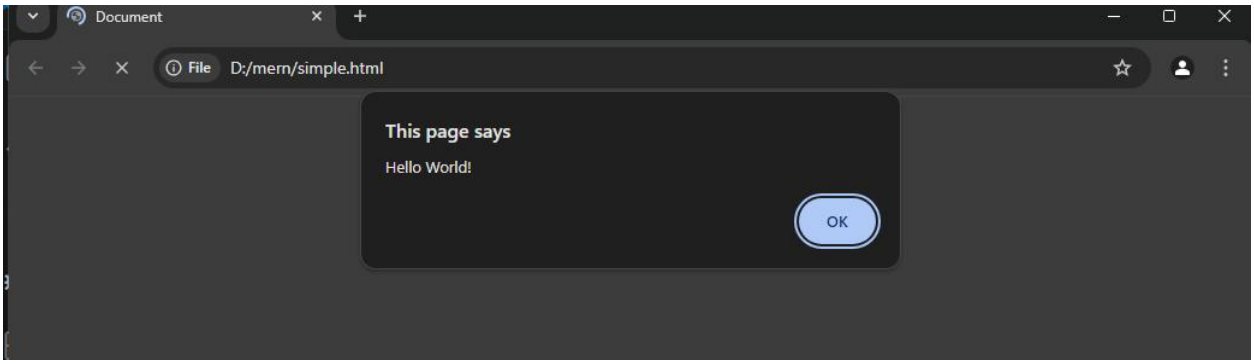
JEYSAN.V - 717823F225

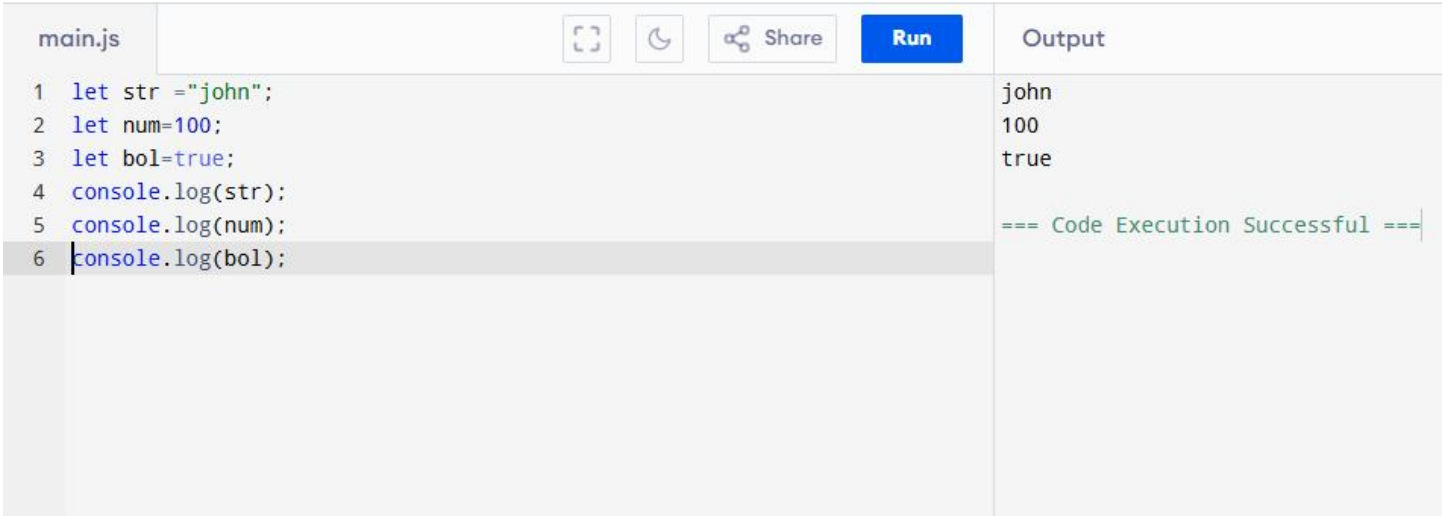
TASK 1:

```html
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Document</title>
</head>
<body>
    <script>
        alert("Hello World!");
    </script>
</body>
</html>
```

