**1. What is the difference between let, var, const in JavaScript?**

|  |  |  |  |
| --- | --- | --- | --- |
| **S NO** | **var** | **let** | **const** |
| 1 | It can be updated or re-declared. | It can only be updated and can't be re-declared. | It can't be updated or re-declared. |
| 2 | It's an old way to declare a variable. | It's a new way to declare a variable introduced in ES6. | It's also a new way to declare a variable, which introduced in ES6. |
| 3 | It can be declared without initializations. | It can be declared without initializations. | It cannot be declared without initializations. |
| 4 | Var variable can be updated and re-declared within its scope. | Let variable can be updated but not re-declared. | const variable can neither be updated nor re-declared. |

**2. What are the various data types that exist in JavaScript?**

1. **Number** :- This data type is used to represent numerical values. Numbers can be integers (whole no) or floating-point no (decimal no).
2. **String** :- This data type represent textual dat. String are sequences of characters enclosed in quotes (either single or double).
3. **Boolean** :- This data type is used to represent logical values. Boolean values can be either true or false.
4. **Undefined** :- This data type represents a variable that has not been assigned a value.
5. **Null** :- This data type represent a deliberate absence of any object value.
6. **Object** :- This data type represent complex data structures, including arrays, functions and object.
7. **Symbol**:- This data type represent unique values that can be used as object keys.

**3. What are Global, function, block scopes of a variable in JavaScript?**

* **Global Scope**
  + Variables declared outside of all function are known as global variables and in the global scope. Global variables are accessible anywhere in the program.
* **Function Scope**
  + Variables that are declared inside a function are known as local variables and in the function scope. Local variables are accessible anywhere inside the function.
* **Block Scope** 
  + Variables that are declared inside a specific block and can't be accessed outside of that block. In order to access the variables of that specific block, we need to create an object for it.

**4. How do you create an array and object in JavaScript?**

* **Creating an Array**
  + To create an array , you can use square bracket [] and optionally initialize with its value.
  + **ex:** let mobiles = [ 'Lava', 'Vivo', 'Mi'] ;
* **Creating an Object**
  + To create an object, you can use curly braces {} and define key-value pairs.
  + **ex:** let empDetails = {firstName: "Kanna”, age: 16};

**5. What are the built-in methods in JavaScript?**

* **Number methods**
  + The Number object contains only the default methods that are part of every object definition.
    - toString()-> Returns the string representation of the number's value.
    - valueOf()-> Returns the number's value.
* **Boolean methods**
  + Here is a list of each method and its description.
    - toString()-> Returns a string of either "true" or "false" depending upon the value of the object.
    - valueOf()-> Returns the primitive value of the Boolean object.
* **String methods**
  + Here is a list of each method and its description.
    - charAt()-> Returns the character at the specified index.
    - length()-> Returns the length of the string.
    - match()-> Used to match a regular expression against a string.
    - slice()-> Extracts a section of a string and returns a new string.
    - toLowerCase()-> Returns the calling string value converted to lower case.
    - toUpperCase()-> Returns the calling string value converted to uppercase.
* **Array methods**
  + Here is a list of each method and its description.
    - forEach()-> Calls a function for each element in the array.
    - join()-> Joins all elements of an array into a string.
    - map()-> Creates a new array with the results of calling a provided function on every element in this array.
    - pop()-> Removes the last element from an array and returns that element.
    - push()-> Adds one or more elements to the end of an array and returns the new length of the array.
    - sort()-> Sorts the elements of an array.
* **Data methods**
  + Here is a list of each method and its description.
    - Date()-> Returns today's date and time
    - getDate()-> Returns the day of the month for the specified date according to local time.
    - getDay()-> Returns the day of the week for the specified date according to local time.
    - getFullYear()-> Returns the year of the specified date according to local time.
    - getHours()-> Returns the hour in the specified date according to local time.
    - getMilliseconds()-> Returns the milliseconds in the specified date according to local time.
    - getMinutes()-> Returns the minutes in the specified date according to local time.
    - getMonth()-> Returns the month in the specified date according to local time.
    - getSeconds()-> Returns the seconds in the specified date according to local time.
    - setDate()-> Sets the day of the month for a specified date according to local time.
    - setFullYear()-> Sets the full year for a specified date according to local time.
    - setHours()-> Sets the hours for a specified date according to local time.
    - setMilliseconds()-> Sets the milliseconds for a specified date according to local time.
    - setMinutes()-> Sets the minutes for a specified date according to local time.
* **Math methods**
  + Here is a list of each method and its description.
    - abs()-> Returns the absolute value of a number.
    - max()-> Returns the largest of zero or more numbers.
    - min()-> Returns the smallest of zero or more numbers.
    - pow()-> Returns base to the exponent power, that is, base exponent.
    - random()-> Returns a pseudo-random number between 0 and 1.

**6. What is the difference between Function declaration and Function expression?**

* **Function Declaration:**
  + A function declaration also known as a function statement declares a function with a function keyword. The function declaration must have a function name.
  + Function declaration does not require a variable assignment as they are standalone constructs and they cannot be nested inside a functional block.
  + These are executed before any other code.
  + The function in function declaration can be accessed before and after the function definition.
  + **Syntax:** function Demo(paramA, paramB) {

// Set of statements

}

* **Function Expression:**
  + A function Expression is similar to a function declaration without the function name.
  + Function expressions can be stored in a variable assignment.
  + Function expressions load and execute only when the program interpreter reaches the line of code.
  + The function in function expression can be accessed only after the function definition.
  + **Syntax:** var Demo1= function(paramA, paramB) {

// Set of statements

}

**7. What is arrow function in JavaScript?**

* Arrow function {()=>} is concise way of writing JavaScript functions in shorter way. Arrow functions were introduced in the ES6 version. They make our code more structured and readable.
* Arrow functions are anonymous functions i.e. functions without a name and are not bound by an identifier. Arrow functions do not return any value and can be declared without the function keyword. They are also called Lambda Functions.
* **Syntax:**  const Demo2 = () => {

console.log( "Hi All!" );

}

**8. Difference between “ == “ and “ === “ operators ?**

* **== Double equals:-**
  + == is a comparison operator, which transforms the operands having the same type before comparison.
  + == in JavaScript is used for comparing two variables, but it ignores the datatype of variable.
  + Checks the equality of two operands without considering their type.
  + Return true if the two operands are equal. It will return false if the two operands are not equal.
* **=== Triple equals:-**
  + === is a strict equality comparison operator in JavaScript, which returns false for the values which are not of a similar type. This operator performs type casting for equality. If we compare 2 with “2” using ===, then it will return a false value.
  + === is used for comparing two variables, but this operator also checks datatype and compares two values.
  + Compares equality of two operands with their types.
  + It returns true only if both values and data types are the same for the two variables.

1. **What is Callback function?**

* A callback is a function passed as an argument to another function.
  + This technique allows a function to call another function.
  + A callback function can run after another function has finished.

**10. What are the popular JavaScript Array methods: map, filter, find, reduce, and forEach?**

* map() when you want to transform elements in an array.
* filter() when you want to select a subset of multiple elements from an array..
* find() When you want to select a single element from an array.
* reduce() when you want derive a single value from multiple elements in an array.
* forEach() is great you need to execute a function for each individual element in an array.