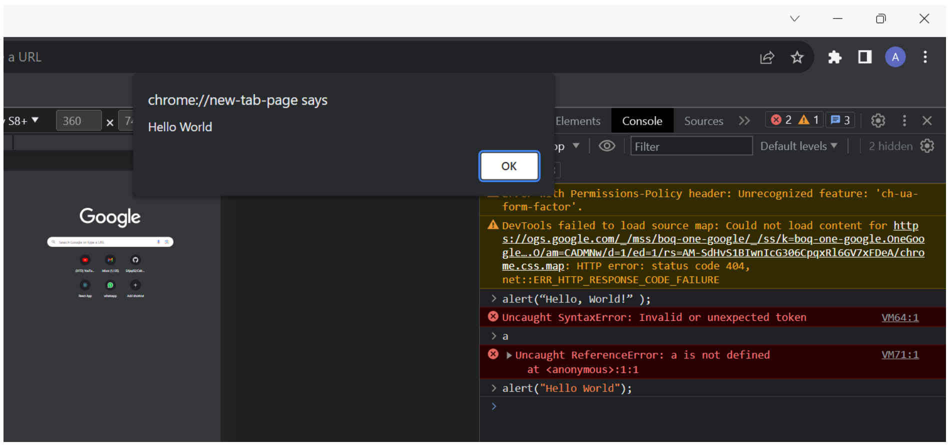
717821e202_Ajay S

Day 1: JavaScript Language - An Introduction to JavaScript, Code structure

Week-1

Task 1: Write a simple script that displays “Hello, World!” on the web page using an alert box.

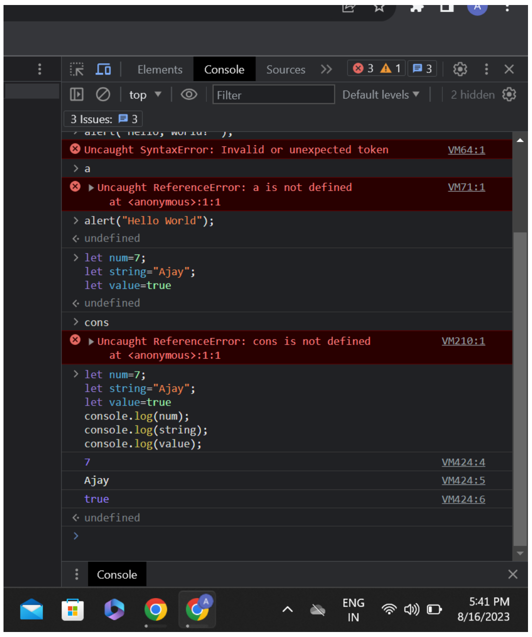


chrome://new-tab-page says

Hello World

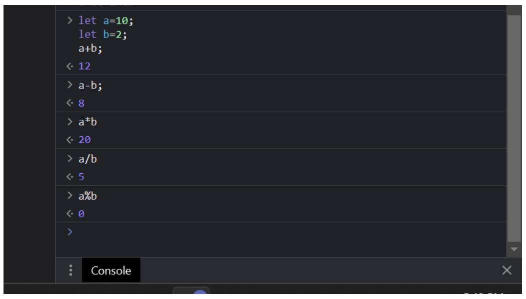
```
Error with Permissions-Policy header: Unrecognized feature: 'ch-ua-
form-factor'.
DevTools failed to load source map: Could not load content for http
s://ogs.google.com/_/mss/boq-one-google/_/ss/k=boq-one-google.OneGoo
gle….O/am=CADMNw/d=1/ed=1/rs=AM-SdHvS1BIwnIcG306CpqxRl6GV7xFDeA/chro
me.css.map: HTTP error: status code 404,
net::ERR_HTTP_RESPONSE_CODE_FAILURE
> alert(“Hello, World!” );
Uncaught SyntaxError: Invalid or unexpected token                VM64:1
> a
▸Uncaught ReferenceError: a is not defined                      VM71:1
    at <anonymous>:1:1
> alert("Hello World");
>
```

Task 2: Experiment with different data types in JavaScript (e.g., string, number, boolean) by declaring and logging them in the console.



```
Uncaught SyntaxError: Invalid or unexpected token        VM64:1
> a
▸Uncaught ReferenceError: a is not defined              VM71:1
    at <anonymous>:1:1
> alert("Hello World");
< undefined
> let num=7;
  let string="Ajay";
  let value=true
< undefined
> cons
▸Uncaught ReferenceError: cons is not defined          VM210:1
    at <anonymous>:1:1
> let num=7;
  let string="Ajay";
  let value=true
  console.log(num);
  console.log(string);
  console.log(value);
  7                                                    VM424:4
  Ajay                                                 VM424:5
  true                                                 VM424:6
< undefined
>
```

