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

var variableName = value;
according to value type of variable will be determined
default scope is function scope or block scope until executed then become global scope
variable hoisting is to use variable before it's declaration
ex:
console.log(x)

var x = 5

engine detect variable at beginning of script and define it with undefined value

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
block scope
when it runs become global

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

Number -> var x = 5; var x = new Number(5);
Boolean -> var x = true; var x = new Boolean(true);
String -> var x = "ahmed"; var x = new String("ahmed");
Undefined -> var x;
NULL -> var x = null;

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

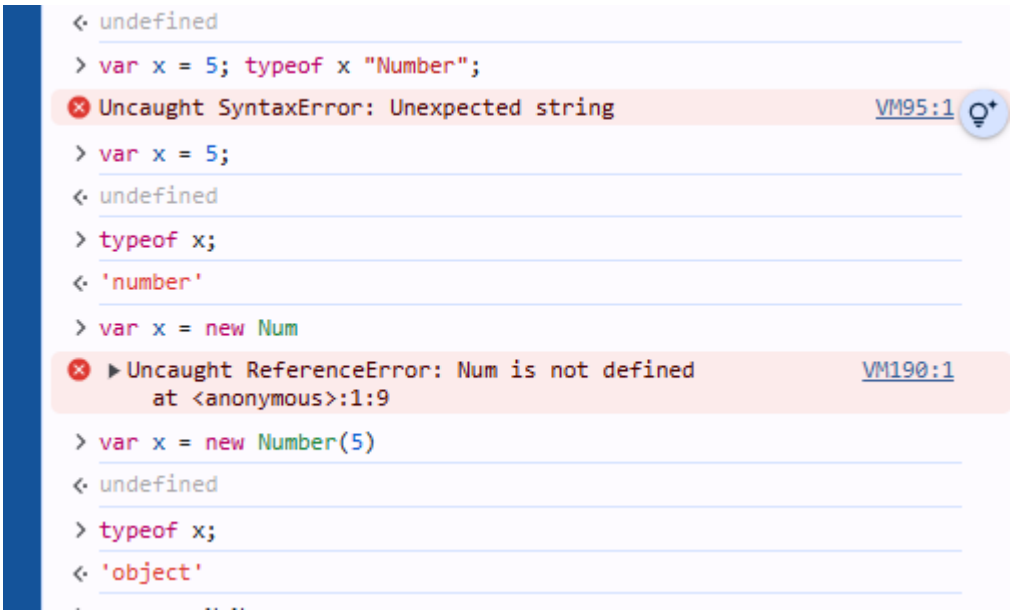
primitive is immutable and stored in memory directly
object is mutable and has attributes that contain variables and object is referencing this value

a = 5;
b = 5
x = new Number(5);
y = new Number(5);

a == b -> return true
x == y -> return false

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

Number -> var x = 5; typeof x "number" , var x = new Number(5); typeof x "object"
Boolean -> var x = true; typeof x "boolean" , var x = new Boolean(true); typeof x "object"
String -> var x = "ahmed"; typeof x "string" , var x = new String("ahmed"); typeof x "object"



