Hositing function :-

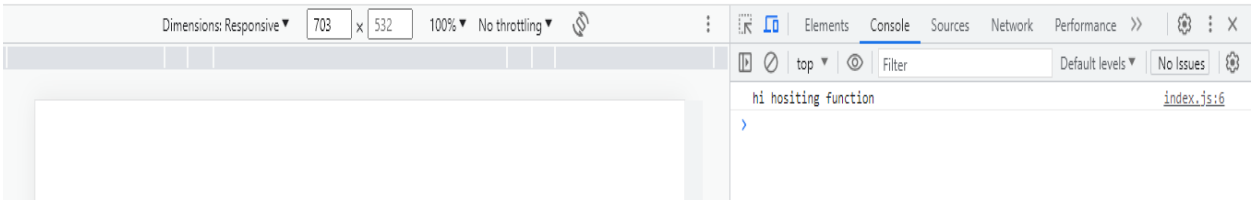
Html code

```
<!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 src="/fun1/index.js"></script>
</body>
</html>
```

JavaScript code

```
//using Hositing Function

hositingFunction();

function hositingFunction(){
    console.log('hi hositing function');
}
```

Output

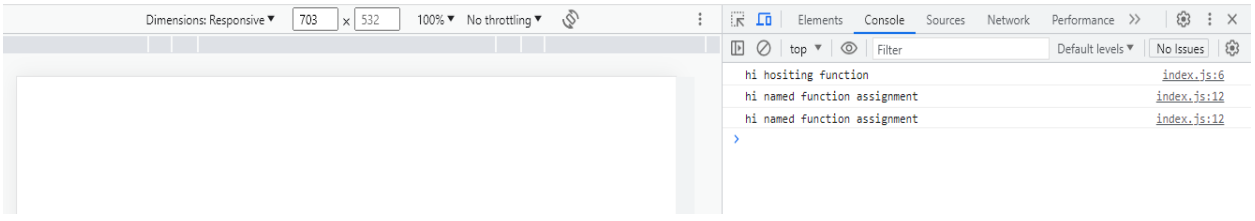


Named function assignment

```
//using Hositing Function

hositingFunction();

function hositingFunction(){
    console.log('hi hositing function');
}

//named function assignment

let stand=function walk(){
    console.log('hi named function assignment');
}

stand();

let jump=stand;

jump();
```