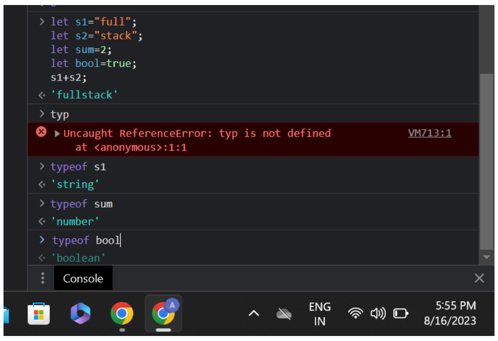
Task 3: Use the console to perform basic math operations like addition, subtraction, multiplication, and division.



```
> let a=10;
  let b=2;
  a+b;
< 12
> a-b;
< 8
> a*b
< 20
> a/b
< 5
> a%b
< 0
>
```

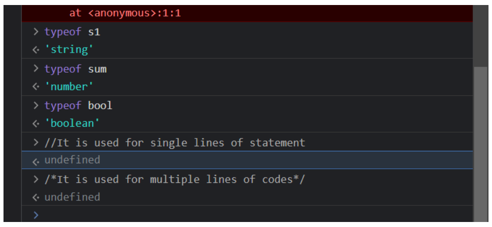
717821e202_Ajay S

Task 4: Declare two strings and concatenate them using the + operator. And
Task 5: Use the typeof operator to check the data type of various variables.



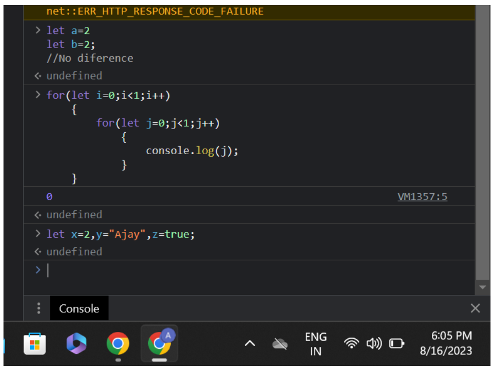
```
> let s1="full";
  let s2="stack";
  let sum=2;
  let bool=true;
  s1+s2;
< 'fullstack'
> typ
✖ ▶Uncaught ReferenceError: typ is not defined          VM713:1
      at <anonymous>:1:1
> typeof s1
< 'string'
> typeof sum
< 'number'
> typeof bool
< 'boolean'
```

Task 6: Write a multi-line JavaScript comment and a single-line comment. Explain the difference.



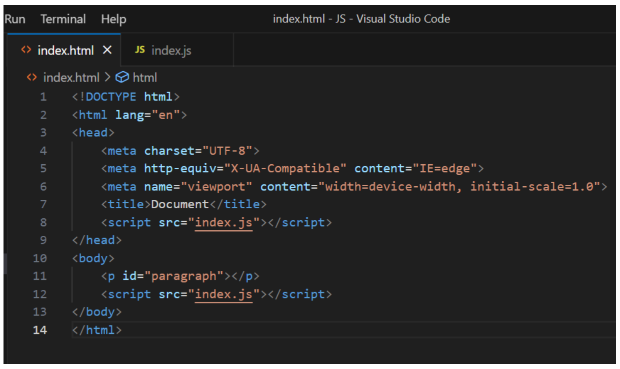
```
      at <anonymous>:1:1
> typeof s1
< 'string'
> typeof sum
< 'number'
> typeof bool
< 'boolean'
> //It is used for single lines of statement
< undefined
> /*It is used for multiple lines of codes*/
< undefined
>
```

Task 7: Create a script with both semicolon-separated and not separated lines. Note any differences in behavior.
Task 8: Use proper indentation to format a nested loop.
Task 9: Declare multiple variables in a single line.

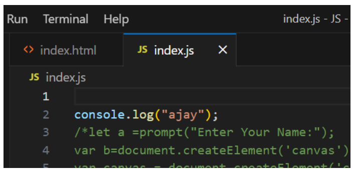


```
  net::ERR_HTTP_RESPONSE_CODE_FAILURE
> let a=2
  let b=2;
  //No diference
< undefined
> for(let i=0;i<1;i++)
      {
          for(let j=0;j<1;j++)
              {
                  console.log(j);
              }
      }
  0                                                VM1357:5
< undefined
> let x=2,y="Ajay",z=true;
< undefined
> |
```

Task 10: Place a script tag at the top and bottom of an HTML document. Note any differences in behavior.