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

literals are faster and more cleaner and can use value of variable directly

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

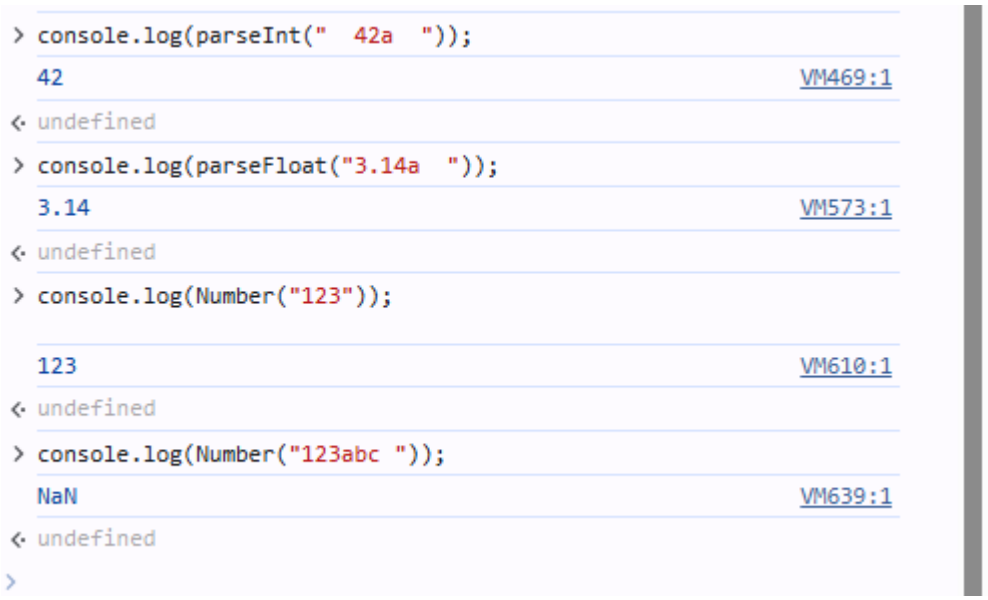
toFixed round to n digits after decimel point '.'
toPrecision round to n digits in total of number

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

Nan is Not a Number
var x = NaN; var x = parseInt("123a"); isNaN(number) - > "true"

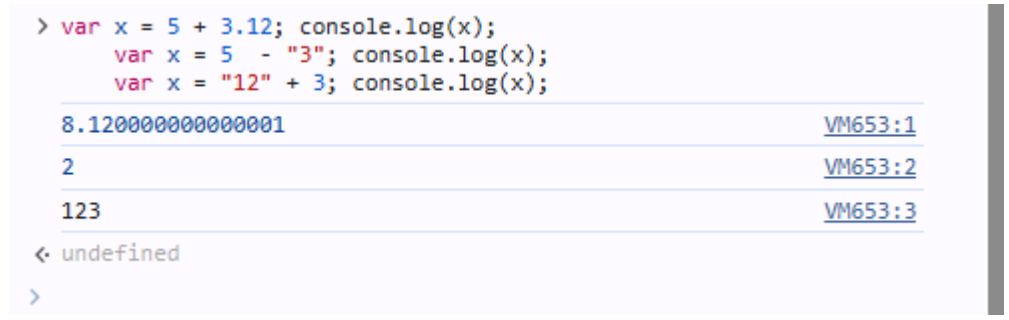
9. What is the difference between parseInt, parseFloat, and Number? Give an example for each.

parseInt -> convert string to integer excluding spaces in begin and end of string and decimal point and characters at end or output NaN
parseFloat-> convert string to Float , include decimal point in string excluding spaces in begin and end of string
Number -> convert string to number, string must be only number chars or will output NaN



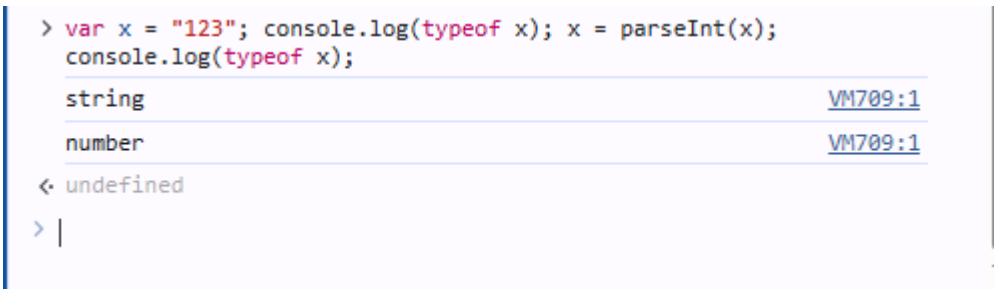
10. What is the difference between implicit and explicit type casting? Give an example of each.

