TASK 51:

```
<!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>
    const greet=(subject)=>{
        return `Welcome to, ${subject}`;
    }
document.writeln(greet("JavaScript"));
  </script>
</body>
</html>
```

OUTPUT:

TASK 52:

```
<!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>
    const add=(num1,num2)=>{
        return num1+num2 ;
    }
document.writeln(add(5,6));
  </script>
</body>
</html>
```

OUTPUT:

```
11
```

TASK 53:

```
<!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>
    const iseven=(num1)=>{
        return (num1%2==0) ;
    }
    let num1=prompt("Enter number")*1;
document.writeln(iseven(num1));
  </script>
</body>
</html>
```

OUTPUT:

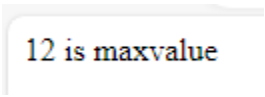
```
false
```

TASK 54:

```
<!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>
    const maxvalue=(num1,num2)=>{
        if(num1>num2){
            return `${num1} is maxvalue`;
        }
        return `${num2} is maxvalue`;
    }
```

```
    let num1=prompt("Enter number")*1;
    let num2=prompt("Enter number")*1;
document.writeln(maxvalue(num1,num2));
  </script>
</body>
</html>
```

OUTPUT:



TASK 55:

```
<!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>
  const myObject = {
    value: 10,
    Traditional: function(factor) {
      console.log("Traditional function,", this.value);
      return this.value * factor;
    },
    Arrow: (factor) => {
      console.log("Arrow function,", this.value);
      return this.value * factor;
    } };
  console.log(myObject.Traditional(2));
  console.log(myObject.Arrow(2));
  </script>
</body>
</html>
```

OUTPUT:

```
Traditional function, 10
20
Arrow function, undefined
NaN
```