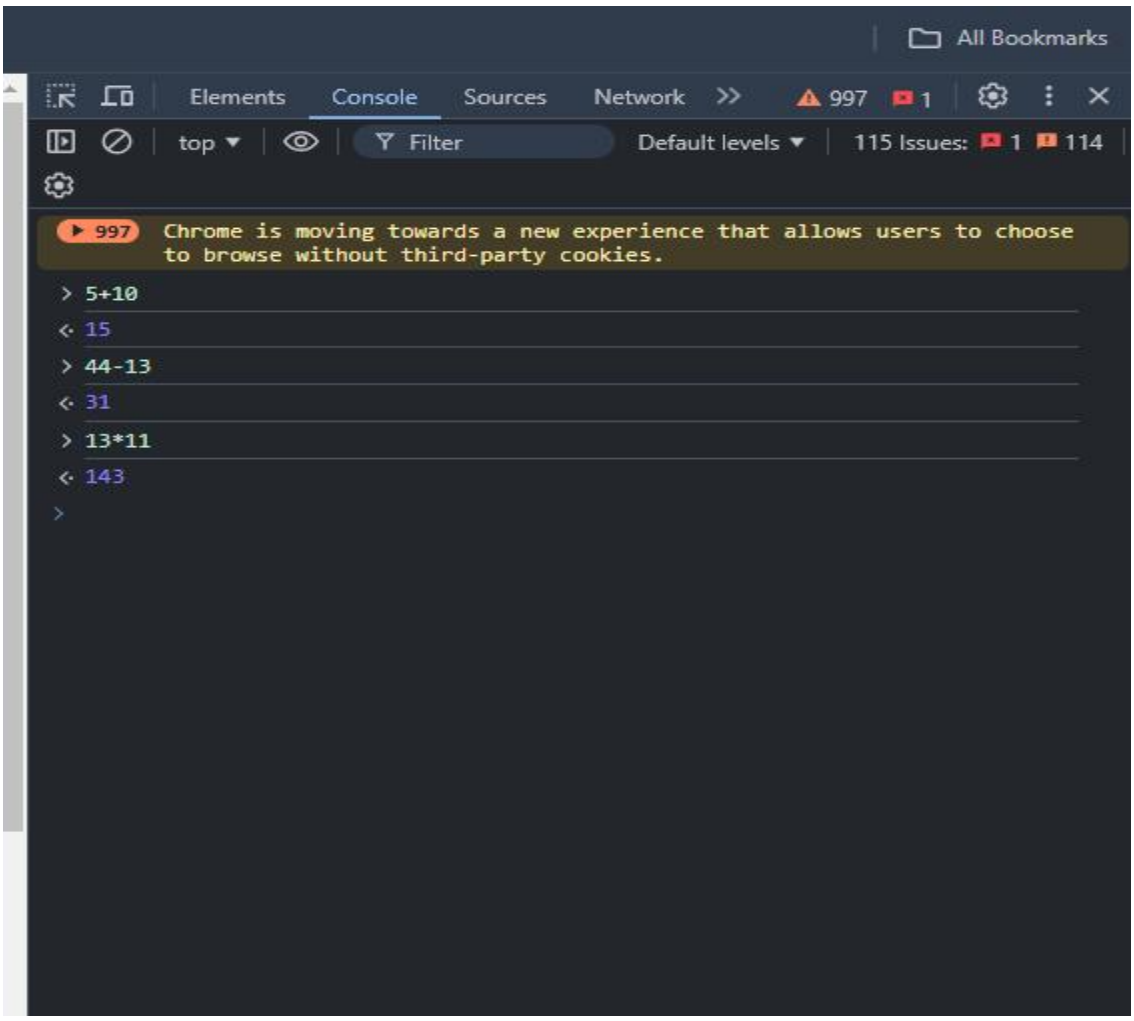
OUTPUT:



Task 2:



TASK 3:


All Bookmarks
Elements
Console
Sources
Network
997
1
top
Filter
Default levels
115 Issues: 1 114
997
Chrome is moving towards a new experience that allows users to choose to browse without third-party cookies.
> 5+10
< 15
> 44-13
< 31
> 13*11
< 143

TASK 4:


main.js
Share
Run
Output
1 var str="welcome To ";
2 var str1="JavaScript";
3 console.log(str+str1);
4
welcome To JavaScript
=== Code Execution Suc

TASK 5:

main.js

```
var str="welcome To ";
var num= 200;
var bol=false;
console.log(typeof str);
console.log(typeof num);
console.log(typeof bol);
```

