

DAY - 2

DESTRUCTURING

</> Java

```
1 const person={name:'sai',roll:34,gender:'male'}
2 let {'n':name,'r':roll,'g':gender} = person //n as name r as roll --- we are
  assigning the aliance name
3 console.log(name," ",roll," ",gender)
4 console.log(person)
```

REST OPERATOR

</> Java

```
1 const person={name:'sai',roll:34,gender:'male'}
2
3 let {name,...restdatas}=person
4 console.log(name,restdatas)
```

JAVA EXTRA'S

</> Java

```
1 public static void main(String args[]){
2     public void add(int ...a){
3         int s=0;
4         for (int i=0;i<a.length;i++){
5             s=s+a[i];
6         }
7     }
8     add(10,20,30,40,50,60)
9     //known as var arg component that ...a acts as an array and stores all the
  values in it
```

OPERATORS

</> Java

```
1 let a=[1,2,3,4]
2 let b=['hai','hello','welcome',a]
3 console.log('added the array: ',b) //[ 'hai', 'hello', 'welcome', [ 1, 2, 3,
  4 ] ]
4 b=['hai','hello','welcome',...a]
5 console.log(b) //[ 'hai', 'hello', 'welcome', 1, 2, 3, 4 ]
```

FUNCTIONS

1. No argument no return

</> Java

```
1 function add(){
2     console.log('Welcome')
3 }
4 console.log('Hai')
5 add() //o/p Hai Welcome
```

2. With Argument without return type

```

1 function add(a,b){
2     c=a+b
3 }
4 console.log('Hai')
5 add(10,20)

```

</> Java

3. With argument with return type

```

1 function add(...a){
2     s=0
3     for(let i=0;i<a.length;i++){
4         s=s+a[i]
5     }
6     return s
7 }
8 console.log('Hai')
9 console.log(add(10,20))
10 console.log(add(10,20,30,40,50))
11 // ...a,b throws error
12 // a,...b won't throw error

```

</> Java

4. Without argument with return type

```

1 function add(){
2     return 'Welcome'
3 }
4 console.log('Hai')
5 add()

```

</> Java

Array destructuring

```

1 let a=[1,2,3,34]
2 let [p,q,r,s]=a
3
4 console.log(p)
5 console.log(q)
6 console.log(r)
7 console.log(s)

```

</> Java

```

1 let a=[10,20,30,40,50,60,70,80,90]
2 let [p,q,...restdata]=a
3
4 console.log(p)
5 console.log(q)
6 console.log(restdata,typeof(restdata))
7 let {r1,r2}=restdata
8 console.log(r1)
9 console.log(r2)

```

</> Java

closures

</> Java

```

1 function one(){
2     function inner(){
3         console.log("I'm inner the function's function")
4     }
5     console.log("I'm inner the function")
6     inner()
7 }
8 one()

```

</> Java

```

1 function one(){
2     return function(){
3         console.log("I'm inner the function's function")
4     }
5     console.log("I'm inner the function")
6 }
7 console.log(one()) //[Function (anonymous)]
8
9 function one(){
10    console.log("I'm inner the function")
11    return function(){
12        console.log("I'm inner the function's function")
13    }
14 }
15 one() //I'm inner the function

```

</> Java

```

1 let func=function out(){
2     console.log("I'm outside")
3     return function(){
4         console.log("I'm inside")
5     }
6 }
7 console.log(func())
8 //op I'm outside
9 //[Function (anonymous)]

```

ARROW FUNCTION

No need to mention the function name

1. Without arg without return type

</> Java

```

1 let a=()=>{
2     console.log('welcome')
3 }
4 a() //welcome

```

2. with arg with return

</> Java

```

1 let a=(a,b)=>{
2     c=a+b
3     console.log(c)
4 }
5 a(10,20) //30

```

```
6 a(20,30) //50
```

3. with arg without return type

```
1 let a=(x,y)=>x+y
2 console.log(a(1,2)) //3
```

</> Java

TIMEOUT

Asynchronous

```
1 function get(recCheckFun){
2     name='sai'
3     setTimeout( ()=>{
4         if (name==='sai'){
5             recCheckFun()
6         }
7     } ,2000)
8 }
9 function check(){
10     console.log('pass')
11 }
12 get(check) //pass
```

</> Java

Synchronized

```
1 function get(recCheckFun){
2     name='sai'
3     if (name ==='sai'){
4         recCheckFun()
5     }
6     console.log("I'm in get")
7 }
8 function check(){
9     console.log('pass')
10 }
11 get(check) /* op pass
12     I'm in get
13 */
```

</> Java

PROMISE STATE

1. Pending state
2. Resolved state
3. Reject state

```
1 let r=new Promise((resolve,reject)=>{
2     name='sai'
3     setTimeout(()=>{
4         if (name==='sai'){
5             resolve(name)
6         }
7         else{
8             reject('No data')
9         }
10     } ,2000)
11 })
12 r
```

</> Java

```

13 .then((name)=>{
14     console.log("recieve : ",name)
15 })
16 .catch((errr)=>{
17     console.log('pd is: ',errr)
18 })
19 .finally(()=>{
20     console.log("I will always execute")
21 })