Implicit -> automatic casting to correct datatypes according to applied operation
Ex: var x = 5 + 3.12; console.log(x);
  var x = 5 - “3”; console.log(x);
  var x = “12” + 3; console.log(x);



Explicit -> manuel conversion to data types mentioned

var x = “123”; typeof x; x = parseInt(x); typeof x;

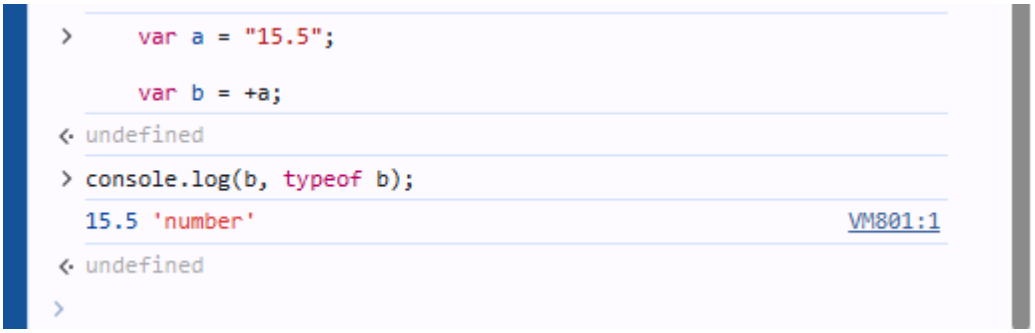


11. What will be the result and type of the following expressions? Explain your answer.

- true + 5 -> 6
  True corresponding to value 1 as number
- "10" - 2 -> 8
  Implicit casting to “10” to number with value 10 so answer will be 10 -2 = 8
- 12 - "1a" -> NaN
  Implicit casting to “1a” to number will become NaN and number - NaN is always NaN
- 5 / 0 -> infinity
  Javascript handles division by zero by output infinity number
- 5 + undefined -> NaN
  Undefined with numbers give NaN

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

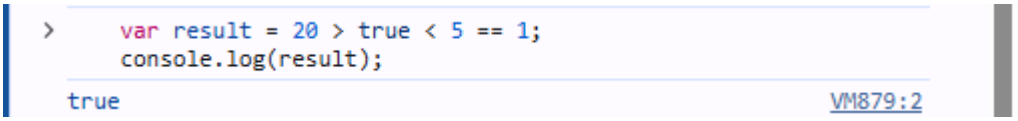
var a = "15.5";
var b = +a;
console.log(b, typeof b);



+ Before variable convert variable to number if only unary expression
  +a , not x+a

13. What will be the output of:

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



20 > true - > true, true < 5 - > true, true == 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 convertStrToNum(str)
{
    if (isNaN(str) == false && str.trim() !== "")
    {
        return true;
    }
    return false
}

console.log(convertStrToNum("123")) // return true
console.log(convertStrToNum("123.5")) // return true
console.log(convertStrToNum("1a")) // return false
console.log(convertStrToNum("")) // return false
```

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