Output

```
string
number
boolean

=== Code Execution
```

TASK 6:

```
<script>

        //It is single line javascript comment

        /* It is a multiline

        javascript comment…

        */

</script>
```

TASK 7:

main.js

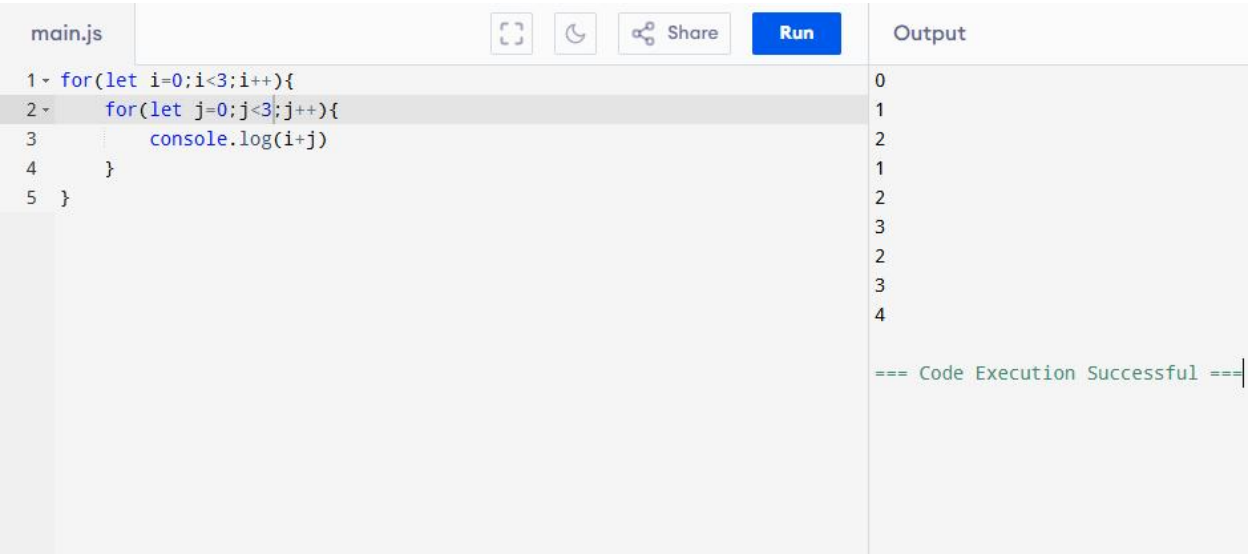
```
var x=5
var str="kce"
var val=true
var a=1.2
console.log(x)
console.log(str)
console.log(val)
console.log(a)
```

Output

```
5
kce
true
1.2

=== Code Execution
```

Task 8:



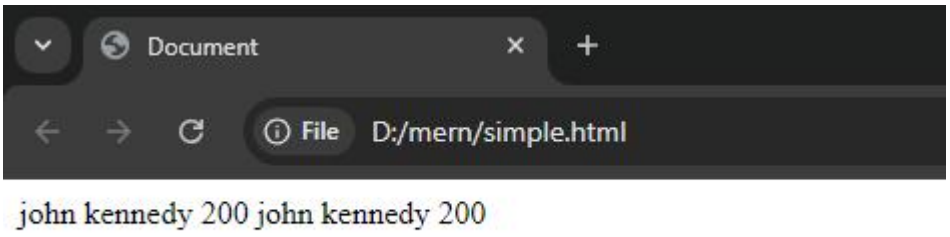
TASK 9:

TASK 10:

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Document</title>
    <script>
        var str="john";
        str2="kennedy";
        num=200;
        document.writeln(str);
        document.writeln(str2);
        document.writeln(num);
    </script>
```

```
</head>
<body>
    <script>
        var str="john";
        str2="kennedy";
        num=200;
        document.writeln(str);
        document.writeln(str2);
        document.writeln(num);
    </script>
</body>
</html>
```

Output:



Task 11:

Task 12:

Task 13:

main.js

```
1
2  let myVariable=45;
3  console.log(myVariable);
4  delete myVariable;
5  console.log("myVariable is deleted");
6
7  function myFunction(myParameter){
8      console.log(myParameter);
9      delete myParameter;
10     console.log("myParameter is deleted");
11 }
12
13 myFunction("Hello world");
14 delete myFunction;
15 console.log("myfunction is deleted");
16
17
```

Output

