

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

Var: declaring the variables and hoisted supported

Hoisting: Is moving declarations on runtime to above

```
console.log(x);  
var x = 5;
```



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

function scope

global scope

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

number (5)

boolean (true)

undefined (undefined)

string (ABC)

null (null)

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

primitive type: they are not objects and easy data structured and has a value

object type: they are objects and have a reference and a value

```
var x = 5;  
var y = new Number(5)  
console.log(x==y) //False!
```

5. Create a number, string, and boolean using both literal and constructor syntax. Show the difference in their types using typeof.

```
var x = 5;
var y = new Number(7);

var s = "String"
var st = new String("STRING")

var b = false
var bol = new Boolean(true)

console.log(typeof(x))
console.log(typeof(y))
console.log(typeof(s))
console.log(typeof(st))
console.log(typeof(b))
console.log(typeof(bol))
```

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

Because the literals are easy in coding and more precise about the types and comparison

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));
```

```
123.46
123.5
```

toFixed() is rounding to specific decimal number (2 steps)

toPrecision() is round to total exactly digits (4 steps)

8. What is NaN? How can you check if a value is NaN? Give an example.

NaN is Not a Number type, which is indicating the non-numerical value

Using isNaN() can make it

```
var a = parseInt("a 11");
if (isNaN(a)) {
  console.log("YES");
}
```

YES

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

***parseInt()* → use when extracting an integer from string**

***parseFloat()* → use when extracting an decimal from string**

***Number()* → use when converting value to number type but more stricter than Parse**

<pre>console.log(parseInt("123px"));</pre>	123
<pre>console.log(parseFloat("123.45px"));</pre>	123.45
<pre>console.log(Number("123px"));</pre>	NaN

10. What is the difference between implicit and explicit type casting? Give an example of each.

Implicit done automatically by using integrated symbols of javascript

Explicit done by using some functions and declarations

<pre>var n = 45;</pre>	
<pre>var s = String(n);</pre>	I am 23
<pre>console.log(s);</pre>	45

11. What will be the result and type of the following expressions? Explain your answer.

- true + 5
- "10" - 2
- 12 - "1a"
- 5 / 0
- 5 + undefined

<pre>console.log(true + 5);</pre>	6
<pre>console.log(typeof(true + 5));</pre>	number
<pre>console.log("10" - 2);</pre>	8
<pre>console.log(typeof("10" - 2));</pre>	number
<pre>console.log(12 - "1a");</pre>	NaN
<pre>console.log(typeof(12 - "1a"));</pre>	number
<pre>console.log(5 / 0);</pre>	Infinity
<pre>console.log(typeof(5 / 0));</pre>	number
<pre>console.log(5+undefined);</pre>	NaN
<pre>console.log(typeof(5+undefined));</pre>	number

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);
```

```
var a = "15.5";  
var b = +a;  
console.log(b, typeof b); 15.5 'number'
```

Declare string "15.5"

Implicit cast to number

Print the type of b which is number

13. What will be the output of Explain why.

```
var result = 20 > true < 5 == 1;  
console.log(result);
```

Output : true

Because JS is taking logical expressions by steps from the left as if it was like this

((20 > true) < 5) == 1 (true = 1)

14. Write a function that takes a string and returns true if it can be converted to a valid number, and false otherwise

```
function canbenum (s) {  
  x = parseInt(s);  
  if (isNaN(x)){  
    return false;  
  }  
  else {  
    return true;  
  }  
}
```

15. Write a program that prints all numbers from 1 to 20 using a while loop.

```
var x = 1;  
while (x<=20) {  
  console.log(x);  
  x = x+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 number;
do {
    number = parseInt(prompt("Enter numbers"));
    if (number !== 0) {
        sum += number;
    }
} while (number !== 0);
console.log("The sum : " + sum);
```

15. Write a program that takes a number from 1 to 7 and prints the corresponding day of the week using a switch statement. Use a for loop to test your program with all numbers from 1 to 7.

```
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 day";
    }
    console.log(i + " is " + day);
}
```