```

</> Java

```

1  const res = () => {
2      return new Promise((resolve, reject) => {
3          setTimeout(() => {
4              const a = 'sai1';
5              if (a==='sai') {
6                  resolve(a);
7              } else {
8                  reject(new Error('no data'));
9              }
10             }, 1000);
11         });
12     };
13     const handleData = async () => {
14         try {
15             const name = await res(); // Wait for the promise to resolve
16             console.log('received', name); // Handle resolved value
17         } catch (err) {
18             console.log(err.stack); // Handle error stack
19         } finally {
20             console.log('received'); // Final message
21         }
22     };
23     handleData(); // Call the async function

```

HTML

WEB PAGE VALIDATION

sample.html

</> Java

```

1  <!doctype html>
2  <head>
3      <title>Sample</title>
4  </head>
5  <body>
6      <form action="sample.html" id="frmstureg" name="frmstureg"
onsubmit="return validate()" onreset="return resetForm()">
7      Student name:
8      <input type="text" id="tb1" placeholder="Enter the name"/>
9      <br>
10     <br>
11     <span id="namevalid"></span><br><br>
12     Password
13     <input type="password" id="tb2" placeholder="Enter your password"/>
14     <br>
15     <br>
16     <span id="pasvalid"></span><br><br>
17     Confirm Password

```

```

18 <input type="password" id="tb3" placeholder="Re-enter your password"/>
19 <br>
20 <br>
21 <span id="conpasvalid"></span><br><br>
22 Roll Number:
23 <input type="number" id="tb4" placeholder="Enter the Roll Number"/>
24 <br>
25 <br>
26 <span id="rollvalid"></span><br><br>
27 Gender:
28 <br>
29 <input type="radio" id="bt1"/>Male<br>
30 <input type="radio" id="bt2"/>Female
31 <br>
32 <br>
33 <span id="gendervalid"></span><br><br>
34 Qualification
35 <select id="Qualification">
36     <option value="n1">--select--</option>
37     <option value="n2">B.Tech</option>
38     <option value="n3">B.E</option>
39     <option value="n4">B.Arch</option>
40     <option value="n5">B.Sc</option>
41 </select>
42 <br>
43 <br>
44 <span id="qualvalid"></span><br><br>
45 <input type="submit" value="SUBMIT"/>
46 <input type="reset" value="RESET"/>
47 </form>
48 <script src="scri.js"></script>
49 </body>
50 </html>

```

scri.js

```

1 function validate(){
2     //text box 1
3     if((document.getElementById("tb1")).value==""){
4         document.getElementById("namevalid").innerHTML="plz enter the name";
5         document.getElementById("namevalid").style.color="red";
6     }
7     else{
8         document.getElementById("namevalid").innerHTML="valid"
9         document.getElementById("namevalid").style.color="green"
10    }
11    // text box 2
12    if((document.getElementById("tb2")).value==""){
13        document.getElementById("pasvalid").innerHTML="plz enter the
password";
14        document.getElementById("pasvalid").style.color="red";
15    }
16    else{
17        document.getElementById("pasvalid").innerHTML="valid"
18        document.getElementById("pasvalid").style.color="green"
19    }
20    //text box 3
21    if((document.getElementById("tb3")).value==""){
22        document.getElementById("conpasvalid").innerHTML="plz enter the name";
23        document.getElementById("conpasvalid").style.color="red";
24    }
25    else
26    if(document.getElementById("tb3").value!=document.getElementById("tb2").value)
    {
        document.getElementById("conpasvalid").innerHTML="Password mismatch

```

```
occured"
27     document.getElementById("conpasvalid").style.color="red"
28 }
29 else{
30     document.getElementById("conpasvalid").innerHTML="valid"
31     document.getElementById("conpasvalid").style.color="green"
32 }
33 //buttons
34 if (document.getElementById("bt1").checked!=true &&
document.getElementById("bt2").checked!=true){
35     document.getElementById("gendervalid").innerHTML="plz choose the
gender";
36     document.getElementById("gendervalid").style.color="red";
37 }
38 else
39 {
40     document.getElementById("gendervalid").innerHTML="valid";
41     document.getElementById("gendervalid").style.color="green";
42 }
43 // qualification
44 if (document.getElementById("Qualification").value="n1"){
45     document.getElementById("qualvalid").innerHTML="plz select the
qualification";
46     document.getElementById("qualvalid").style.color="red"
47 }
48 else{
49     document.getElementById("qualvalid").innerHTML="valid";
50     document.getElementById("qualvalid").style.color="green";
51 }
52 }
53 function resetForm(){
54     document.getElementById("namevalid").innerHTML="";
55     document.getElementById("pasvalid").innerHTML="";
56     document.getElementById("conpasvalid").innerHTML="";
57     document.getElementById("bt1").innerHTML="";
58     document.getElementById("bt2").innerHTML="";
59     document.getElementById("gendervalid").innerHTML="";
60     document.getElementById("qualvalid").innerHTML="";
61 }
```