```
45
myVariable is deleted
Hello world
myParameter is deleted
myfunction is deleted

=== Code Execution Successful ===
```

main.js

```
1  "use strict";
2  let myVariable=45;
3  console.log(myVariable);
4  delete myVariable;
5  console.log("myVariable is deleted");
6
7  function myFunction(myParameter){
8      console.log(myParameter);
9      delete myParameter;
10     console.log("myParameter is deleted");
11 }
12
13 myFunction("Hello world");
14 delete myFunction;
15 console.log("myfunction is deleted");
16
17
```

Output

```
ERROR!
/tmp/N1NBI2Ozjp/main.js:4
delete myVariable;
       ^^^^^^^^^^

SyntaxError: Delete of an unqualified identifier in strict mode.
    at wrapSafe (node:internal/modules/cjs/loader:1515:18)
    at Module._compile (node:internal/modules/cjs/loader:1537:20)
    at Object..js (node:internal/modules/cjs/loader:1708:10)
    at Module.load (node:internal/modules/cjs/loader:1318:32)
    at Function._load (node:internal/modules/cjs/loader:1128:12)
    at TracingChannel.traceSync (node:diagnostics_channel:322:14)
    at wrapModuleLoad (node:internal/modules/cjs/loader:219:24)
    at Function.executeUserEntryPoint [as runMain] (node:internal/modules
        /run_main:170:5)
    at node:internal/main/run_main_module:36:49

Node.js v22.12.0

=== Code Exited With Errors ===
```

Task 14:

main.js

```
1
2  num=45;
3  console.log(num);
```

Output

```
45

=== Code Execution Successful ===
```

main.js Share Run Output

```
1
2 "use strict";
3 num=45;
4 console.log(num);
```

```
ERROR!
/tmp/wOYDTPy6uG/main.js:3
num=45;
   ^

ReferenceError: num is not defined
```

Task 15:

main.js Share Run Output

```
1 "use strict";
2 let for=10;
3 console.log(for);
```

```
ERROR!
/tmp/QNIFaZO7od/main.js:2
let for=10;
^^^

SyntaxError: Unexpected strict mode reserved word
```

TASK 16:

main.js Share Run Output

```
1 var n1=10;
2 let n2=20;
3 const n3=30;
4 console.log(n1);
5 console.log(n2);
6 console.log(n3);
7
```

```
10
20
30

=== Code Execution
```

TASK 17:

main.js Share Run Output

```
1 const n3=30;
2 n3=50;
3 console.log(n3);
4
```

```
ERROR!
/tmp/mpajyRYGSt/main.js:2
n3=50;
  ^

TypeError: Assignment to constant variable.
```

TASK 18:

main.js

```
1   var n;
2   console.log(n);
3
```

Output

```
undefined

=== Code Execution
```

TASK 19:

main.js

```
1  var str="welcome To ";
2  var num= 200;
3  var bol=false;
4  console.log(typeof str);
5  console.log(typeof num);
6  console.log(typeof bol);
7
```

Output

```
string
number
boolean

=== Code Execution
```

Task 20:

main.js

```
1  var m=20;
2  var m= 25;
3  console.log(m);
4  let n=20;
5  let n=10;
6  console.log(n);
7
```

Output

```
ERROR!
/tmp/KOxpfm9uyY/main.js:5
let n=10;
    ^

SyntaxError: Identifier 'n'
    at wrapSafe (node:interr
    at Module._compile (node
```

Task 21:

main.js

```
let str ="john";
let num=100;
let bol=true;
let n=null;
let m;
let student ={
    name:"john",
    age: 19
    };
console.log(str);
console.log(num);
console.log(bol);
console.log(n);
console.log(m);
console.log(student.name);
console.log(student.age);

```

Output

```
john
100
true
null
undefined
john
19

=== Code Execution Successful
```

Task 22:

main.js

```
let str ="john";
let num=100;
let bol=true;
let n=null;
let m;
let student ={
    name:"john",
    age: 19
    };
console.log(typeof str);
console.log(typeof num);
console.log(typeof bol);
console.log(typeof n);
console.log(typeof m);
console.log(typeof student);

```

Output

