JavaScript Practice

Data Types and Variables:

1. What are the different data types used in JavaScript variables in the provided code?

   Number, String, Boolean, Object, Array, Null, Undefined

2. Explain the difference between var, let, and const in JavaScript.

   Var declares a variable whose value can be redeclared and whose scope is within a function. Let declares a variable whose scope is within a block, cannot be redeclared but can be updated. Const is used to declare a constant value within a block of code and cannot be redeclared or updated.

3. Why does JavaScript allow assigning different data types to the same variable?

   JavaScript is dynamically typed, which allows variables to hold values of any data type

4. How does JavaScript handle variables declared but not initialized? Illustrate with an example from the code.

   JavaScript assigns these variables as undefined.

   ```
   let myUndefinedValue;

   console.log(myUndefinedValue);
   ```

5. Discuss the significance of variable names in programming and how they are used in JavaScript.

   Variable names should be descriptive and meaningful. In JavaScript, variable names are used as identifiers for data stored in memory.

Numeric Data Types:

1. What are the various numeric data types used in JavaScript, as shown in the code?
   Number
   BigInt

2. Explain the difference between integers, doubles, and Infinity in JavaScript with examples.

   Integers are whole numbers whereas doubles include floating-point numbers while infinity is a numeric value representing infinity

   ```
   let x=10;

   let y=20.13;

   let z= Infinity;
   ```

3. How does JavaScript handle arithmetic operations involving different numeric data types?

   JavaScript converts numeric data types into one common type before doing arithmetic operations

String Data Type:

1. How are strings represented in JavaScript?

   In JavaScript, strings are represented using single quotations, double quotations, or using template literals(`)

2. Discuss the difference between declaring strings with single quotes (") and double quotes ("") in JavaScript.

   Both singe and double quotes are functionally identical, but using one type allows the other to be used within a string without using the escape sequence.

3. Explain why characters are automatically treated as strings in JavaScript.

   Since JavaScript does not have a separate data type for handling single characters, they are treated as strings of length one.

Boolean and Undefined Data Types:

1. Explain the purpose of boolean variables in JavaScript.

   Boolean variables in JavaScript represent logical values, either true or false.

2. Discuss the concept of undefined in JavaScript variables and provide examples from the code.

   Undefined is a primitive value which represents the absence of a value in a variable. It is automatically assigned to variables that have been declared but not initialized.

   ```
   let y;
   if (typeof y === "undefined") {
       console.log("y is undefined");
   }
   ```

3. How are boolean variables useful in conditional statements and control flow in JavaScript?

   Boolean values can be either true or false and are used in conditional statements and loops to determine which code blocks should be executed.

Null Data Type:

1. Describe the significance of the null value in JavaScript.

   The null value is a keyword that represents the intentional absence of any value. Null is used when a variable has been assigned a value that explicitly represents no value.

2. Differentiate between null and undefined in JavaScript.

   Undefined means that a variable has been declared but has not yet been assigned a value, whereas null is used when a variable has been assigned a value that explicitly represents "no value."

3. Provide an example from the code illustrating the use of null.

   ```
   let student;
   console.log(typeof student); // Output: undefined

   // Null value
   let age = null;
   console.log(age); // Output: null
   ```

Object Data Type:

1. Explain how objects are represented in JavaScript.

   In JavaScript, objects are represented using object literals. Object literals involve creating an object by directly specifying its properties and values within curly braces{}.

2. Discuss the structure and purpose of the countryInfo object in the provided code.

   The countryInfo object is used to store information about a person's citizenship and identification number. The properties are citizenShip and idNumber and their values are 'Kenyan' and 44455567 respectively.

3. How can objects be nested within other objects in JavaScript?

   A nested object can be created by defining an object as the value of a property within another object.

```
const person = {
  name: "John Doe",
  age: 30,
  address: {
    street: "123 Main St",
    city: "Metropolis",
    zipCode: "12345"
  },
  contact: {
    email: "johndoe@example.com",
    phone: "555-1234"
  }
};
```

Array Data Type:

1. Describe the purpose and structure of arrays in JavaScript.

   Arrays are used to store collections of items in an ordered and indexed list. They store multiple values in a single variable. Arrays are created by enclosing a comma-separated list of elements within square brackets.

2. Provide examples from the code demonstrating arrays containing different data types.

```
let countryInfo = { citizenShip: 'Kenyan', idNumber: 44455567 };
let marks = [34, 56, 67, 78];
let info = { fname: 'Titus', sname: 'Kimutai', age: 23,
isStudent: true, countryInfo, marks };
let moreInfo = [countryInfo, marks, info];
console.log(moreInfo);
```

3. Discuss the concept of "array of arrays" and its significance.

   An array of arrays is a two-dimensional array of a matrix, where each element of the array is itself an array. Array of arrays is useful in organizing data in a structured manner and representing multidimensional data. They also facilitate the use of nested loops to manipulate data efficiently.

Variable Naming Conventions:

1. What are the conventions for naming variables in JavaScript?

   Variable names in JavaScript should be descriptive, non-keywords, start with a letter, underscore, or dollar sign and cannot start with a number. Variable names in JavaScript are case sensitive.

2. Discuss the importance of choosing meaningful and descriptive variable names.

   Meaningful and descriptive variable names ensure readability of code, they make it easier to debug, they act as a form of self-documentation and come in handy during collaboration.

3. Identify any variable naming conventions followed or violated in the provided code.

   Camel case: myRoom, myName, bankBalance, yearsInHeaven, firstChar, isAdmin, isPermitted

   Descriptive names: countryInfo, marks, phoneNumber

   Avoiding reserved keywords: All variable names avoid using reserved keywords.

   Meaningful context: info, moreInfo, bankBalance

Constants in JavaScript:

1. Explain the use of const keyword in JavaScript.

   The const keyword is used to declare variables whose values are intended to remain constant throughput the program. Once a value is assigned to a const variable, it cannot be reassigned.

2. Discuss why reassigning a value to a constant variable result in an error.

   When a variable is declared using const, an immutable reference is made to the value, meaning that the variable identifier cannot be reassigned to a different value. Reassigning a const variable will result in a TypeError because JavaScript enforces immutability to that reference.

3. Provide examples from the code demonstrating the declaration and use of constants.

   ```
   const phoneNumber = 254789567364; // Integer using const, hence it can never be changed
   ```