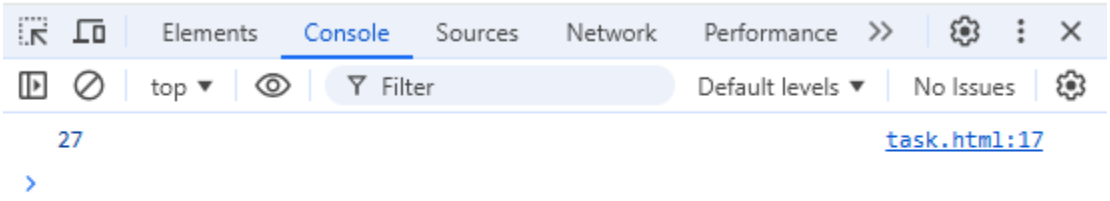
JSON VARIABLE LENGTH ARGUMENTS/SPREAD SYNTAX

TASK-1

```
<!DOCTYPE html>
<html>
  <head>

  </head>
  <body>
    <script>
      function sum(...arguments)
      {
        let total = 0;
        for(let num of arguments)
      {
        total+=num;
      }
      return total;
      }
      console.log(sum(1,7,8,6,5));
    </script>
  </body>
</html>
```

OUTPUT:

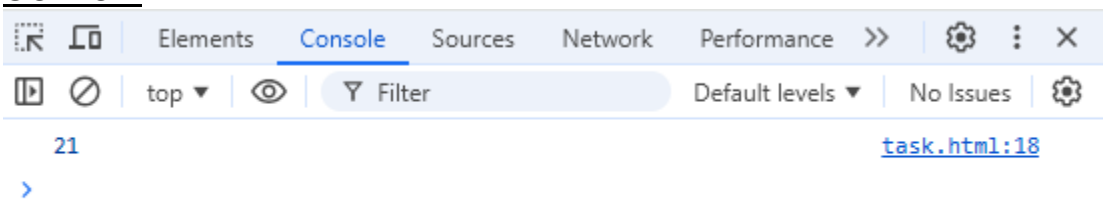


TASK-2

```
<!DOCTYPE html>
<html>
  <head>
```

```
  </head>
  <body>
    <script>
      function sum(...arguments)
      {
        let total = 0;
        for(let num of arguments)
      {
        total+=num;
      }
      return total;
      }
      const arr = [1,6,7,7]
      console.log(sum(...arr));
    </script>
  </body>
</html>
```

OUTPUT:



TASK-3

```
<!DOCTYPE html>
<html>
  <head>

  </head>
  <body>
    <script>
      function deepclone(obj)
      {
```

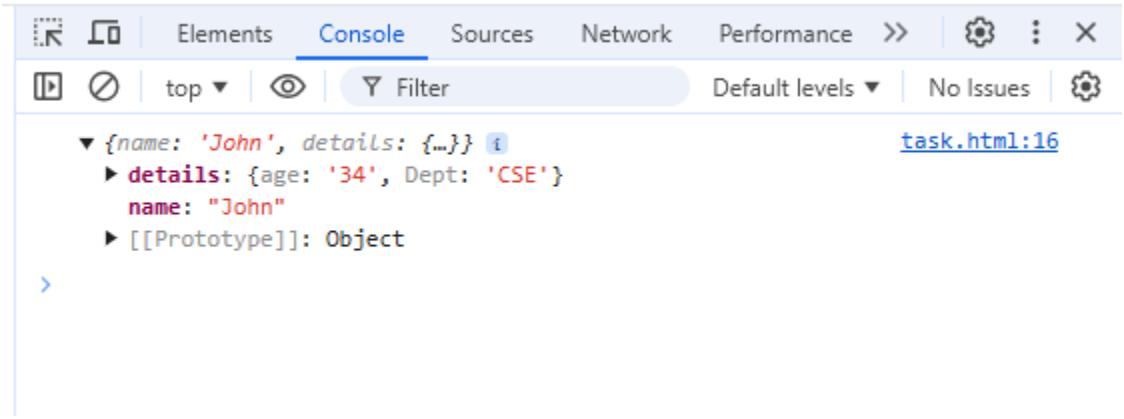
```
        return JSON.parse(JSON.stringify(obj));
      }

      const originalobj = {name:"John",details:{age: "34",Dept : "CSE"}};
      const cloneobj = deepclone(originalobj);

      console.log(cloneobj);


    </script>
  </body>
</html>
```