```
string
number
boolean
object
undefined
object

=== Code Execution Successful
```

Task 23:

main.js

```
var s=Symbol("hi");
console.log(typeof s);
```

Output

```
symbol

=== Code
```

TASK 24:

main.js

```
var s=null;
console.log(typeof s);
```

Output

```
object

=== Code
```

Task 25:

main.js

```
var a=10;
var b=20;
{
    let a=100;
    console.log(a);
    console.log(b);
}
console.log(a);
console.log(b);
```

Output

```
100
20
10
20

=== Code Execution
```

TASK 26;

main.js

```
1  //explicit
2  var a="10";
3  var num=Number(a);
4  console.log(typeof num);
5  // Implecit
6  var num2=a-0;
7  console.log(typeof num2);
```

Output

```
number
number

=== Code Execution
```

Task 27:

main.js

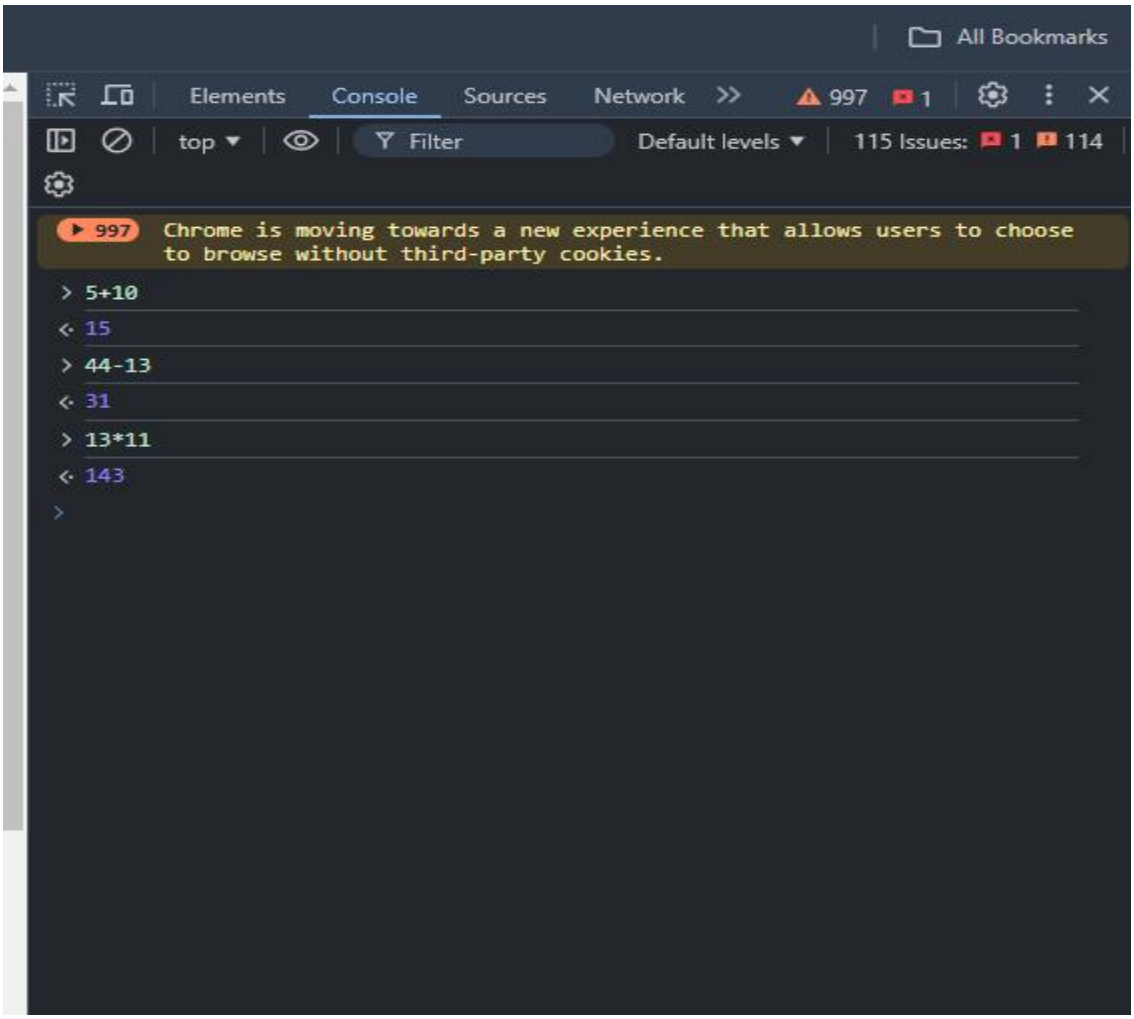
```
1  var a=false;
2  var num=String(a);
3  console.log(typeof num);
4
5  var b=Boolean(num);
6  console.log(typeof b);
7
```

Output

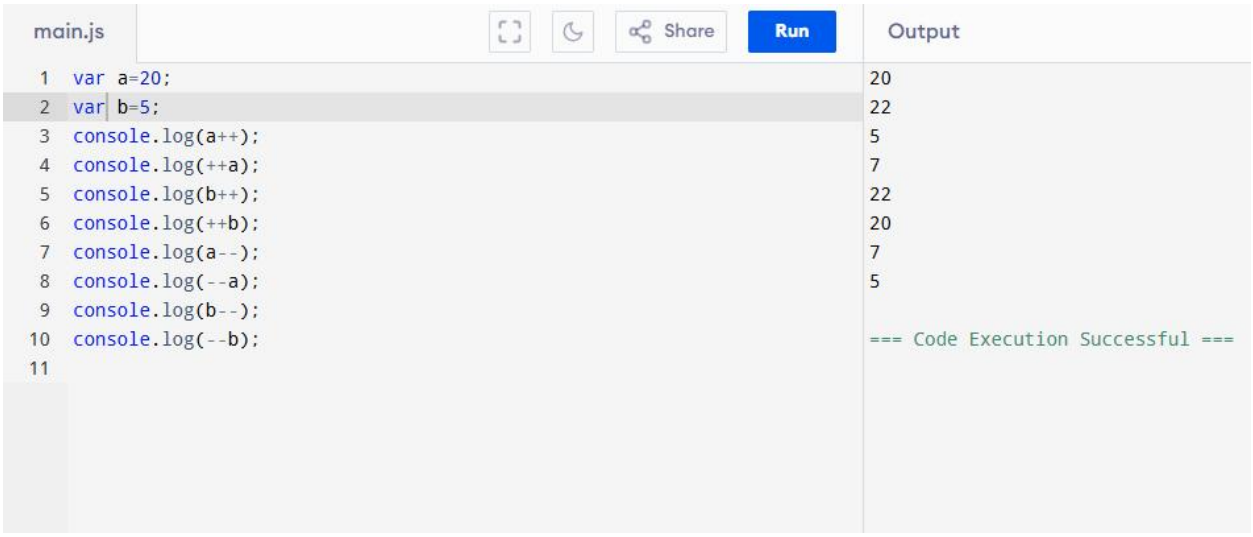
```
string
boolean

