

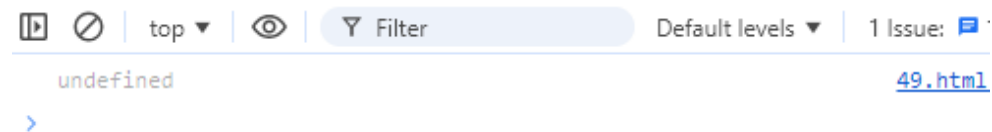
### Task 49:

Write a function that returns nothing and observe the default return value.

Code:

```
<html>
  <head>
    <title>Task</title>
  </head>
  <body>
    <script>
      function test(){
        return ;
      }
      let result= test();
      console.log(result);
    </script>
  </body>
</html>
```

Output:



undefined 49.html

### Task 50:

Declare a function with default parameters and call it with different arguments.

Code:

```
<html>
  <head>
    <title>Task</title>
  </head>
  <body>
    <script>
      function test(a,b){
        console.log(a,b);
      }
      var a="john";
      var b=20;
      test();
      test("alex",18);
      test("kayal",18);
      test (25);
    </script>
  </body>
</html>
```

Output:

undefined undefined	<a href="#">50</a>
alex 18	<a href="#">50</a>
kayal 18	<a href="#">50</a>
25 undefined	<a href="#">50</a>

### 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.

Code:

```
<html>
  <head>
    <title>Task</title>
  </head>
  <body>
    <script>
      let greet=(a)=>{
        console.log(`hello, ${a}`);
      }
      var a;
      greet("john!");
      greet("alex!");
    </script>
  </body>
</html>
```

Output:

top	Filter	Default levels	1 Issue: 1
"hello, john!"	<a href="#">51.html</a>		
"hello, alex!"	<a href="#">51.html</a>		

### Task 52:

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

Code:

```

<html>
  <head>
    <title>Task</title>
  </head>
  <body>
    <script>
      let add=(a,b)=>{
        c=a+b;
        console.log(c);
      }
      var a,b,c;
      add(10,37);
      add(5,29);
      add(67,82);
      add(26,48);
    </script>
  </body>
</html>

```

Output:

47	<a href="#">52.h</a>
34	<a href="#">52.h</a>
149	<a href="#">52.h</a>
74	<a href="#">52.h</a>
>	

### 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 arrow function body has a single statement, you can omit the curly braces.

Code:

```

<html>
  <head>
    <title>Task</title>
  </head>
  <body>
    <script>
      let isEven=(a)=>{
        if(a%2==0)
          return true;
        else
          return false;
      }
      var a;
      console.log(isEven(3));
      console.log(isEven(52));
      console.log(isEven(0));
    </script>
  </body>
</html>

```

Output:

false	<a href="#">53.html:</a>
true	<a href="#">53.html:</a>
true	<a href="#">53.html:</a>

#### 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.

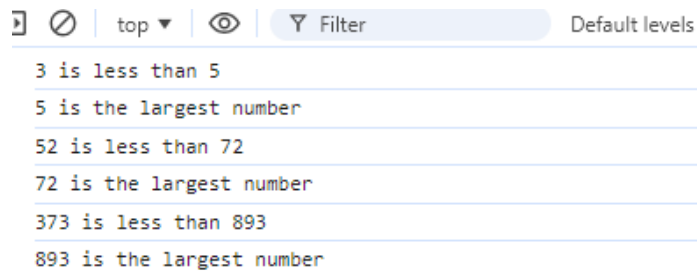
Code:

```

<html>
  <head>
    <title>Task</title>
  </head>
  <body>
    <script>
      let maxValue=(a,b)=>{
        if(a<b){
          console.log(a+" is less than "+ b);
          console.log( b+" is the largest number");
        }
        else if(a>b){
          console.log(a+" is greater than "+ b);
          console.log(a+" is the largest number");
        }
      }
      var a,b;
      maxValue(3,5);
      maxValue(52,72);
      maxValue(373,893);
    </script>
  </body>
</html>

```

Output:



```
3 is less than 5
5 is the largest number
52 is less than 72
72 is the largest number
373 is less than 893
893 is the largest number
```





### Task 55:

Examine the behavior of 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.

Code:

```
<html>
  <head>
    <title>Task</title>
  </head>
  <body>
    <script>
      const myObject= {
        value:10,
        multiplyTraditional:function(a){
          console.log("Traditional function multiplication");
          return this.value*a;
        },
        multipleArrow: (a)=>{
          console.log("Arrow function multiplication");
          return this.value*a;
        }
      };
      console.log(myObject.multiplyTraditional(23));
      console.log(myObject.multipleArrow(73));
    </script>
  </body>
</html>
```

Output:

  top ▼  Filter	Default levels ▼	1 Issue:  1
Traditional function multiplication		<a href="#">55.html:</a>
230		<a href="#">55.html:</a>
Arrow function multiplication		<a href="#">55.html:</a>
NaN		<a href="#">55.html:</a>