

## Rest parameter (...):

- Rest parameter is a parameter used to handle the input parameters passed to a function.
- The rest parameter allows us to represent indefinite number of arguments as an array.

Ex:

```
let printNumbers = (...input) => {  
  console.log(input);  
}  
printNumbers(10,20,30,60,50,78,99,999,55)
```

- Rest parameter has to be the last argument, as its job is to collect all the remaining arguments into an array.

```
let fun = (a,b,...c) => {  
  console.log(a);  
  console.log(b);  
  console.log(c);  
}  
fun("virat","dhoni","sachin","raina","ashwin")
```

Note : Rest parameters return an array.

## Spread Operator (...):

- Spread operator used to spread or expand the iterable object.

```
let add = (a, b, c,d) => {  
  console.log(a);  
  console.log(b);  
  console.log(c);  
  console.log(d);  
  console.log(a+b+c);  
}  
let array = [10,20,30]  
add(...array)
```

## Destructuring :

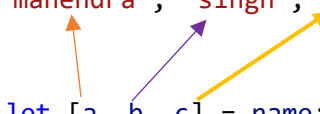
- ES6 feature
- It is JS expression that makes it possible to unpack values from arrays or properties from object into distinct variables.
- It makes the JS code more readable.

### 1) Array Destructuring :

```
let name = ["mahendra", "singh", "dhoni"]
let [a, b, c] = name;
console.log(a);
console.log(b);
console.log(c);
```

Here we are mapping the values

```
let name = ["mahendra", "singh", "dhoni"]
let [a, b, c] = name;
```



Always the first variable will hold the value present in the array index 0, and so on..

Ex 2: Access only first and last element

```
let fruits = ["apple", "pineapple", "custurdapple", "elephantapple"]
let [a, , , b] = fruits;
console.log(a);
console.log(b);
```

Ex 2: Exclude pineapple and custurdapple in given array.

```
let fruits = ["apple", "pineapple", "custurdapple", "elephantapple", "greenapple",
"yellowapple"];
let [a, , , ...x] = fruits;
console.log(a);
console.log(x);
```

## 2)Object destructuring:

In object we are destructuring the properties present in that object.

```
let emp = {  
  name: "Rahul",  
  age :25  
}  
let { name, age } = emp;  
console.log(name);  
console.log(age);
```

### Ex for nested object

```
let emp1 = {  
  name: "shreya",  
  age: 20,  
  hobby: {  
    fhobby: "drawing",  
    shobby:"singing"  
  }  
}  
let { name, age, hobby :{fhobby,shobby} } = emp1  
console.log(name);  
console.log(age);  
console.log(fhobby);  
console.log(shobby);
```