🔹 1. Declare and Print Variables

Question:

Declare a variable `name` with value `"John"` and print it using `console.log()`.

Answer:

```
let name = "John";
console.log(name);
```

Explanation:

We use `let` to declare a variable that can be reassigned. Strings are enclosed in double quotes.

🔹 2. Reassigning Variables

Question:

Declare a variable `age` with value `25`. Later, change the value to `30` and print the updated value.

Answer:

```
let age = 25;
age = 30;
console.log(age);
```

Explanation:

Variables declared with `let` can be reassigned new values.

🔹 3. Using `const`

Question:

Declare a constant `PI` with value `3.14`. Try changing its value to `3.1415` and observe what happens.

Answer:

```
const PI = 3.14;
PI = 3.1415; // ❌ This will throw an error
```

Explanation:

Constants declared with `const` cannot be reassigned. Trying to do so will throw a `TypeError`.

🔹 4. Type Checking with `typeof`

Question:

Check the type of variables:

```
let x = 42;
let y = "Hello";
let z = true;
```

Print their types using `typeof`.

Answer:

```
let x = 42;
let y = "Hello";
let z = true;

console.log(typeof x); // number
console.log(typeof y); // string
console.log(typeof z); // boolean
```

🔹 5. Declare All Primitive Data Types

Question:

Create one variable for each JavaScript primitive data type:

- String
- Number
- Boolean
- Null
- Undefined
- Symbol
- BigInt

Print their types.

Answer:

```
let str = "Hi";
let num = 123;
let bool = false;
let empty = null;
let notDefined;
let sym = Symbol("id");
let bigNum = 12345678901234567890123456789012345678901234567890n;

console.log(typeof str);     // string
console.log(typeof num);     // number
console.log(typeof bool);    // boolean
console.log(typeof empty);   // object (quirk in JavaScript)
console.log(typeof notDefined); // undefined
console.log(typeof sym);     // symbol
console.log(typeof bigNum);  // bigint
```

🔹 6. Variable Hoisting with `var`

Question:

What will be the output of the following code?

```
console.log(a);
var a = 10;
```

Answer:

undefined

Explanation:

`var` declarations are hoisted to the top and initialized with `undefined`.

🔹 7. Arithmetic with Different Data Types

Question:

Predict the output of:

```
let result = "5" + 3;
console.log(result);
```

Answer:

53

Explanation:

The number `3` is converted to a string and concatenated with `"5"`.

🔹 8. Boolean Conversion

Question:

What will be the output?

```
console.log(Boolean(0));
console.log(Boolean("Hello"));
console.log(Boolean(""));
```

Answer:

false
true
false

Explanation:

- `0` and `""` are falsy values.
- Non-empty strings like `"Hello"` are truthy.

🔹 9. Implicit Type Conversion

Question:

What will be the output?

```
console.log("10" - 2);
```

Answer:

8

Explanation:

In subtraction, `"10"` is converted to a number automatically.

🔹 10. Undefined vs Null

Question:

What is the output?

```
let x;
let y = null;

console.log(typeof x); // ?
console.log(typeof y); // ?
```

Answer:

undefined
object

Explanation:

- `typeof undefined` is `"undefined"`
- `typeof null` is `"object"` (a known JavaScript bug)

🔹 11. Declare Multiple Variables in One Line

Question:

Declare three variables `firstName`, `lastName`, and `age` in a single line with values `"Jane"`, `"Doe"`, and `28` respectively.

Answer:

let firstName = "Jane", lastName = "Doe", age = 28;
console.log(firstName, lastName, age);

🔹 12. Check Equality of Two Different Data Types

Question:

What will the following code print?

let a = 5;
let b = "5";
console.log(a == b);
console.log(a === b);

Answer:

true
false

Explanation:

- `==` compares values after type conversion (loose equality)
- `===` compares values and types (strict equality)

🔹 13. Use Template Literals

Question:

Use a template literal to print:

`My name is Sam and I am 20 years old.`

Answer:

```
let name = "Sam";
let age = 20;
console.log(`My name is ${name} and I am ${age} years old.`);
```

🔹 14. Swap Two Variable Values

Question:

Swap values of `a = 10` and `b = 20` using a third variable.

Answer:

```
let a = 10, b = 20;
let temp = a;
a = b;
b = temp;
console.log(a, b); // 20 10
```

🔹 15. Add Two Numbers Entered by User (Prompt)

Question:

Use `prompt()` to get two numbers from the user, add them, and print the sum.

Answer:

```
let num1 = Number(prompt("Enter first number:"));
let num2 = Number(prompt("Enter second number:"));
console.log("Sum:", num1 + num2);
```

> Note: prompt() works in browser environments only.

