

1 july 2025(variable ,Datatype,operations)

Go Run ... ← → InternshipJS

... JS letvariable.js X datatypes.html task1.html condition1.html JS conditionif.js condition1.html

```
variables > JS letvariable.js
1 console.log("first program");
2 var x=4;
3 var y=5;
4 var z=x+y;
5 console.log(z);
6
7 //let
8 let a=10;
9 let b=20;
10 let c=a+b;
11 console.log(c);
12
13 //const
14 const s=2;
15 const w=3;
16 const q=s+w;
17 console.log(q);
18
19 //let const
20 let price=20;
21 let price1=30;
22 const total=price+price1;
23 console.log(total);
24
25 let p=1;
26 p=p++;
27 console.log(p);
28
29 //
30 const pi=3.14;
31 let person="john";
32 let answer="5"+5+3;
```

Ln 39, Col 27 Spaces: 4

← → InternshipJS

or.js Untitled-1 JS task.js conditionalStatement evenorodd.html printnumber.html loop.html JS task.js variables

```
variables > JS datatypes.js > ...
1 //string
2 let x=16+4+"volvo";
3 console.log(x);
4
5 let y="volvo"+16+4;
6 console.log(y);
7
8 let carname="volvo";
9 let car="BMW";
10 console.log(carname);
11 console.log(car);
12
13 //number
14
15 let A=34.00;
16 let A2=34;
17 console.log(A);
18 console.log(A2);
19
20 //bigint
21 let b=BigInt("123456789012345678901234567890");
22 console.log(b);
23
24 //Boolean
25 let c=5;
26 let d=5;
27 let e=6;
28 console.log((c==d));
29 console.log((c==e));
30
31 //Array
32 const cars=["bmw","volvo"];
33 console.log(cars[0]);
34
35 //Object
36 const persons={firstName:"john",lastName:"dao",age:23};
37 console.log(persons.firstName+" "+persons.lastName+" "+persons.age);
38
39 console.log(typeof "");
40 console.log(typeof firstName);
```

Ln 25, Col 9 Spaces: 4 UTF-8

2 july 2025(conditional statement,loop,logical operator,program:odd or even and voter age).

```
condition1.html X switch.html task.html calc.html calc.js calculator.js Untitled-1 task.js
conditionalStatement > condition1.html > html > body > script
1 <html lang="en">
2 <head>
3   <title>Document</title>
4 </head>
5 <body>
6   <p id="demo"></p>
7
8   <script>
9     const hour=new Date().getHours();
10    if(hour<18){
11      greeting="goodday";
12    }
13    else{
14      greeting="bad day";
15    }
16    document.getElementById("demo").innerHTML=greeting;
17  </script>
18 </body>
19 </html>
```

```
conditionF.html X conditionF.js condition1.html switch.html task.html calc.html calc.js
conditionalStatement > conditionF.html > html > body > script
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset="UTF-8">
5   <meta name="viewport" content="width=device-width, initial-scale=1.0">
6   <title>Document</title>
7 </head>
8 <body>
9   <p id="para">Good evening</p>
10  <script src="conditinf.js">
11
12 </script>
13 </body>
14 </html>
```

```
conditionF.html X conditionF.js condition1.html switch.html task.html calc.html calc.js
conditionalStatement > conditionF.js
1
2   if(new Date().getHours())>10){
3     document.getElementById("para").innerHTML="good day";
4   }
5   else{
6     document.getElementById("para").innerHTML="good bad evening";
7   }
8
```

