```javascript
console.clear();

/*

====================================================================

                Operators Interview Questions 🚀

====================================================================

*/


/*

1. What is the difference between == and === in JavaScript?

👉 == checks whether the values of two operands are equal or not. === checks whether the values and data type of two operands are equal or not. For example: 👇

*/

let num1 = 10;

let num2 = "10";

console.log(num1 == num2);

console.log(num1 === num2);

/*

2. Write a program that determines if a person is eligible to drive based on their age being greater than or equal to 18 and having a valid driving license using ternary and logical operators.

👉 If the person is eligible, print "You are eligible to drive". If the person is not eligible, print "You are not eligible to drive". 👇

```

```
*/

let age = 18;

let haveDrivingLicense = true;

console.log(age >= 18 && haveDrivingLicense ? "You are eligible to drive." : "You are not eligible to drive.");

/*

3. What is the difference between null, undefined and Not defined in JavaScript?

👉 null is an assigned value representing the intentional absence of any object value.

👉 undefined means a variable has been declared but has not been assigned a value.

👉 Not defined means a variable has not been declared yet but we are trying to access it.

*/

let a;

console.log(a); // undefined

let b = null;

console.log(b); // null

// console.log(c); ReferenceError: c is not defined

/*

4. What is the difference between && and || operators in JavaScript?

👉 && is the logical AND operator. It returns true if both the operands are true.

👉 || is the logical OR operator. It returns true if at least one of the operands is true.

```

```
*/

console.log(true && false); // false

console.log(false || true); // true

console.log(0 || "Hello"); // "Hello"

console.log(5 && 10); // 10

console.log(0 && "Hello"); // 0

/*

5. What is the difference between && and ?? (Nullish Coalescing Operator) in JavaScript?

👉 && is the logical AND operator. It returns true if both the operands are true.

👉 ?? is the nullish coalescing operator. It returns the first non-nullish operand.?? (Nullish Coalescing) only checks for null or undefined and returns the right operand if the left is null or undefined.

*/

console.log(false && "Hello"); // false

console.log(null ?? "Default"); // "Default"

console.log(0 ?? "Fallback"); // 0

console.log("" ?? "Empty"); // ""

/*

=====================================================================

                Output Based Interview Questions 🚀

=====================================================================
```

```js

*/

console.log("5" - 3); // 2 (String is converted to number)

console.log(2 < 12 < 5); // true (2 is less than 12 and 12 is less than 5)

console.log("20" + 10 + 10); // 201010 (String concatenation)

console.log("20" - 10 - 10); // 0 (String is converted to number)

/*

=====================================================================

                Questions on Bitwise Operators 🚀

=====================================================================

*/

/*

👉 To solve this questions take the reference of binary representation table.

👉 We will draw the whole table here:

*/

/*

Decimal   Binary

0         0000
1         0001
2         0010
3         0011
4         0100
```

```
5         0101
6         0110
7         0111
8         1000
9         1001
10        1010
11        1011
12        1100
13        1101
14        1110
15        1111

*/

// By applying the bitwise operators, we get the following results:

// In this example, the binary representation of 5 is 0101 and 3 is 0011

console.log(5 & 3);

// If the element on both sides is 1, the result will be 1 otherwise 0.

// Result: 0001

console.log(5 | 3);

// If the element on either side is 1, the result will be 1 otherwise 0.

// Result: 0111

console.log(5 ^ 3);

// If the element on both sides is 0, the result will be 0 otherwise 1.

// Result: 0110

console.log(~5);

```

```
// The result will be the one's complement of 5.

// Result: -6

/*

=====================================================================

            Questions based on Ternary (Conditional) Operators 🚀

=====================================================================

*/

let c = 0;

let d = 10;

console.log(c || d && "Hello");

// Result: Hello because 0 is false and 10 is true

console.log(c && d || "World");

// Result: World because 0 is false and 10 is true

console.log(c ?? d ?? "Fallback");

// Result: Fallback because both c and d are false

/*

=====================================================================

            Questions based on Type Coercion 🚀

=====================================================================

*/

console.log([] + {});

// Result: [object Object] because [] is an array and {} is an object and both are implicitly converted to strings

console.log({} + []);

// Result: [object Object] because {} is an object and [] is an array and both are implicitly converted to strings

console.log(true + +"10");

// Result: 11 because true is implicitly converted to 1 and "10" is implicitly converted to Number 10

console.log(!!"false" == !!"true");

// Result: true because both are implicitly converted to true ("false" coerced to false and "true" coerced to true)

console.log([] == ![]);

// Result: true because both are implicitly converted to false (empty array coerced to empty string)

/*

=====================================================================

            Questions based on typeof Operator 🚀

=====================================================================

*/

console.log(typeof NaN); // number

console.log(typeof null); // object

console.log(typeof undefined); // undefined

console.log(typeof []); // object

console.log(typeof function() {}); // function
```