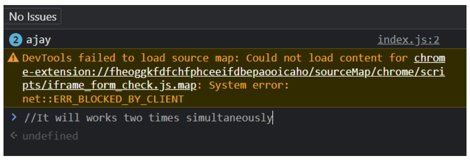


```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Document</title>
    <script src="index.js"></script>
</head>
<body>
    <p id="paragraph"></p>
    <script src="index.js"></script>
</body>
</html>
```



```

console.log("ajay");
/*let a =prompt("Enter Your Name:");
var b=document.createElement('canvas')
```

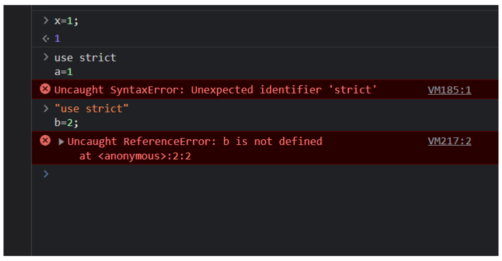
717821e202_Ajay S



```
No Issues
2 ajay                                                        index.js:2
⚠ DevTools failed to load source map: Could not load content for chrom
e-extension://fheoggkfdfchfphceeifdbepaooicaho/sourceMap/chrome/scri
pts/iframe_form_check.js.map: System error:
net::ERR_BLOCKED_BY_CLIENT
> //It will works two times simultaneously
< undefined
```

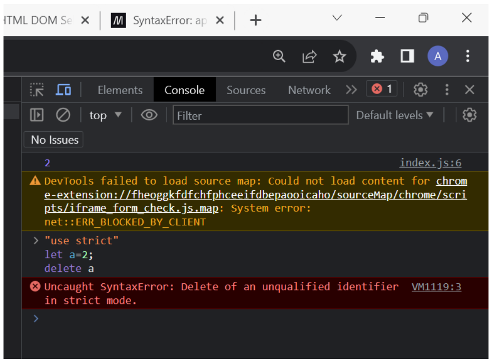
Day 2: The modern mode, “use strict”, Variables

Task 11: Write a script without using “use strict” and try to assign a value to an undeclared variable. Note the result.
Task 12: Enable “use strict” mode and repeat the above action, noting the difference.



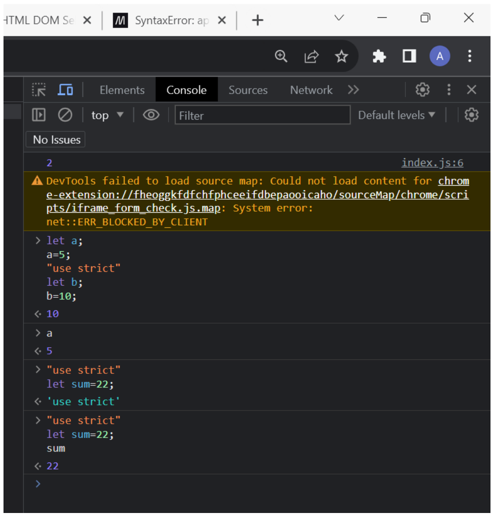
```
> x=1;
< 1
> use strict
  a=1
⊗ Uncaught SyntaxError: Unexpected identifier 'strict'       VM185:1
> "use strict"
  b=2;
⊗ ▸Uncaught ReferenceError: b is not defined                 VM217:2
     at <anonymous>:2:2
>
```

Task 13: In “use strict” mode, try to delete a variable, function, or function parameter.



```
No Issues
  2                                                           index.js:6
⚠ DevTools failed to load source map: Could not load content for chrom
  e-extension://fheoggkfdfchfphceeifdbepaooicaho/sourceMap/chrome/scri
  pts/iframe_form_check.js.map: System error:
  net::ERR_BLOCKED_BY_CLIENT
> "use strict"
  let a=2;
  delete a
⊗ Uncaught SyntaxError: Delete of an unqualified identifier  VM1119:3
  in strict mode.
>
```

Task 14: Assign a value to an undeclared variable without “use strict” and then with “use strict”.
Task 15: Declare a variable with a reserved keyword in “use strict” mode.