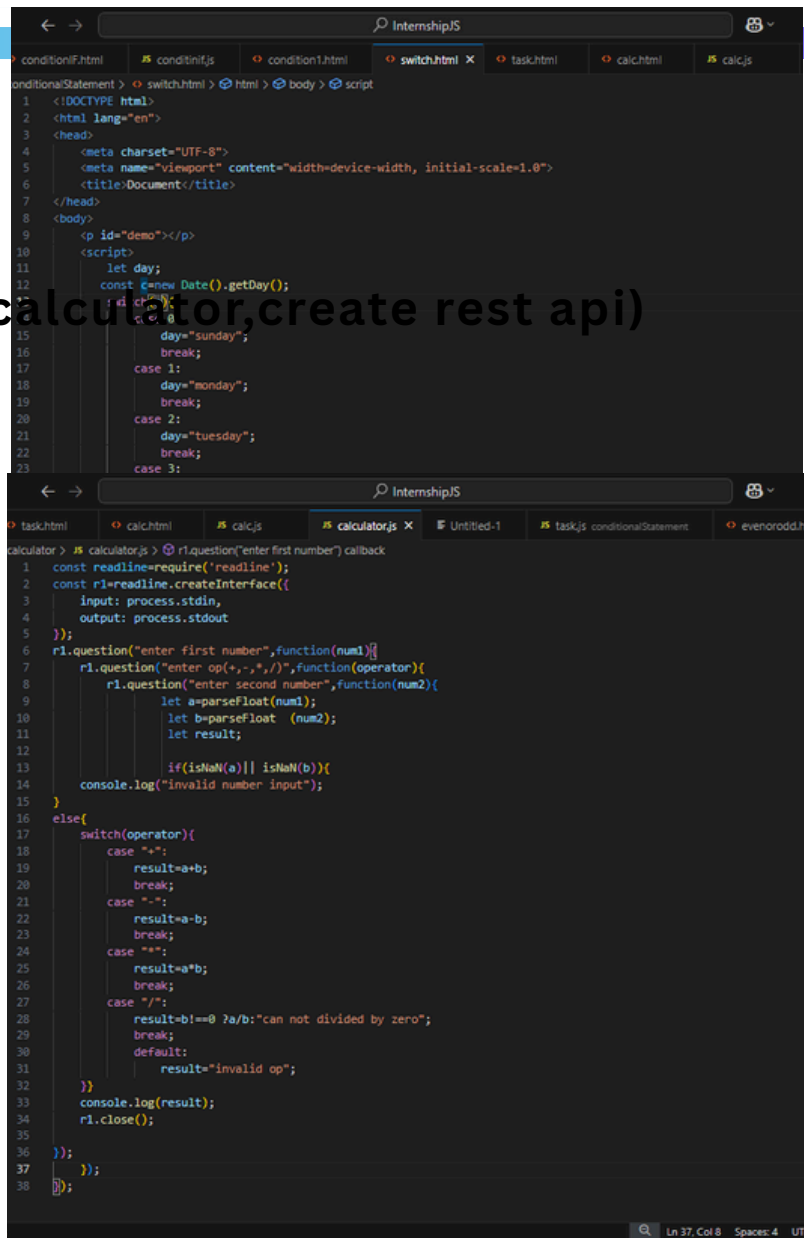
```
task.html calc.html calc.js calculator.js Untitled-1 task.js conditionalStatement evenorodd.html
conditionalStatement > loop.html > html > body > script
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset="UTF-8">
5   <meta name="viewport" content="width=device-width, initial-scale=1.0">
6   <title>Document</title>
7 </head>
8 <body>
9   <p id="demo"></p>
10
11  <script>
12    let c="";
13    const car=["bmw","toyotta"];
14    for(let i=0;i<car.length;i++){
15      c=c+car[i];
16    }
17    document.getElementById("demo").innerHTML=c;
18  </script>
19 </body>
20 </html>
```

```
← → InternshipJS
task.html × calc.html JS calc.js calculator.js Untitled-1 task.js conditionalStatement evenorodd.h
conditionalStatement > task.html > html > body > input#age
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset="UTF-8">
5   <meta name="viewport" content="width=device-width, initial-scale=1.0">
6   <title>Document</title>
7 </head>
8 <body>
9   <p id="demo">---</p>
10  <input type="number" id="age" />
11  <button onclick="display()">click</button>
12  <script src="task.js"></script>
13 </body>
14 </html>
```

```
← → InternshipJS
task.html calc.html JS calc.js calculator.js Untitled-1 task.js conditionalStatement × evenorodd.htm
conditionalStatement > JS task.js > display
1 function display(){
2   let vote;
3   const age=Number(document.getElementById("age").value);
4   if(isNaN(age)){
5     vote="input is not a number";
6   }else{
7     vote=(age<18)?"too young":"too old";
8   }
9   document.getElementById("demo").innerHTML=vote;
10 }
```

```
← → InternshipJS
JS calculator.js Untitled-1 JS task.js conditionalStatement evenorodd.html × printnumber.html loop.html
conditionalStatement > evenorodd.html > html > body > script > display
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset="UTF-8">
5   <meta name="viewport" content="width=device-width, initial-scale=1.0">
6   <title>Document</title>
7 </head>
8 <body>
9   <p id="para"></p>
10  <input id="num" type="number">
11  <button type="submit" onclick="display()">click</button>
12
13  <script>
14    function display(){
15      let num=Number(document.getElementById("num").value);
16      let y;
17      if(isNaN(num)){
18        y="not a number";
19      }
20      else{
21        y= (num%2==0) ?"even":"odd";
22      }
23      document.getElementById("para").innerHTML=y;
24    }
25  </script>
26 </body>
27 </html>
```