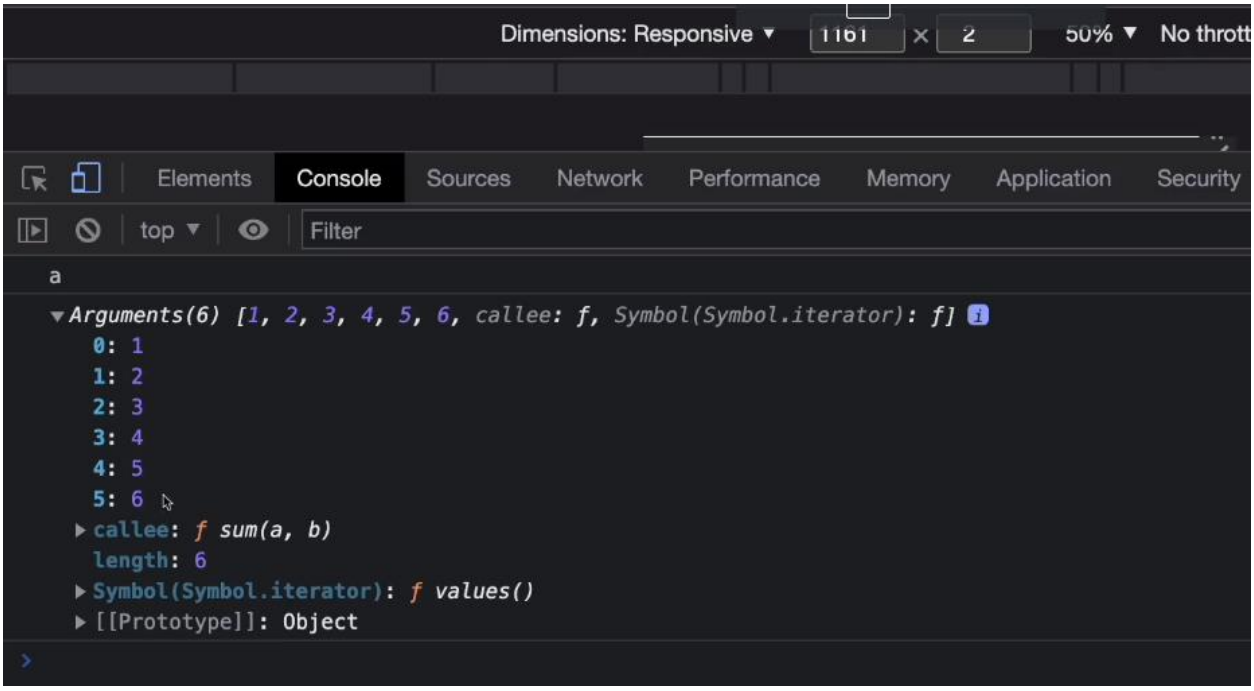
function declararion

function declaration mai function ko call function declaration ke uppar bhi kar sakte hai aur function declaration ke baad bhi



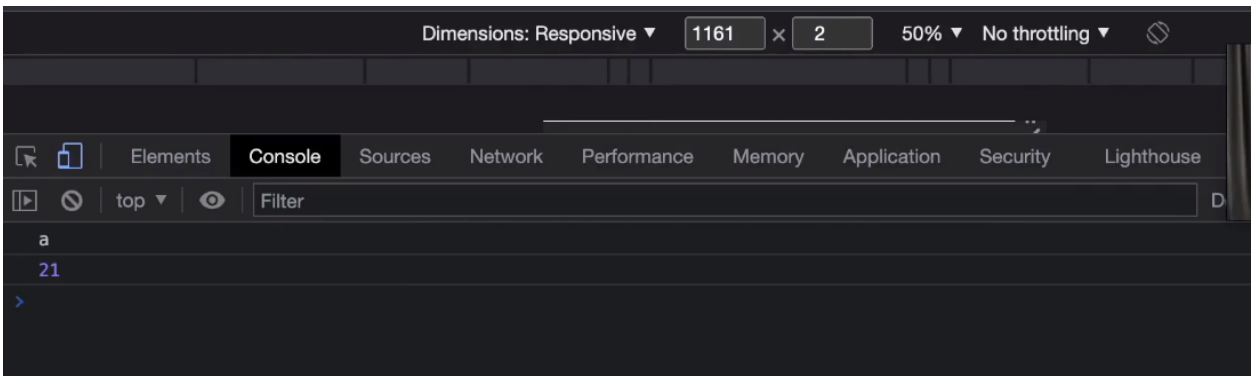
Special objects - arguments

```
41  function sum(a, b) {
42      console.log(arguments);
43      return a+b;
44  }
45
46  // console.log(sum(1,2));
47  //console.log(sum(1));
48  // console.log(sum());
49  //console.log(sum(1,2,3,4,5));
50
51  let ans = sum(1,2,3,4,5,6);
52
```



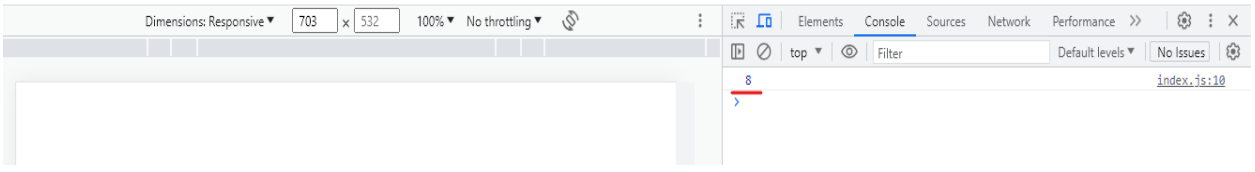
Sab value arguments mai hoti hai

```
41  function sum(a, b) {
42      let total = 0;
43      for(let value of arguments)
44          total = total + value;
45      return total;
46  }
47
48  // console.log(sum(1,2));
49  //console.log(sum(1));
50  // console.log(sum());
51  //console.log(sum(1,2,3,4,5));
52
53  let ans = sum(1,2,3,4,5,6);
54  console.log(ans);
55
```