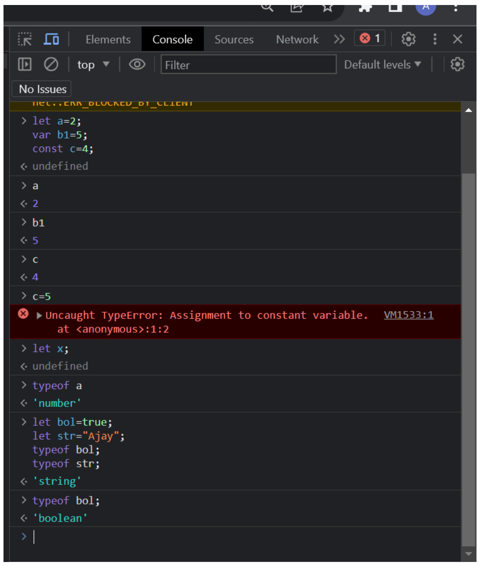


```
No Issues
  2                                                           index.js:6
⚠ DevTools failed to load source map: Could not load content for chrom
  e-extension://fheoggkfdfchfphceeifdbepaooicaho/sourceMap/chrome/scri
  pts/iframe_form_check.js.map: System error:
  net::ERR_BLOCKED_BY_CLIENT
> let a;
  a=5;
  "use strict"
  let b;
  b=10;
< 10
> a
< 5
> "use strict"
  let sum=22;
< 'use strict'
> "use strict"
  let sum=22;
  sum
< 22
>
```

Task 16: Declare variables using let, const, and var. Discuss when each should be used. o Task 17: Attempt to reassign a const variable and observe the result. o Task 18: Declare a variable without initializing it and print its value.

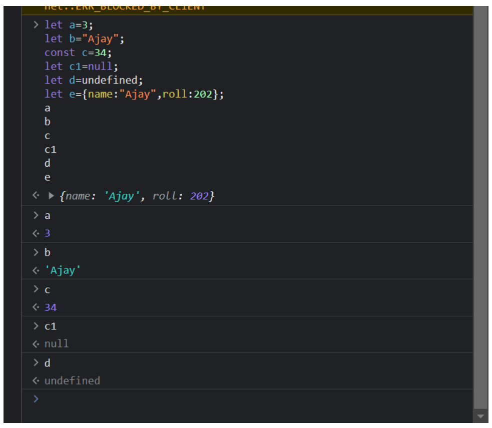
717821e202_Ajay S

Task 19: Assign a number, string, and boolean value to a variable and print its type using typeof.
Task 20: Rename a variable and observe the outcome.

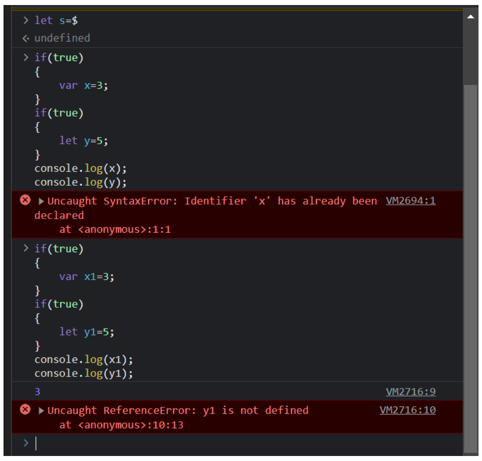


```
net::ERR_BLOCKED_BY_CLIENT
> let a=2;
  var b1=5;
  const c=4;
< undefined
> a
< 2
> b1
< 5
> c
< 4
> c=5
✖ ▶Uncaught TypeError: Assignment to constant variable.   VM1533:1
      at <anonymous>:1:2
> let x;
< undefined
> typeof a
< 'number'
> let bol=true;
  let str="Ajay";
  typeof bol;
  typeof str;
< 'string'
> typeof bol;
< 'boolean'
>
```

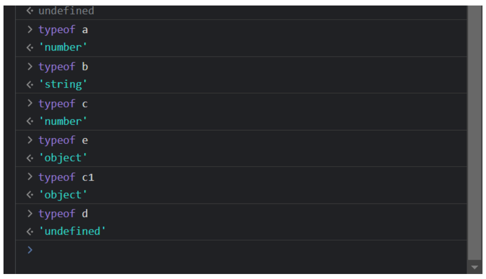
Day 3: Data types, Basic operators, maths
Task 21: Create variables of different data types (e.g., string, number, boolean, null, undefined, object).
Task 22: Use the typeof operator to determine the type of various variables. o Task 23: Declare a symbol and print its type. Task 24: Assign the value null to a variable and check its type using typeof.
Task 25: Differentiate between declaring a variable using var and let in terms of scope.



```
net::ERR_BLOCKED_BY_CLIENT
> let a=3;
  let b="Ajay";
  const c=34;
  let c1=null;
  let d=undefined;
  let e={name:"Ajay",roll:202};
  a
  b
  c
  c1
  d
  e
< ▶{name: 'Ajay', roll: 202}
> a
< 3
> b
< 'Ajay'
> c
< 34
> c1
< null
> d
< undefined
>
```



