of a given value. It returns a string indicating the type of the unevaluated operand. This operator can handle most of the basic data types.

instanceof Operator

The instance of operator is used to check whether an object is an instance of a specific constructor or class. It returns true if the object is an instance of the constructor, and false otherwise. This is particularly useful for checking inheritance and prototype chains.

13.How do you convert a string to a number in JavaScript?

Using Number() Function

Using parseInt() Function

Using parseFloat() Function

Using Unary Plus (+) Operator

Using Math.floor(), Math.ceil()

Handling Edge Cases

14.How do you convert a number to a string in JavaScript?

Using ‘toString()’ Method

Using ‘string() Function’

Using string Concatenation

Using ‘toFixed()’, ‘toExponential()’, and ‘toPrecision()’ Method

15.What is implicit type conversion?

Implicit type conversion, also known as type coercion, is the automatic conversion of values functions expect a certain type of value, and the provided value is of a different type. JavaScript tries to convert the value to the expected type to make the operation possible.

16.What are the different methods to convert a string to a number? Explain with example.

1. Using Number() Function

The Number function converts a string to a number. If the string cannot be converted to a valid number, it returns NaN.

2. Using parseInt() Function

The parseInt function parses a string and returns an integer. It allows you to specify a radix (base) for the conversion.

3. Using parseFloat() Function

The parseFloat function parses a string and returns a floating-point number.

4. Using Unary Plus (+) Operator

The unary plus operator converts a string to a number. It is a concise way to perform the conversion.

5. Using Math.floor(), Math.ceil(), or Math.round()

These methods can be used to convert strings to numbers by first using Number or parseFloat, and then applying the appropriate rounding method.

6. Handling Edge Cases

It's important to handle edge cases where the input string might not be a valid number, leadin to NaN results.

17.How do you handle type conversion when adding a number and a string.

To handle type conversion when adding a number and a string in JavaScript, explicitly convert the string to a number using **Number()** or **parseInt()** before performing the addition operation.

18.Explain how parselnt() and parseFloat() functions work.

The **parseLnt()** and **parseFloat()** functions in JavaScript are used to convert strings into numbers, with parseInt() specifically converting strings into integers and parseFloat() converting strings into floting-point numbers.

19.What are the arrays and how do you declare them?

In JavaScript, arrays are special types of objects that store multiple values in a single variable. They are used to store collections of data. Such as lists of items or sequences of values.

Let fruits = [‘Apple’ , ‘Orange’ , ‘Banana’]’;

20.What is an object in JavaScript?

In JavaScript, an object is a fundamental data structure reparenting a standalone entity with properties and methods. Objects encapsulate data and behavior, facilitating structured data storage, manipulation, and interaction with in applications, forming the backone of object-oriented programming paradigms in JavaScript.