=== Code Execution
```

Task 28:



Task 29:



TASK 30:

main.js

```
var a=20;
var b=5;
var c=20;
console.log(a+b-c*a%c+(a/c));
```

Output

```
26

=== Code E
```

TASK 31:

main.js

```

console.log(33>32);
console.log(43<35);
console.log(11>=11);
console.log(21<=31);
```

Output

```
true
false
true
true

=== Code Execution
```

TASK 32:

main.js

```
var a = `30`
var b = 30
console.log(a==b)
console.log(a===b)
```

Output

```
true
false

=== Code Execution Successful ===
```

Task 33:

main.js

```
var a = `Welcome`
var b =  `welcome`
 if(a<b)
 console.log('a is lexicographically greater than b')
 else if
 (b<a) console.log(`b is lexicographically greater than a`)
  else
  console.log(`Both are same`)

```

Output

```
a is lexicographically greater than b

=== Code Execution Successful ===
```

Task 34:

main.js

```
var a = `Welcome`
let b = `Welcome`
if(a!=b)
console.log(true)
else
console.log(`different`)
if(b!==a)
console.log(`both a nd b are has different datatype`)
else
console.log(`same datatypes`)
```

Output

```
different
same datatypes

=== Code Execution Successful ===
```

Task 35:

main.js

```
var a = null
var b = undefined
if(a==b)
console.log(`same`)
else
console.log(`different`)

if(a===b)
console.log(`same`)
else
console.log(`different`)
```

Output

```
same
different

=== Code Execution Successful ===
```

Task 36:

main.js

```
1  var a = 7
2  if(a%2==0) console.log(`EVEN`)
3  else console.log(`odd`)
```

Output

