Answer 1: various console methods

**Console.log ()**

Mainly used to log(print) the output to the console. We can put any type inside the log (), be it a string, array, object, Boolean etc.

Console.log(“name”);

Console.log (1,2,3,4);

Console.log(true);

**console. error ()**

Used to log error message to the console. Useful in testing of code. By default, the error message will be highlighted with red color.

Console.log (“This is an error”);

**console. warn ()**

Used to log warning message to the console. By default, the warning message will be highlighted with yellow color.

**console. clear ()**

Used to clear the console. The console will be cleared, in case of Chrome a simple overlayed text will be printed like: ‘Console was cleared’ while in Firefox no message is returned.

console. clear ();

**console. time () and console. timeEnd ()**

Whenever we want to know the amount of time spend by a block or a function, we can make use of the time () and timeEnd () methods provided by the JavaScript console object. They take a label which must be same, and the code inside can be anything (function, object, simple console).

|  |
| --- |
| console. time('abc');   let fun = function () {       console. log ('fun is running');   }   let fun2 = function () {       console.log ('fun2 is running.');   }   fun (); // calling fun();   fun2(); // calling fun2();  console. timeEnd('abc'); |

Answer 2: Let, var const

**var:**The scope of a variable defined with the keyword “var” is limited to the “function” within which it is defined. If it is defined outside any function, the scope of the variable is global.  
**var is “function scoped”.**

**let:**The scope of a variable defined with the keyword “let” or “const” is limited to the “block” defined by curly braces i.e. {} .  
**“let” and “const” are“block scoped”.**

**const:** The scope of a variable defined with the keyword “const” is limited to the block defined by curly braces. However, if a variable is defined with keyword const, it cannot be reassigned.  
**“const” cannot be re-assigned to a new value. However, it CAN be mutated.**

Answer 3: datatypes in JavaScript

JavaScript variables can hold many **data types**: numbers, strings, objects and more:

Var length=18; // number the number type represents both integer and floating-point numbers.

Var name=” Priyanka”; //strings

Var x=true; // Boolean The Boolean type has only two values: true and false.

**null** for unknown values – a standalone type that has a single value null.

**undefined**for unassigned values – a standalone type that has a single value undefined