Part 1: Variables and Scope

1. Explain how var works in JavaScript. What is variable hoisting? Give a code example.

- var is one of the ways to creating variables (var x = 5;) , and it's a function scope not block scope

- hoisting is a concept of declare a variable to the top befor it's declaration but the value assigned to is undefined

console.log(a);

var a = 5;

console.log(a);

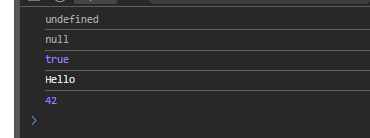


2. What is the scope of a variable declared with var inside a function? What about inside a block (e.g., an if statement)?

- it's a function scope, but about inside block it's a global variables.

3. List all JavaScript primitive types in ES5. Give an example of each.

- number => var n = 5;

- string => var str = ‘string’;

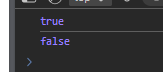
- bool => var bol = True ;

- null => var x = null;

- undefined => var y ;

4. What is the difference between a primitive type and an object type? Give an example where this difference is important.

Primitive type holding the value of the variable but the object holds a reference

*var* x = 5;

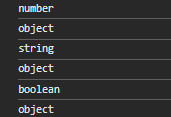
*var* y = 5;

console.log(x === y);

*var* b = new Number(10);

*var* a = new Number(10);

console.log(a == b);

5. Create a number, string, and boolean using both literal and constructor syntax. Show the difference in their types using typeof.  
*var* num1 = 42;

*var* num2 = new Number(42);

*var* str1 = "hello";

*var* str2 = new String("hello");

*var* bool1 = true;

*var* bool2 = new Boolean(true);

6 - Why is it generally recommended to use literals instead of constructors for primitive types?

* It’s better to use literals than using constructors as literals is more simple , readable and because of the unexpected output of comparisons like

var a = 4;

var b = new Number(4);

console.log(a === b);

7- Given the following code, what will be the output? Explain why.

 var x = 123.4567;

console.log(x.toFixed(2));

console.log(x.toPrecision(4));

because tofixed method approximates the decimal numbers (from right) but toprecision approximates integers (from left)

8- Nan is a (toxic value ) not a number value assigned to a variable when it’s a wrong math operation like   
var x = 5 / 0 ;  
or var y = 5\* ‘1a’

9 - What is the difference between What is the difference between parseInt, parseFloat, and Number? Give an example for each. parseFloat, and Number? Give an example for each.

parseInt convert the the variable to be integer and parseFloat convert the the variable to be float and about the number it can convert the bool and string to a number  
*var* y = 5.5;

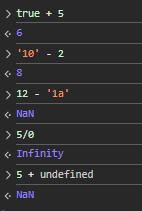
console.log(parseInt(y))

*var* x = '5.1';

console.log(parseFloat(x))

*var* n = true ;

console.log(parseFloat(n))

10 - using + allow you to concate text and convert Boolean to numeric 1 - 0   
and \* - / try to convert to a numeric too but if can’s the output will be Nan ,and undefined   
consider as Nan and if add any thing to nan the result is Nan

12 - What will be logged to the console in the following code? Explain each step.

var a = "15.5";

var b = +a;

console.log(b, typeof b);

the output will be 15.5 ‘number’ as + opetator convert the string to a number .

13 - What will be the output of:

var result = 20 > true < 5 == 1;

console.log(result);

Explain why. The output is true as   
from left to right true (which ==1) < 20 and <5 🡺 so output in true

14 –

*function* strtonum(*str*) {

*var* num = +str;

  if (!isNaN(num) ){

      return true;

  } else {

    return false;

  }

}

console.log(strtonum('11'))

console.log(strtonum('12.5'))

console.log(strtonum('1ff'))

console.log(strtonum('.021'))

console.log(strtonum('asd'))

15 - *var* num = 1;

while (num <= 20) {

  console.log(num);

  num += 1;

}

16 - Write a program that asks the user to enter numbers until they enter 0, using a do...while loop. After the loop ends, print the sum of all entered numbers (excluding 0).

*var* sum = 0;

*var* n;

do {

  n = +(prompt("Enter a number or 0 to stop:"));

  if (n !== 0) {

    sum += n;

  }

} while (n !== 0);

console.log("The summation of numbers = " + sum);

17 - for (*var* i = 1; i <= 7; i++) {

*var* day;

  switch (i) {

    case 1:

      day = "Sunday";

      break;

    case 2:

      day = "Monday";

      break;

    case 3:

      day = "Tuesday";

      break;

    case 4:

      day = "Wednesday";

      break;

    case 5:

      day = "Thursday";

      break;

    case 6:

      day = "Friday";

      break;

    case 7:

      day = "Saturday";

      break;

    default:

      day = "Invalid number";

  }

  console.log(i + "=>" + day);

}