🔹 16. What's the Output (String + Boolean)?

Question:

What will this code print?

```
let result = "Value is " + true;
console.log(result);
```

Answer:

Value is true

Explanation:

Boolean `true` is converted to a string and concatenated.

🔹 17. Check if a Variable is `undefined`

Question:

Write code to check if variable `score` is `undefined`.

Answer:

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

🔹 18. Difference Between `null` and `undefined`

Question:

Explain the difference between `null` and `undefined` using code.

Answer:

```
let a;          // undefined (not assigned any value)
let b = null;   // null (explicitly assigned as "empty")

console.log(typeof a); // undefined
console.log(typeof b); // object
```

Explanation:

- `undefined`: variable declared but not assigned
- `null`: intentional absence of value

🔹 19. Use BigInt for Large Number

Question:

Declare a very large number using `BigInt` and print its type.

Answer:

```
let big = 1234567891234567891234567891n;
console.log(typeof big); // bigint
```

🔹 20. Check if Value is a Number

Question:

Use `isNaN()` to check if a variable contains a valid number.

Answer:

```
let value = "123a";
console.log(isNaN(value)); // true
```

Explanation:

`isNaN()` returns `true` if the value is *not* a number.

🔹 **21. Declare a String and Check Its Type**

Question:

Declare a variable `city` with the value `"Mumbai"` and print its type.

Answer:

```
let city = "Mumbai";
console.log(typeof city); // string
```

🔹 **22. Declare a Boolean Variable**

Question:

Create a variable `isStudent` and assign it the value `true`. Print it and its data type.

Answer:

```
let isStudent = true;
console.log(isStudent);        // true
console.log(typeof isStudent); // boolean
```

🔹 **23. Declare an Undefined Variable**

Question:

Declare a variable `marks` without assigning any value. Print its value and type.

Answer:

```
let marks;
console.log(marks);        // undefined
console.log(typeof marks); // undefined
```

🔹 **24. Declare a Null Variable**

Question:

Create a variable `response` and assign it `null`. Print its type.

Answer:

```
let response = null;
console.log(typeof response); // object (JavaScript bug)
```

🔹 **25. Use `const` for a Fixed Value**

Question:

Declare a constant variable `country` with the value `"India"`. Try printing it.

Answer:

```
const country = "India";
console.log(country); // India
```

🔹 **26. Check Type of a BigInt**

Question:

Create a variable `bigNumber` with a large number using `n` at the end. Print its type.

Answer:

```
let bigNumber = 98765432198765432 1n;
console.log(typeof bigNumber); // bigint
```

🔹 **27. Create a Symbol and Check Type**

Question:

Declare a symbol variable `id` with description `"userID"` and print its type.

Answer:

```
let id = Symbol("userID");
console.log(typeof id); // symbol
```

🔹 **28. Declare Multiple Data Types Together**

Question:

Declare the following variables:

- `title` as `"Engineer"`
- `experience` as `5`
- `available` as `false`

Print all three values and their types.

Answer:

```
let title = "Engineer";
let experience = 5;
let available = false;

console.log(title, typeof title);           // Engineer string
console.log(experience, typeof experience); // 5 number
console.log(available, typeof available);   // false boolean
```

🔹 **29. Implicit Type from Assigned Value**

Question:

Assign `"500"` to variable `price` without mentioning the type. What type will JavaScript assume?

Answer:

```
let price = "500";
console.log(typeof price); // string
```

🔹 30. Use `var` to Declare a Variable

Question:

Declare a variable `language` using `var` with the value `"JavaScript"` and print it.

Answer:

```
var language = "JavaScript";
console.log(language); // JavaScript
```

🔹 JavaScript Operator Precedence Table

Precedence	Operator	Type	Associativity	Example
1 (Highest)	`()`	Grouping	Left to Right	`(a + b)`
2	`. [] ()`	Member / Call / Access	Left to Right	`obj.prop`, `arr[0]`
3	`new` (with argument list)	Object creation	Right to Left	`new Person()`
4	`++ --`	Postfix Increment/Decrement	Left to Right	`a++`, `b--`

5	`++ -- + - ~ !`	Unary	Right to Left	`++a`, `-a`, `!a`
6	`**`	Exponentiation	Right to Left	`2 ** 3`
7	`* / %`	Multiplication, Division, Modulus	Left to Right	`a * b`, `a % b`
8	`+ -`	Addition, Subtraction	Left to Right	`a + b`, `a - b`
9	`<< >> >>>`	Bitwise Shifts	Left to Right	`a << 2`
10	`< <= > >=`	Comparison	Left to Right	`a < b`
11	`== != === !==`	Equality	Left to Right	`a == b`
12	`&`	Bitwise AND	Left to Right	`a & b`
13	`^`	Bitwise XOR	Left to Right	`a ^ b`
14	`	`	Bitwise OR	Left to Right
15	`&&`	Logical AND	Left to Right	`a && b`
16	`		`	Logical OR