3 july 2025(calculator,create rest api)

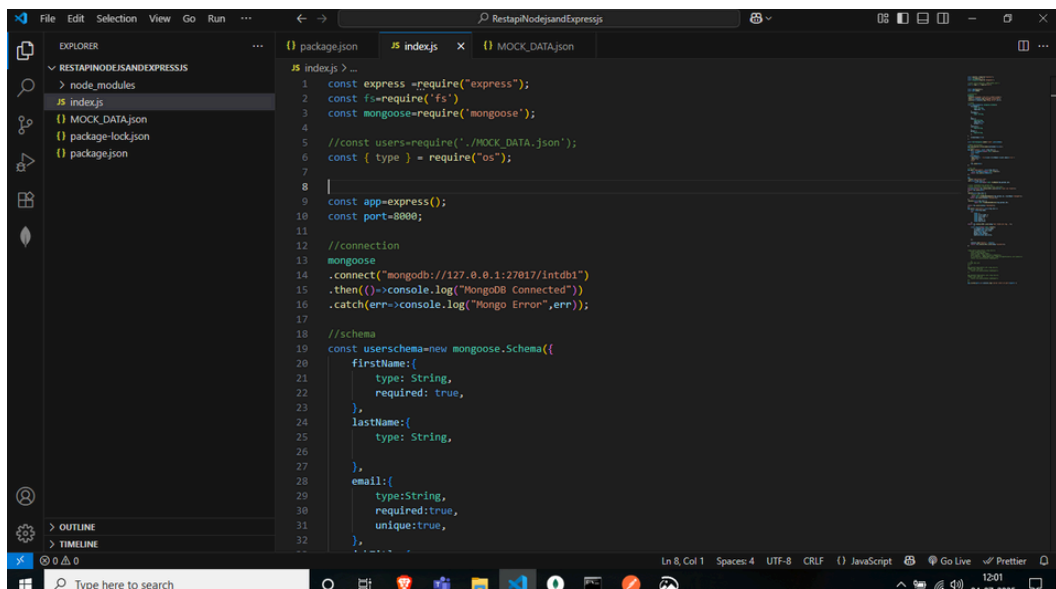


The image shows two screenshots of a VS Code editor. The top screenshot displays the `switch.html` file, which contains an HTML document with a JavaScript script. The script uses a `switch` statement to determine the day of the week based on the current date. The bottom screenshot displays the `calculator.js` file, which is a Node.js script that uses the `readline` module to create a command-line interface for a calculator. It handles basic arithmetic operations (+, -, *, /) and includes error handling for invalid inputs and division by zero.

```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset="UTF-8">
5   <meta name="viewport" content="width=device-width, initial-scale=1.0">
6   <title>Document</title>
7 </head>
8 <body>
9   <p id="demo"></p>
10  <script>
11    let day;
12    const date = new Date().getDay();
13    switch(date) {
14      case 0:
15        day = "sunday";
16        break;
17      case 1:
18        day = "monday";
19        break;
20      case 2:
21        day = "tuesday";
22        break;
23      case 3:
24        day = "wednesday";
25        break;
26      case 4:
27        day = "thursday";
28        break;
29      case 5:
30        day = "friday";
31        break;
32      case 6:
33        day = "saturday";
34        break;
35    }
36    document.getElementById("demo").innerHTML = day;
37  </script>
38 </body>
39 </html>
```

```
1 const readline = require('readline');
2 const rl = readline.createInterface({
3   input: process.stdin,
4   output: process.stdout
5 });
6 rl.question("enter first number", function(num1) {
7   rl.question("enter op(+, -, *, /)", function(operator) {
8     rl.question("enter second number", function(num2) {
9       let a = parseFloat(num1);
10      let b = parseFloat(num2);
11      let result;
12
13      if (isNaN(a) || isNaN(b)) {
14        console.log("invalid number input");
15      } else {
16        switch(operator) {
17          case "+":
18            result = a + b;
19            break;
20          case "-":
21            result = a - b;
22            break;
23          case "*":
24            result = a * b;
25            break;
26          case "/":
27            result = a / b;
28            if (b === 0) {
29              result = "can not divided by zero";
30            }
31            break;
32          default:
33            result = "invalid op";
34        }
35      }
36      console.log(result);
37      rl.close();
38    });
39  });
40 });
```