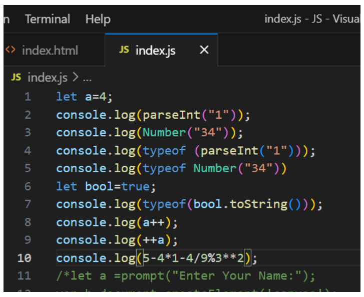
```
> let s=$
< undefined
> if(true)
  {
      var x=3;
  }
  if(true)
  {
      let y=5;
  }
  console.log(x);
  console.log(y);
✖ ▶Uncaught SyntaxError: Identifier 'x' has already been   VM2694:1
  declared
      at <anonymous>:1:1
> if(true)
  {
      var x1=3;
  }
  if(true)
  {
      let y1=5;
  }
  console.log(x1);
  console.log(y1);
  3                                                        VM2716:9
✖ ▶Uncaught ReferenceError: y1 is not defined             VM2716:10
      at <anonymous>:10:13
>
```



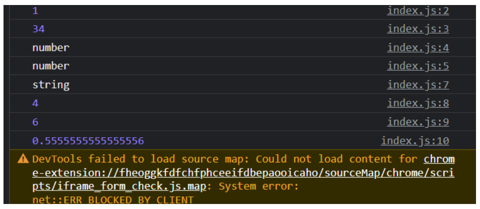
```
< undefined
> typeof a
< 'number'
> typeof b
< 'string'
> typeof c
< 'number'
> typeof e
< 'object'
> typeof c1
< 'object'
> typeof d
< 'undefined'
>
```

717821e202_Ajay S

Task 26: Convert a string to a number using both implicit and explicit conversion. Task 27: Convert a boolean to a string and vice versa.Task 29: Use the ++ and -- operators on a numeric variable.



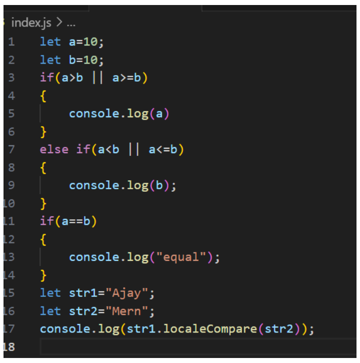
```
let a=4;
console.log(parseInt("1"));
console.log(Number("34"));
console.log(typeof (parseInt("1")));
console.log(typeof Number("34"))
let bool=true;
console.log(typeof(bool.toString()));
console.log(a++);
console.log(++a);
console.log(5-4*1-4/9%3**2);
/*let a =prompt("Enter Your Name:");
```



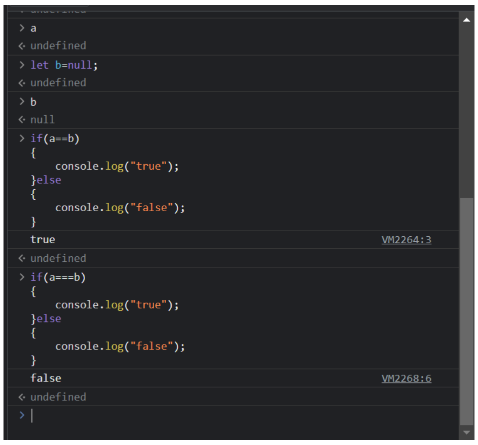
```
1                                              index.js:2
34                                             index.js:3
number                                         index.js:4
number                                         index.js:5
string                                         index.js:7
4                                              index.js:8
6                                              index.js:9
0.5555555555555556                             index.js:10
⚠ DevTools failed to load source map: Could not load content for chrom
e-extension://fheoggkfdfchfphceeifdbepaooicaho/sourceMap/chrome/scri
pts/iframe_form_check.js.map: System error:
net::ERR_BLOCKED_BY_CLIENT
```

Day 4: Comparisons, Conditional branching: if, ‘?’