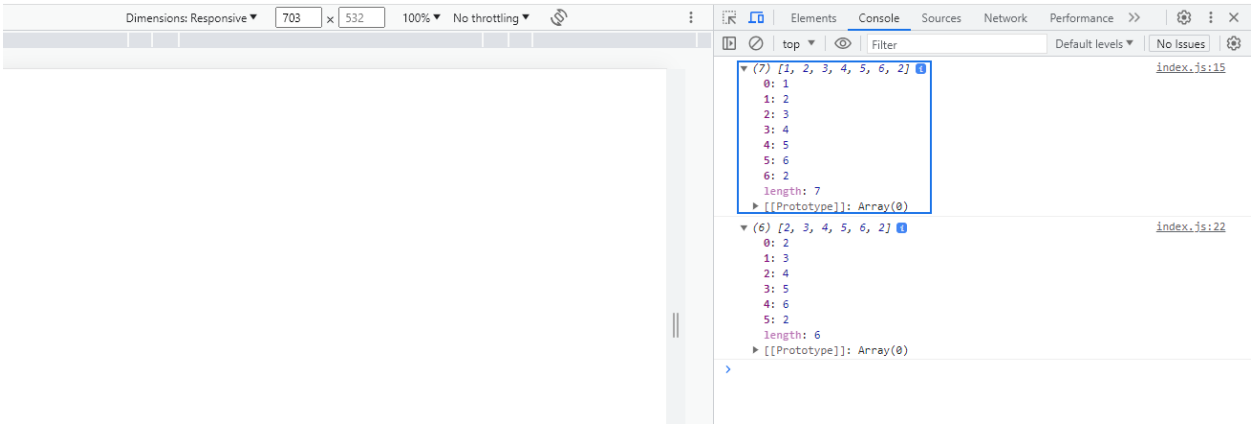
Arguments

```
function sum(a,b){
    let total=0;
    for(let value of arguments){
        total=total+value;
    }
    return total;
}

let total=sum(1,2,5);
console.log(total);
```



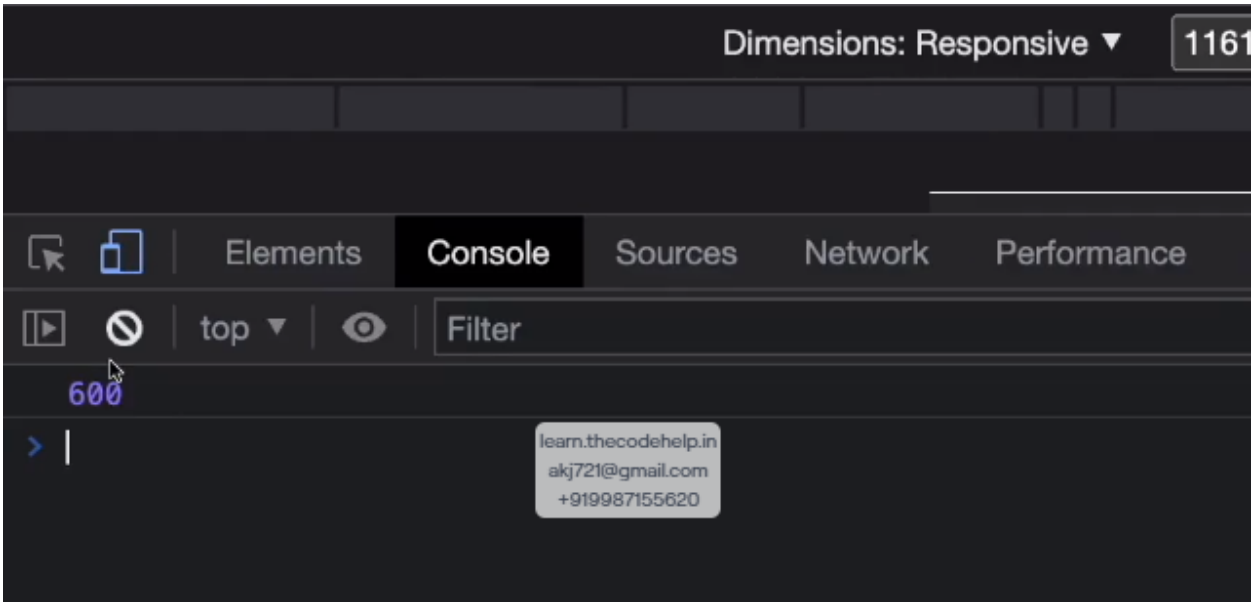
Rest parameter

```
//rest parameter

function sum(...args){
    console.log(args);
}

sum(1,2,3,4,5,6,2);


function rest1(a,...args){
    console.log(args);
}

rest1(1,2,3,4,5,6,2);
```



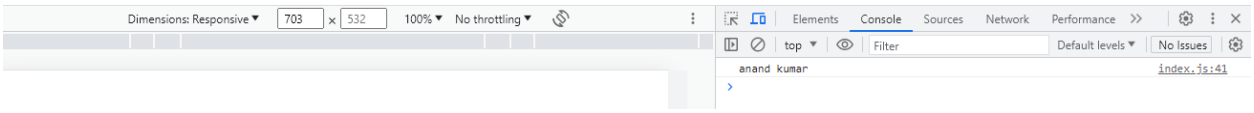
Default parameter

```
65  //Default Parameters
66  function interest(p,r=6,y=10) {
67      return p*r*y/100;
68  }
69
70
71  console.log(interest(1000,));
72
```