17	`??`	Nullish Coalescing	Left to Right	`a ?? b`
18	`?:`	Ternary Conditional	Right to Left	`a ? b : c`
19	`= += -=` etc.	Assignment	Right to Left	`a = b`, `a += 5`
20	`yield`, `await`	Control Flow	Right to Left	`await fetch()`
21	`,`	Comma	Left to Right	`a = 1, b = 2`

🧠 **Notes:**

- **Higher precedence operators are evaluated first.**
- **Parentheses** `()` can always override default precedence.
- **Right-to-left associativity** means the operator on the right is evaluated first (e.g., `a = b = 5`).

🔹 **31. Add Two Numbers**

Question:

Declare `a = 10` and `b = 15`. Print their sum.

Answer:

```
let a = 10;
let b = 15;
console.log(a + b); // 25
```

🔹 **32. Subtract Two Numbers**

Question:

Declare `x = 50` and `y = 20`. Print the result of `x - y`.

Answer:

```
let x = 50;
let y = 20;
console.log(x - y); // 30
```

🔹 **33. Multiply Two Numbers**

Question:

Multiply `num1 = 7` and `num2 = 6`. Print the result.

Answer:

```
let num1 = 7;
let num2 = 6;
console.log(num1 * num2); // 42
```

🔹 **34. Divide Two Numbers**

Question:

Divide `total = 100` by `parts = 4` and print the result.

Answer:

```
let total = 100;
let parts = 4;
console.log(total / parts); // 25
```

- **35. Find Remainder using Modulus Operator**

Question:

Declare `a = 13` and `b = 5`. Print the remainder when `a` is divided by `b`.

Answer:

```
let a = 13;
let b = 5;
console.log(a % b); // 3
```

- **36. Exponentiation Operator**

Question:

Calculate `base = 3` raised to the power `exp = 4` using the `**` operator.

Answer:

```
let base = 3;
```

```
let exp = 4;
console.log(base ** exp); // 81
```

🔹 **37. Use All Operators Together**

Question:

Let `a = 5` and `b = 2`. Print the result of:

- `a + b`
- `a - b`
- `a * b`
- `a / b`
- `a % b`

Answer:

```
let a = 5;
let b = 2;

console.log(a + b); // 7
console.log(a - b); // 3
console.log(a * b); // 10
console.log(a / b); // 2.5
console.log(a % b); // 1
```

🔹 **38. Order of Operations**

Question:

Evaluate `10 + 2 * 5` and print the result.

Answer:

```
console.log(10 + 2 * 5); // 20
```

Explanation:

Multiplication happens before addition.

🔹 **39. Parentheses to Change Order**

Question:

What will `(10 + 2) * 5` return?

Answer:

```
console.log((10 + 2) * 5); // 60
```

Explanation:

Parentheses change the evaluation order.

🔹 **40. Negative Numbers and Arithmetic**

Question:

Evaluate `result = -5 + 3 * 2` and print it.

Answer:

```
let result = -5 + 3 * 2;
console.log(result); // 1
```

🔹 **41. Multiply and Add in One Line**

Question:

If `a = 3`, `b = 4`, and `c = 5`, print the result of `a * b + c`.

Answer:

```
let a = 3, b = 4, c = 5;
console.log(a * b + c); // 12 + 5 = 17
```

🔹 **42. Divide and Subtract in One Line**

Question:

Declare `x = 20`, `y = 4`, `z = 2`. Evaluate `x / y - z`.

Answer:

```
let x = 20, y = 4, z = 2;
console.log(x / y - z); // 5 - 2 = 3
```

🔹 **43. Use Modulus on Even Number**

Question:

If `num = 10`, what will `num % 2` return?

Answer:

let num = 10;
console.log(num % 2); // 0

- **44. Use Modulus on Odd Number**

Question:

If `num = 17`, what will `num % 2` return?

Answer:

let num = 17;
console.log(num % 2); // 1

- **45. Evaluate Expression with All Operators**

Question:

Evaluate: `5 + 3 * 2 - 4 / 2`.

Answer:

console.log(5 + 3 * 2 - 4 / 2); // 5 + 6 - 2 = 9

🔹 46. Cube a Number

Question:

Declare `n = 4`. Print the cube of `n` using the exponentiation operator.

Answer:

```
let n = 4;
console.log(n ** 3); // 64
```

🔹 47. Divide Floating Point Numbers

Question:

If `x = 7.5` and `y = 2.5`, what is `x / y`?