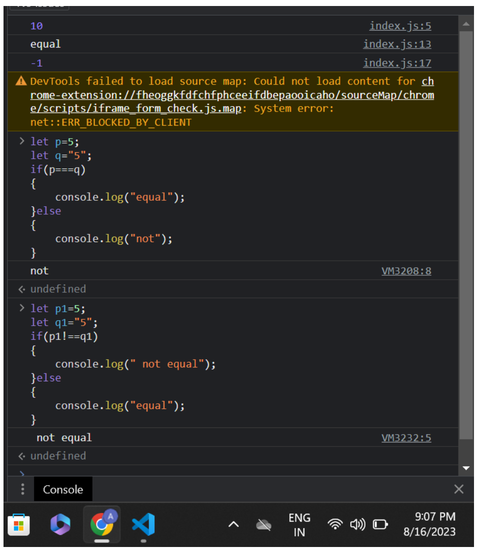
Task 31: Compare two numbers using relational operators (>, <, >=, <=). Task 32: Use equality () and strict equality (=) operators to compare different data types and note the differences. Task 33: Compare two strings lexicographically.
Task 34: Use the inequality (!=) and strict inequality (!==) operators to compare values.



```
let a=10;
let b=10;
if(a>b || a>=b)
{
    console.log(a)
}
else if(a<b || a<=b)
{
    console.log(b);
}
if(a==b)
{
    console.log("equal");
}
let str1="Ajay";
let str2="Mern";
console.log(str1.localeCompare(str2));
```



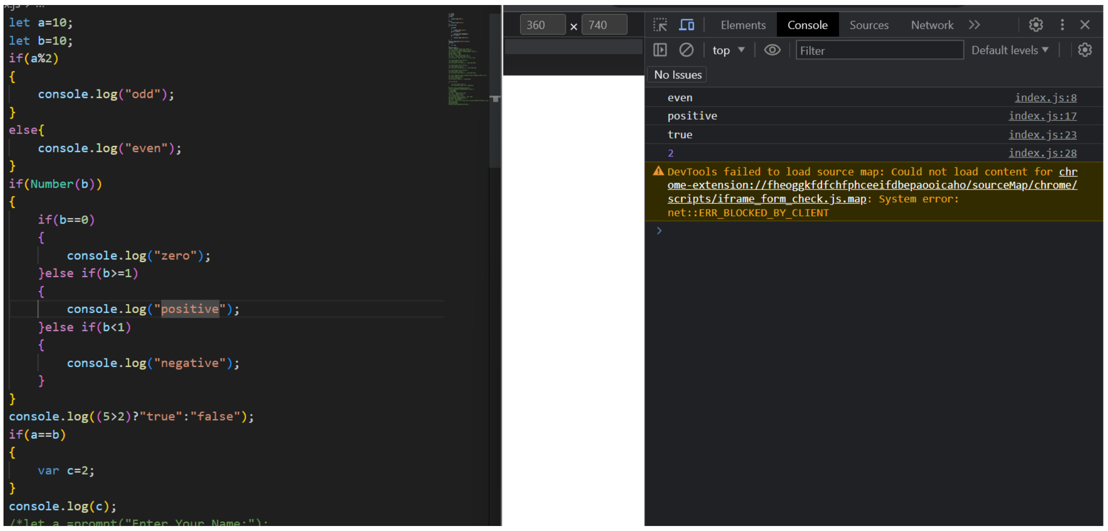
```
> a
< undefined
> let b=null;
< undefined
> b
< null
> if(a==b)
  {
      console.log("true");
  }else
  {
      console.log("false");
  }
  true                                    VM2264:3
< undefined
> if(a===b)
  {
      console.log("true");
  }else
  {
      console.log("false");
  }
  false                                   VM2268:6
< undefined
> |
```



```
10                                        index.js:5
equal                                     index.js:13
-1                                        index.js:17
⚠ DevTools failed to load source map: Could not load content for ch
rome-extension://fheoggkfdfchfphceeifdbepaooicaho/sourceMap/chrom
e/scripts/iframe_form_check.js.map: System error:
net::ERR_BLOCKED_BY_CLIENT
> let p=5;
  let q="5";
  if(p===q)
  {
      console.log("equal");
  }else
  {
      console.log("not");
  }
  not                                     VM3208:8
< undefined
> let p1=5;
  let q1="5";
  if(p1!==q1)
  {
      console.log(" not equal");
  }else
  {
      console.log("equal");
  }
   not equal                              VM3232:5
< undefined
```

717821e202_Ajay S

Task 36: Write an if statement that checks if a number is even or odd. Task 37: Use nested if statements to classify a number as negative, positive, or zero. Task 38: Use the conditional (ternary) operator '?' to rewrite a simple if…else statement. Task 39: Check the validity of a variable using the ? operator.
Task 40: Use the conditional operator to assign a value to a variable based on a condition.

