Answer 6

**Javascript data type**

1.There are seven primitive data types in JavaScript: String, Number, BigInt, Boolean, Null, Undefined, and Symbol.

2.A string is a sequence of characters enclosed between the double quotes "..." Example: "abc123" "Hello World" "Hello, who is your plan ?"

3.An integer is a number without a decimal point that means these are just a numbers like 1,2,34,344 etc. A float is a number with a decimal point that means these are just a numbers with points like 1.5,2.56,34.667,344.678 etc.

4.Booleans are a binary data type with a total of two values. Using booleans, a programmer represents two opposite programming states or logical values, such as: On/Off. Yes/No.

5.BigInt is a special numeric type that provides support for integers of arbitrary length. A bigint is created by appending n to the end of an integer literal or by calling the function BigInt that creates bigints from strings, numbers etc.

6.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 .

7.Data Type Explanation

Object It is an instance that allows us to access members in JavaScript.

Array It represents a collection of similar elements in JavaScript.

RegExp It represents a regular expression in JavaScript.

**Javascript Arithmetic Operators**

8.Javascript arithmetic operators are used to perform mathematical calculations between variables and/or values. These include addition (+), subtraction (-), multiplication (\*), division (/), modulus (%), increment (++), and decrement (--).

9.The correct order of arithmetic operations in C follows the PEMDAS/BODMAS rule: Parentheses/Brackets first, followed by Exponents/Orders, Multiplication and Division (evaluated from left to right), and finally Addition and Subtraction (evaluated from left to right).

10.parentheses are used in mathematics to clarify numbers, to indicate multiplication, and to group numbers in the order of operations.

**Javascript Comparison Operators**

11.The main difference between the == and === operator in javascript is that the == operator does the type conversion of the operands before comparison, whereas the === operator compares the values as well as the data types of the operands.

12.==) The strict inequality ( !== ) operator checks whether its two operands are not equal, returning a Boolean result

13.It helps to think of the ternary operator as a shorthand way or writing an if-else statement. Here's a simple decision-making example using if and else: int a = 10, b = 20, c; if (a < b) { c = a; } else { c = b; } printf("%d", c); This example takes more than 10 lines, but that isn't necessary.

Control flow

14.an example of how to use an if-else statement to check if a number is positive or negative: num = -5 if num > 0: print("The number is positive.") else: print("The number is negative.")

15.The switch statement evaluates the expression inside the parenthesis. The resulting value is then compared with the values of each case. If there's a match, the block of code associated with that case is executed. If none of the case values matches, the default block is executed (if provided).

16.

**Logical functions**

17. If you are using || the compiler checks left and right from this operator and it enough if min ONE side return true. If you are using a && operator BOTH sides have to be true (or false).

18.The && requires that both items are true – think of it as having two statements on either side (rather than just true or false itself). The && means, essentially: “Are BOTH of these statements true?” If the answer is yes, then it returns true. in any other case, it returns false.

Assignment operators

19.= Assignment

+= Addition assignment

-= Subtraction assignment

\*= Multiplication assignment

/= Division assignment

20. 10,15

**Functions**

21.Function Keyword - The function keyword is used to create the function.

Function Name - The name of the function is greet , followed by parentheses () .

Function Body - The code that is executed when we call the function. In our case, it is console. log("Hello World!");

22.The main difference between a function expression and a function declaration is the function name, which can be omitted in function expressions to create anonymous functions. A function expression can be used as an IIFE (Immediately Invoked Function Expression) which runs as soon as it is defined.

23.The code inside a function is executed when the function is invoked.function myFunction(a, b) {

return a \* b;

}

myFunction(10, 2); // Will return 20

.

24.The values that are declared within a function when the function is called are known as an argument. The variables that are defined when the function is declared are known as parameters

25.A higher order function is a function that takes one or more functions as arguments, or returns a function as its result. There are several different types of higher order functions like map and reduce

**Array methods**

26. In JavaScript, map() is a method of the Array object. It creates a new array by calling a function on every element of the original array and storing the results in a new array. map() returns the new array, and the original array is unchanged.