```
odd

=== Code
```

Task 37:

main.js

```
1  var a = 7
2  if(a>0) console.log(`Positive`)
3  else if(a<0) console.log(`Negative`)
4  else console.log(`Zero`)
```

Output

```
Positive

=== Code
```

Task 38:

main.js

```
1  var a = 6;
2  ((a%2==0)?console.log(`Even`):console.log(`odd`) );
3
```

Output

```
Even

=== Code Execution
```

Task 39:

main.js

```
1 var a ,b=10;
2 console.log(a?? 'is undefined')
3 console.log(b?? 'is undefined')
```

Output

```
is undefined
10

=== Code Execution Successful ===
```

Task 40:

main.js

```
1 var a = 30
2 var b = 80;
3 (a>b)?a=90:b=20
4 console.log(a,b)
```

Output

```
30 20

=== Code Execution Successful ===
```

Task 41:

main.js

```
1 var a = true
2 console.log(!a)
3 var b = 30
4 var c = 40
5 console.log(b&&c)
6 console.log(b||c)
```

Output

```
false
40
30

=== Code Execution Successful ===
```

Task 42:

main.js

```
1 var a = 20
2 var b = 50
3 var c = 30
4 if((c>a) && (c<b)) console.log("in range")
```

Output

```
in range

=== Code Execution Successful ===
```

Task 43:

main.js

```
var a = true
console.log(!a)
```

Output

```
false

=== Code Execution
```

Task 44:

main.js

```
var a= true
var b= false
var c = 10
console.log(a&&c)
console.log(b&&c)
console.log(a||c)
console.log(b||c)
```

Output

```
10
false
true
10

=== Code Execution Successful ===
```

Task 45:

main.js

```
var a= true
var b= false
var c = 10
console.log(a&&c)
console.log(b&&c)
console.log(a||c)
console.log(b||c)
```

Output

```
10
false
true
10

=== Code Execution Successful ===
```

Task 46:

main.js

```
function add(a,b){
    return a+b
}
var b = 30
var c = 40
console.log(add(b,c))
```

Output

```
70

=== Code Execution Successful ===
```

Task 47:

main.js

```
function area(a,b){
    return a*b
}
var b = 30
var c = 40
console.log(area(b,c))
```

Output

```
1200

=== Code Execution Successful ===
```

Task 48:

main.js

```
function sample(){
    console.log('HEllo World')
}
sample()
```

Output

```
HEllo World

=== Code Execution Successful ===
```

Task 49:

main.js

```
function sample(){

}
console.log(sample())
```

Output

```
undefined

=== Code Execution Successful ===
```

Task 50:

main.js

```
function sample(a,b=10){
    console.log(a+b)
}
sample(1)
```

Output

```
11

=== Code Execution Successful ===
```

Task 51:

main.js

```
let greet=(name)=>{
    console.log(`Hello ${name}`);
    };
greet("jey");
```

Output

```
Hello jey

=== Code Execution Successful ===
```

Task 52:

main.js

```
let add=(num1,num2)=>{
    return num1+num2;
}

console.log(add(9,15));
console.log(add(72,83));
console.log(add(323,458));
```

Output

```
24
155
781

=== Code Execution Successful ===
```

Task 53 :

main.js

```
let isEven=(num)=>{
    return num%2==0;
}
console.log(isEven(8));
console.log(isEven(13));
```

Output

```
true
false

=== Code Execution Successful ===
```

Task 54:

main.js

```
let maxValue=(num1,num2)=>{
    return (num1>num2)?num1:num2;
}

console.log(maxValue(10,12));
console.log(maxValue(43,34));
```

Output

```
12
43

=== Code Execution Successful ===
```

Task 55:

main.js

```
const myObject = {
    value: 10,
    multiplyTraditional: function (number) {
        console.log('Traditional Function:', this.value);
        return this.value * number;
    },
    multiplyArrow: (number) => {
        console.log('Arrow Function:', this.value);
        return this.value * number;
    }
};

console.log(myObject.multiplyTraditional(2));
console.log(myObject.multiplyArrow(2));
```

Output

```
Traditional Function: 10
20
Arrow Function: undefined
NaN

=== Code Execution Successful ===
```