4 july 2025(Mongo db integration+create rest api)



The image shows a screenshot of a VS Code editor with the `index.js` file open. The code sets up an Express.js application and integrates it with MongoDB using Mongoose. It defines a `users` schema with fields for `firstName`, `lastName`, and `email`. The application is configured to listen on port 8000 and connect to a MongoDB instance.

```
1 const express = require('express');
2 const fs = require('fs');
3 const mongoose = require('mongoose');
4 //const users = require('./MOCK_DATA.json');
5 const { type } = require('os');
6
7
8
9 const app = express();
10 const port = 8000;
11
12 //connection
13 mongoose
14 .connect("mongodb://127.0.0.1:27017/intdb1")
15 .then(() => console.log("MongoDB Connected"))
16 .catch(err => console.log("Mongo Error", err));
17
18 //schema
19 const userschema = new mongoose.Schema({
20   firstName: {
21     type: String,
22     required: true,
23   },
24   lastName: {
25     type: String,
26   },
27   email: {
28     type: String,
29     required: true,
30     unique: true,
31   },
32 });
```

```
RestapiNodejsandExpressjs

package.json JS indexjs x MOCK_DATA.json

JS indexjs > ...
19 const userschema=new mongoose.Schema({
28   email:{
30     required:true,
31     unique:true,
32   },
33   jobTitle:{
34     type:String,
35   },
36   gender:{
37     type:String,
38   },
39 },
40 {timestamps:true}
41 );
42
43 const User=mongoose.model('user',userschema);
44
45 //mddle ware-plugin
46 app.use(express.urlencoded({extended:false}));
47 //routers
48 app.get("/users", async (req,res)=>{
49   const allDbUsers=await User.find({});
50   const html=`
51   <ul>
52   ${allDbUsers
53     .map((user)=> `<li>${user.firstName}-${user.email}</li>`)
54     .join("")}
55   </ul>
56   `;
57   res.send(html);
58 })
59
60 //next api
```

```
RestapiNodejsandExpressjs

EXPLORER
RESTAPINODEJSANDEXPRESSJS
  > node_modules
  JS indexjs
  () MOCK_DATA.json
  () package-lock.json
  () package.json

JS indexjs > ...
59 //rest api
60 app.get("/api/users", async(req,res)=>{
61   const allDbUsers=await User.find({});
62   return res.json(allDbUsers);
63 });
64
65
66 app
67 .route("/api/users/:id")
68 .get(async(req,res)=>{
69   const user=await User.findById(req.params.id);
70
71   //const id=Number(req.params.id);
72   //const user=users.find((user)=>user.id===id);
73   if(!user)return res.status(404).json({error:"user not found"});
74   return res.json(user);
75 });
76 .patch(async(req,res)=>{
77   //edit user with id
78   await User.findByIdAndUpdate(req.params.id, {lastName:"changed"});
79   return res.json({status:"success"});
80 });
81 .delete(async(req,res)=>{
82   await User.findByIdAndDelete(req.params.id);
83
84   return res.json({status:"deleted"});
85 });
86 app.post("/api/users",async(req,res)=>{
87   const body=req.body;
88   if(
89     !body ||
90     !body.first_name ||
```