```
var i = 1;
while( i < 21)
{
    console.log(i)
    i += 1;
}
```

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

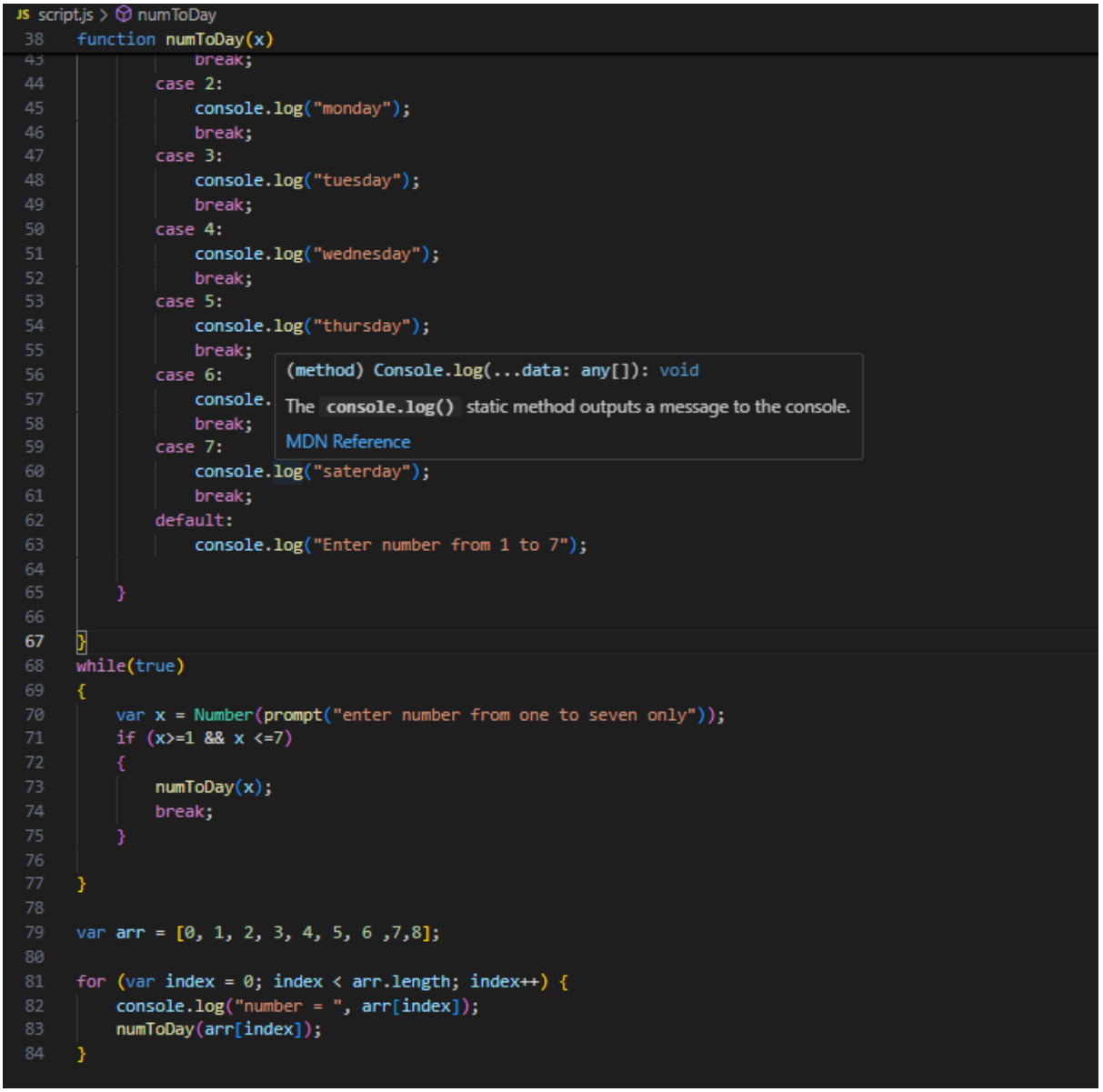
16. Write a program that asks the user to enter numbers until they enter 0, using a do...while loop. After the loop ends, print the sum of all entered numbers (excluding 0).

```
var sum = 0;

while(true)
{
    var x = prompt("enter number to add to sum");
    if( x == 0)
    {
        break;
    }
    console.log("user enterd number = ", x);
    sum += Number(x);
}
console.log("sum = ", sum)
```

```
user enterd number =  1                                    script.js:32
user enterd number =  10.5                                 script.js:32
user enterd number =  0.5                                  script.js:32
sum =  12                                                  script.js:35
>
```

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.



```
number =  0                                   script.js:85
Enter number from 1 to 7                      script.js:63
number =  1                                   script.js:85
sunday                                        script.js:42
number =  2                                   script.js:85
monday                                        script.js:45
number =  3                                   script.js:85
tuesday                                       script.js:48
number =  4                                   script.js:85
wednesday                                     script.js:51
number =  5                                   script.js:85
thursday                                      script.js:54
number =  6                                   script.js:85
friday                                        script.js:57
number =  7                                   script.js:85
saterday                                      script.js:60
number =  8                                   script.js:85
Enter number from 1 to 7                      script.js:63
>
```