OUTPUT:



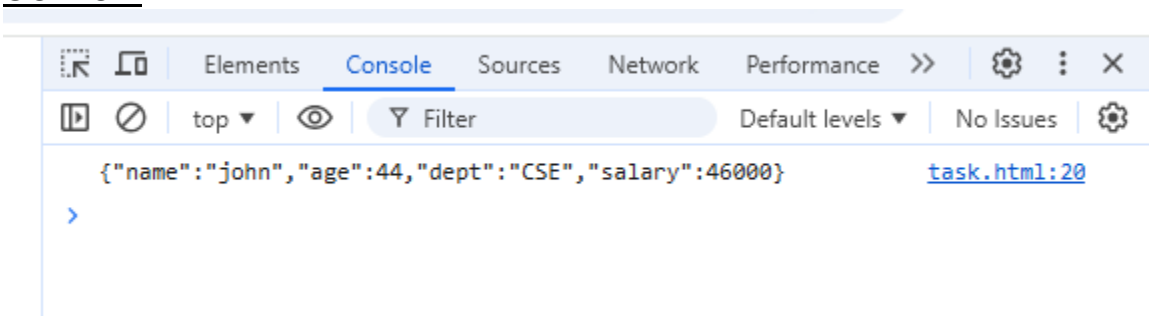
TASK-4

```
<!DOCTYPE html>
<html>
  <head>

  </head>
  <body>
    <script>
      let student =
      {
        name:"john",
```

```
            age:45,
            dept:"CSE"
        };
        let Employee = {
            name:"john",
            age:44,
            salary:46000
        };
        let obj = {...student,...Employee};
        console.log(JSON.stringify(obj));
    </script>
  </body>
</html>
```

OUTPUT:



TASK-5

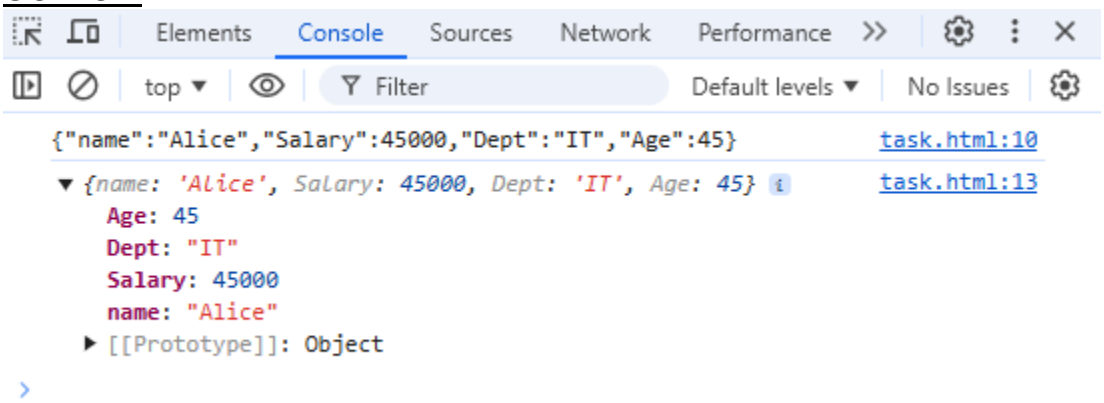
```
<!DOCTYPE html>
<html>
  <head>

  </head>
  <body>
    <script>
        const obj = {name:"Alice",Salary:45000,Dept:"IT",Age:45};
        let jsonString = JSON.stringify(obj);
        console.log(jsonString);
```

```
        let parseobj = JSON.parse(jsonString);
        console.log(parseobj);

      </script>
  </body>
</html>
```

OUTPUT:



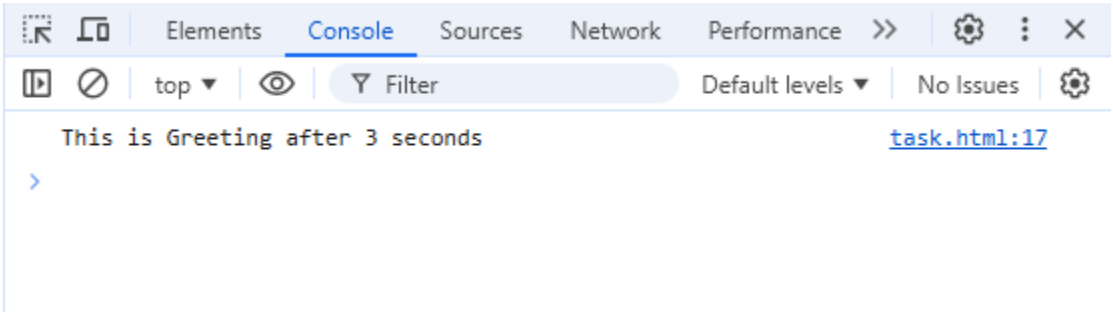
PROMISES,PROMISES CHAINING
TASK-1

```
<!DOCTYPE html>
<html>
  <head>

  </head>
  <body>
    <script>
      function greetafter(seconds)
      {
        return new Promise((resolve)=> {
          setTimeout(() => {
            resolve("This is Greeting after "+ seconds + " seconds");
          },seconds * 1000);
        });
      }
```

```
        greetafter(3).then((greeting) => console.log(greeting));
      </script>
   </body>
</html>
```

OUTPUT:

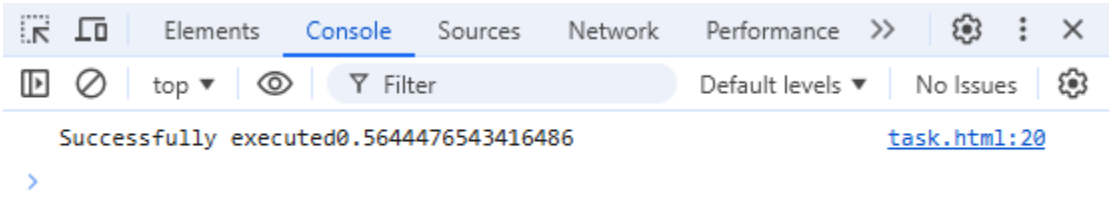


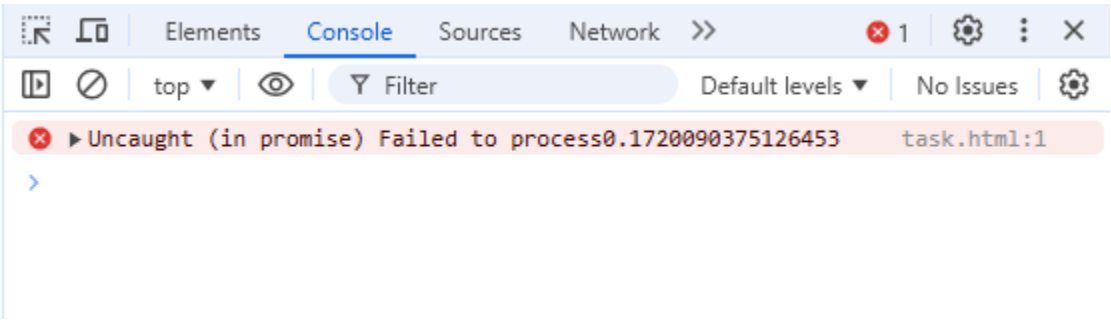
TASK-3

```
<!DOCTYPE html>
<html>
   <head>

   </head>
   <body>
      <script>
         function RandomPromise(){
            return new Promise((resolve,reject) => {
               const RandomNumber = Math.random();
               if(RandomNumber > 0.5)
            {
               resolve("Successfully executed" +RandomNumber);
            }
            else{
               reject("Failed to process" +RandomNumber);
            }
            });
         }
         RandomPromise().then((message)=>console.log(message));
         </script>
</html>
```

OUTPUT:





PROMISE.ALLSETTLED PROGRAM

```
<!DOCTYPE html>
<html>
  <head>

  </head>
  <body>
    <script>
      function ReachHome(name,timeToReach,isSuccessful){
        return new Promise((resolve,reject) =>
          setTimeout(() => {
           if(isSuccessful)
        {
          resolve(`${name} safely reach the home`);
        }
        else{
          reject(`${name} not reach the Home`);
        }
```

```
    }),timeToReach
  )}
    let friend1 = ReachHome("Arun",1000,true);
    let friend2 = ReachHome("john",5000,true);
    let friend3 = ReachHome("sam",8000,false);

    Promise.allSettled ([friend1,friend2,friend3]).then ((response) =>
  {
    response.forEach((res) => {
      document.write(res.status=="fulfilled"?res.value:res.reason);
      document.write("<br>");
    })
  })

      </script>
      </body>
</html>
```

OUTPUT: