Part 1: Variables and Scope

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

Var statement declares **function-scoped** or **globally-scoped** variables, optionally initializing each to a value

Hoisting is JavaScript's behavior of moving variable and function declarations to the top of their containing scope during the compile phase before code execution.

Var variables, the **declaration** is hoisted but **not the initialization**.

This means the variable is known throughout its scope from the beginning, but its value is `undefined` until the assignment line is executed.

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

```
undefined                    script.js:118
5                            script.js:120
```

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

variable declared with var inside a function is accessible only within that function (function scope). However if declared `if` statement, it is not limited to that block but is accessible to the entire function or globally if outside any function.

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

Primitive data types
Number
String
Boolean
Undefined
Null

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

Primitive type : Number,String,Boolean,Undefined
it holds immutable values and are passed by values

```
var a = 5;
var b= a;
b = 10;
console.log(a)
```

```
5                                                    script.js:126
```

Object Types : Arrays,objects,functions

It holds references to values and passwed by reference

```
var object1 = {x:5}
var object2 = object1
object2.x = 10
console.log(object1.x)
```

```
10                                                   script.js:131
```

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

```
1
2   var numLiteral = 10
3   var numCons = new Number(10)
4
5   console.log(typeof(numLiteral));
6   console.log(typeof(numCons));
7
```

```
number                                               script.js:5
object                                               script.js:6
```

```
9
10  var strLiteral = "Abdo"
11  var strCons = new String("Abdo")
12
13  console.log(typeof strLiteral);
14  console.log(typeof strCons);
```

```
string                                               script.js:13
object                                               script.js:14
```

```
var boolLiteral = true
var boolCons = new Boolean(true)

console.log(typeof boolLiteral);
console.log(typeof boolCons);
```

```
boolean                                                    script.js:20
object                                                     script.js:21
```

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

Because literals create simple **primitive values** directly, which are more efficient and readable while constructors gives more complexity and not readable

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));

```
var x = 123.4567;
console.log(x.toFixed(2));
console.log(x.toPrecision(4));
```

```
123.46                                                     script.js:24
123.5                                                      script.js:25
```

toFixed(2) rounds to 2 decimal places
toPrecision(4) rounded to 4 significant digits

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

NaN means *Not-a-Number*

```
var x= "abdo"
var y = 5
console.log(isNaN(x));
console.log(isNaN(y));
```

```
true                                    script.js:29
false                                   script.js:30
```

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

parseInt: parses integer from string, stops at first non-digit.

parseFloat: parses float from string, includes decimals.

Number: converts whole string to number, returns `NaN` if invalid.

```
var x = "10.31abc"
console.log(parseInt(x));

var y = "10.31abc"
console.log(parseFloat(y));

var z = "10.31abc"
console.log(Number(z));
```

```
10                                      script.js:33
10.31                                   script.js:36
NaN                                     script.js:39
```

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

Implicit type casting happens automatically by JavaScript.
Explicit type casting is when you manually convert types

```
console.log("5" + 1); //Implicit

console.log(+"5" + 1); //Explicit
```

```
51                                      script.js:41
6                                       script.js:43
```

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

```
console.log(true + 5);
console.log("10" - 2);
console.log("12" - "1a");
console.log(5 / 0);
console.log(5 + undefined);
```

```
6                    script.js:45
8                    script.js:46
NaN                  script.js:47
Infinity             script.js:48
NaN                  script.js:49
```

1- Ture is converted to 1 so 1 +5 = 6
2- In subtraction it converts string like “10” to number so 10 – 2 = 8
3- “1a” can’t be converted to number so it will give NaN
4- 5/0 devesion by 0 in js results infinity
5- Undefined can’t be converted to number so numer + undefined would return 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);

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

```
15.5 'number'                                                    script.js:55
>
```

Unary + Operator converts string to Number so it returns number with datatype number

13. What will be the output of:

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

    console.log(result);

    Explain why.

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

```
true                                                             script.js:58
```

1- 20 > true → 20 > 1 → true
2- True < 5 → 1 < 5 → true
3- True == 1 → 1 == 1 → true

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

```
function str (str){
    if(+str){
        return true
    }else{
        return false
    }
}

console.log(str("123")); //true
```

```
function str (str){
    if(+str){
        return true
    }else{
        return false
    }
}

console.log(str("Abdo")); //false
```

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

```
var i=1
while (i <=20){
    console.log(i);
    i++

}
```

```
1                                                    script.js:73
2                                                    script.js:73
3                                                    script.js:73
4                                                    script.js:73
5                                                    script.js:73
6                                                    script.js:73
7                                                    script.js:73
8                                                    script.js:73
9                                                    script.js:73
10                                                   script.js:73
11                                                   script.js:73
12                                                   script.js:73
13                                                   script.js:73
14                                                   script.js:73
15                                                   script.js:73
16                                                   script.js:73
17                                                   script.js:73
18                                                   script.js:73
19                                                   script.js:73
20                                                   script.js:73
```

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 num;
var sum = 0
do{
    num = Number(prompt("Enter numbers (0 to exit)"))
    if(num != 0){
        sum += num
    }

}while(num != 0)

    console.log(sum);
```

17. 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 num = 1; num <= 7; num++) {

    switch (num) {
        case 1:
            console.log("Saturday");
            break;
        case 2:
            console.log("Sunday");
            break;
        case 3:
            console.log("Monday");
            break;
        case 4:
            console.log("Tuesday");
            break;
        case 5:
            console.log("Wednsday");
            break;
        case 6:
            console.log("Thursday");
            break;
        case 7:
            console.log("Friday");
            break;
        default:
    }
}
```

Saturday	script.js:93
Sunday	script.js:96
Monday	script.js:99
Tuesday	script.js:102
Wednsday	script.js:105
Thursday	script.js:108
Friday	script.js:111