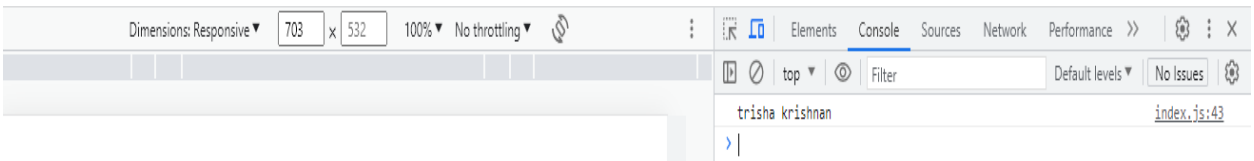


Getter and Setter

```
let person={
    firstName:'anand',
    lastName:'kumar',

    get fullName(){
        return `${person.firstName} ${person.lastName}`;
    },
    set fullName(val){
        let parts=val.split(' ');
        this.firstName=parts[0];
        this.lastName=parts[1];
    }
};

console.log(person.fullName);
```

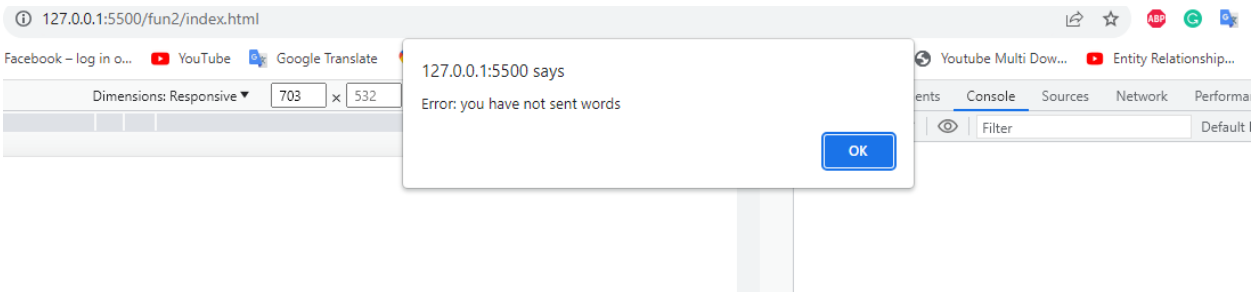



```
let person={
    firstName:'anand',
    lastName:'kumar',

    get fullName(){
        return `${person.firstName} ${person.lastName}`;
    },
    set fullName(val){
        let parts=val.split(' ');
        this.firstName=parts[0];
        this.lastName=parts[1];
    }
};

person.firstName='trisha';
person.lastName='krishnan';
console.log(person.fullName);
```



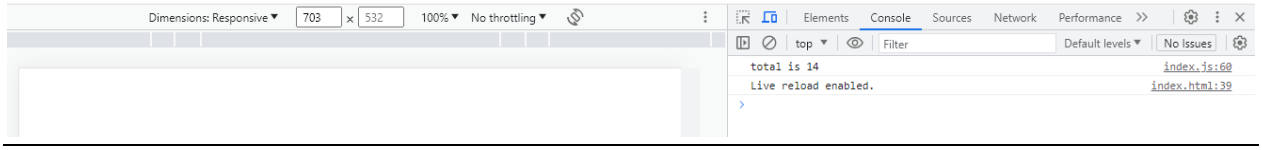
Try catch

```
let person={
    firstName:'anand',
    lastName:'kumar',

    get fullName(){
        return `${person.firstName} ${person.lastName}`;
    },
    set fullName(val){
        if( typeof val!==String){
            throw new Error("you have not sent words ");
        }
        let parts=val.split(' ');
        this.firstName=parts[0];
        this.lastName=parts[1];
    }
};

try {
    person.fullName=true;

} catch (e) {
    alert(e);
}

console.log(person.fullName);
```



Reduce

```
//reduce

let arr=[1,2,6,5];

let total=arr.reduce((val1,val2)=>val1+val2,0);
console.log('total is ' +total);
```




```
let arr = [1,2,3,4];
// let total = 0;

// for(let value of arr)
//     total = total + value;

// console.log(total);

let totalSum = arr.reduce((accumulator, currentValue) => accumulator + currentValue, 0 );
console.log("PRINTING TOTAL SUM:")
console.log(totalSum);
```

callback

accumulator

initialize