Answer:

```
let x = 7.5, y = 2.5;
console.log(x / y); // 3
```

🔹 48. Chained Operations Without Parentheses

Question:

Evaluate `8 + 6 / 2 * 3`.

Answer:

```
console.log(8 + 6 / 2 * 3); // 8 + 3 * 3 = 8 + 9 = 17
```

🔹 **49. Use Negative Values**

Question:

If `a = -4` and `b = 2`, print `a * b`.

Answer:

let a = -4, b = 2;
console.log(a * b); // -8

🔹 **50. Expression Using All Operators with Parentheses**

Question:

Evaluate `((2 + 3) * (4 - 1)) ** 2`.

Answer:

console.log(((2 + 3) * (4 - 1)) ** 2); // (5 * 3)^2 = 15^2 = 225

🔹 **51. String + Number**

Question:

What is the result of `"10" + 5`?

Answer:

console.log("10" + 5); // "105"

Explanation:

`+` with a string causes string concatenation. Number `5` is converted to `"5"`.

🔹 **52. String - Number**

Question:

What will `"10" - 3` return?

Answer:

console.log("10" - 3); // 7

Explanation:

- forces `"10"` to be converted to a number.

🔹 **53. Number + Boolean**

Question:

What is `true + 2`?

Answer:

console.log(true + 2); // 3

Explanation:

`true` is converted to `1`. So, `1 + 2 = 3`.

🔹 **54. Number - Boolean**

Question:

Evaluate `5 - false`.

Answer:

console.log(5 - false); // 5

Explanation:

`false` becomes `0`. So, `5 - 0 = 5`.

🔹 **55. Number + null**

Question:

What is `10 + null`?

Answer:

console.log(10 + null); // 10

Explanation:

`null` is converted to `0` in arithmetic operations.

🔹 **56. Number + undefined**

Question:

What is the result of `10 + undefined`?

Answer:

console.log(10 + undefined); // NaN

Explanation:

`undefined` can't be converted to a number → result is `NaN` (Not a Number).

🔹 **57. String * Number**

Question:

Evaluate `"4" * 2`.

Answer:

console.log("4" * 2); // 8

Explanation:

- triggers string-to-number conversion.

🔹 **58. String / Number**

Question:

What is `"100" / 4`?

Answer:

console.log("100" / 4); // 25

Explanation:

`"100"` becomes `100` for division.

🔹 **59. null + true**

Question:

What is the result of `null + true`?

Answer:

console.log(null + true); // 1

Explanation:

`null → 0`, `true → 1`, so `0 + 1 = 1`.

🔹 **60. Boolean * String**

Question:

What is `true * "3"`?

Answer:

console.log(true * "3"); // 3

Explanation:

`true → 1`, `"3" → 3`, so `1 * 3 = 3`.

- **61. undefined - undefined**

Question:

What will `undefined - undefined` return?

Answer:

console.log(undefined - undefined); // NaN

Explanation:

Arithmetic with `undefined` always gives `NaN` because it cannot be converted to a number.

- **62. NaN + 5**

Question:

What will `NaN + 5` return?

Answer:

console.log(NaN + 5); // NaN

Explanation:

Any arithmetic operation with `NaN` results in `NaN`.

- **63. "abc" - 2**

Question:

What is `"abc" - 2`?

Answer:

console.log("abc" - 2); // NaN

Explanation:

`"abc"` can’t be converted to a number, so the result is `NaN`.

- **64. null / 2**

Question:

What is `null / 2`?

Answer:

console.log(null / 2); // 0

Explanation:

`null` becomes `0`, so `0 / 2 = 0`.

- **65. true - false**

Question:

What is `true - false`?

Answer:

console.log(true - false); // 1

Explanation:

`true → 1`, `false → 0`, so `1 - 0 = 1`.

- **66. "20" + true**

Question:

What will `"20" + true` return?

Answer:

console.log("20" + true); // "20true"

Explanation:

`+` with a string results in string concatenation.

🔹 **67. false + "10"**

Question:

Evaluate `false + "10"`.

Answer:

console.log(false + "10"); // "false10"

Explanation:

String concatenation again. `false` becomes `"false"`.

🔹 **68. true + null**

Question:

What is `true + null`?

Answer:

console.log(true + null); // 1

Explanation:

`true → 1`, `null → 0`, so `1 + 0 = 1`.

- **69. "5" % 2**

Question:

What is the result of `"5" % 2`?

Answer:

console.log("5" % 2); // 1

Explanation:

`"5"` is converted to `5`, so `5 % 2 = 1`.

- **70. "8" * null**

Question:

What is `"8" * null`?

Answer:

console.log("8" * null); // 0

Explanation:

`"8"` becomes `8`, `null` becomes `0`, so `8 * 0 = 0`.