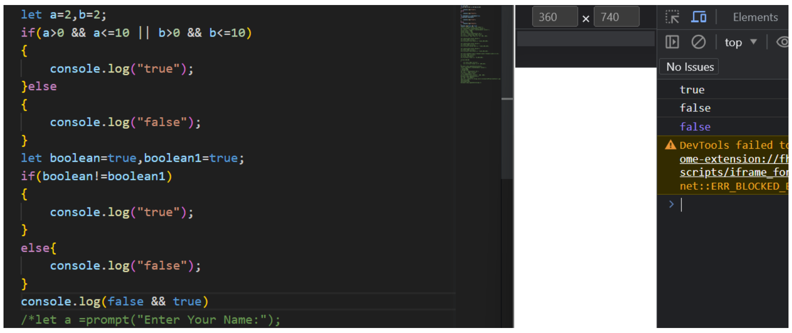


```
let a=10;
let b=10;
if(a%2)
{
    console.log("odd");
}
else{
    console.log("even");
}
if(Number(b))
{
    if(b==0)
    {
        console.log("zero");
    }else if(b>=1)
    {
        console.log("positive");
    }else if(b<1)
    {
        console.log("negative");
    }
}
console.log((5>2)?"true":"false");
if(a==b)
{
    var c=2;
}
console.log(c);
/*let a =prompt("Enter Your Name:");
```

```
No Issues
even                                                    index.js:8
positive                                               index.js:17
true                                                   index.js:23
2                                                      index.js:28
⚠ DevTools failed to load source map: Could not load content for chr
ome-extension://fheoggkfdfchfphceeifdbepaooicaho/sourceMap/chrome/
scripts/iframe_form_check.js.map: System error:
net::ERR_BLOCKED_BY_CLIENT
>
```

Day 5: Logical operators, Functions

Task 41: Evaluate various combinations of logical operators (&&, ||, !). Task 42: Use logical operators to write a condition that checks if a number is in a given range. Task 43: Use the NOT (!) operator to invert a boolean value. Task 44: Evaluate the short-circuiting nature of logical operators. Page 4 of 7 Task 45: Compare two non-boolean values using logical operators and observe the result.

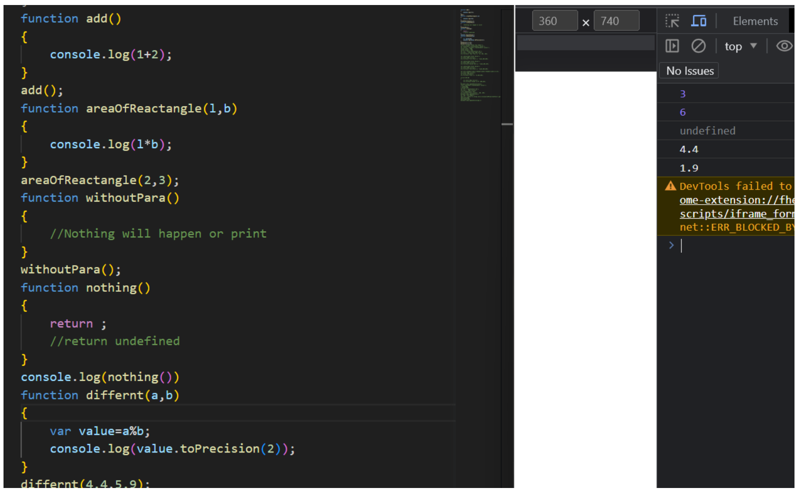


```
let a=2,b=2;
if(a>0 && a<=10 || b>0 && b<=10)
{
    console.log("true");
}else
{
    console.log("false");
}
let boolean=true,boolean1=true;
if(boolean!=boolean1)
{
    console.log("true");
}
else{
    console.log("false");
}
console.log(false && true)
/*let a =prompt("Enter Your Name:");
```

```
No Issues
true
false
false
⚠ DevTools failed to
ome-extension://fh
scripts/iframe_for
net::ERR_BLOCKED_
> |
```

Task 46: Write a function that takes two numbers as arguments and returns their sum. o Task 47: Create a function that calculates the area of a rectangle. Task 48: Declare a function without parameters and call it. Task 49: Write a function that returns nothing and observe the default return value.
Task 50: Declare a function with default parameters and call it with different arguments.

717821e202_Ajay S



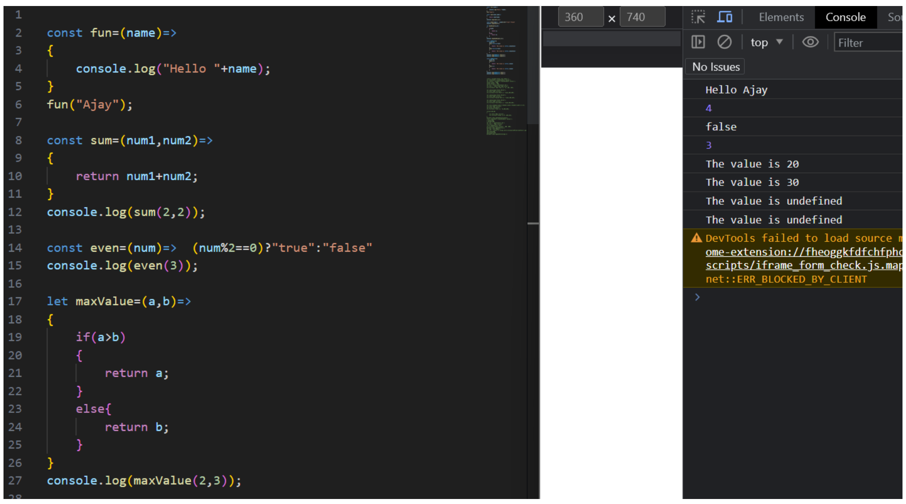
```
function add()
{
    console.log(1+2);
}
add();
function areaOfReactangle(1,b)
{
    console.log(1*b);
}
areaOfReactangle(2,3);
function withoutPara()
{
    //Nothing will happen or print
}
withoutPara();
function nothing()
{
    return ;
    //return undefined
}
console.log(nothing())
function differnt(a,b)
{
    var value=a%b;
    console.log(value.toPrecision(2));
}
differnt(4.4,5.9);
```

```
No Issues
3
6
undefined
4.4
1.9
⚠ DevTools failed to
ome-extension://fhe
scripts/iframe_form
net::ERR_BLOCKED_BY
```

Task 51: Declare a simple arrow function named greet that takes one parameter name and returns the string “Hello, name!”. Test your function with various names.

Task 52: Write an arrow function named add that takes two parameters and returns their sum. Validate your function with several pairs of numbers.

Task 53: Declare an arrow function named isEven that checks if a number is even. If the number is even, it should return true; otherwise, false. Remember that if the arrow function body has a single statement, you can omit the curly braces. Task 54: Implement an arrow function named maxValue that takes two numbers as parameters and returns the larger number. Here, you'll need to use curly braces for the function body and the return statement.

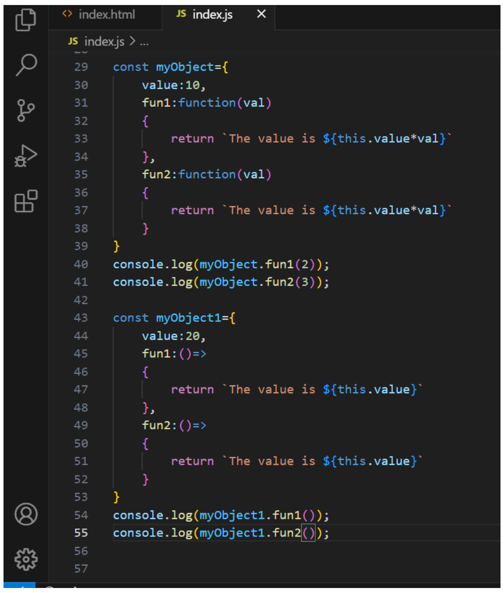


```

const fun=(name)=>
{
    console.log("Hello "+name);
}
fun("Ajay");

const sum=(num1,num2)=>
{
    return num1+num2;
}
console.log(sum(2,2));

const even=(num)=>  (num%2==0)?"true":"false"
console.log(even(3));

let maxValue=(a,b)=>
{
    if(a>b)
    {
        return a;
    }
    else{
        return b;
    }
}
console.log(maxValue(2,3));
```

```
No Issues
Hello Ajay
4
false
3
The value is 20
The value is 30
The value is undefined
The value is undefined
⚠ DevTools failed to load source m
ome-extension://fheoggkfdfchfphc
scripts/iframe_form_check.js.map
net::ERR_BLOCKED_BY_CLIENT
```

717821e202_Ajay S

Task 55: Examine the behavior of the this keyword inside an arrow function vs a traditional function. Create an object named myObject with a property value set to 10 and two methods: multiplyTraditional using a traditional function and multiplyArrow using an arrow function. Both methods should attempt to multiply the value property by a number passed as a parameter. Check the value of this inside both methods.



```javascript
const myObject={
    value:10,
    fun1:function(val)
    {
        return `The value is ${this.value*val}`
    },
    fun2:function(val)
    {
        return `The value is ${this.value*val}`
    }
}
console.log(myObject.fun1(2));
console.log(myObject.fun2(3));

const myObject1={
    value:20,
    fun1:()=>
    {
        return `The value is ${this.value}`
    },
    fun2:()=>
    {
        return `The value is ${this.value}`
    }
}
console.log(myObject1.fun1());
console.log(myObject1.fun2());
```