

## \* Function:-

```
function greeting () {  
    console.log ("Hello world");  
}
```

```
greeting();
```

```
function addNumber(num1, num2) {  
    const sum = num1 + num2;  
    console.log(sum);  
}
```

```
addNumber(3, 4);
```

```
addNumber(5, 1);
```

## \* // <sup>rest</sup> ~~rest~~ operator

```
function addNumber(... num) {  
    let sum += n;  
}
```

```
console.log(sum);  
}
```

```
addNumber(6, 7);
```

```
addNumber(6, 7, 8);
```

```
addNumber(6, 7, 8, 9);
```

```
addNumber(7, 8, 12, 12, 11, 12);
```

## \* Function expression :-

```
const addNumber = function (num1, num2) {  
  return num1 + num2;  
}
```

```
console.log(addNumber(3, 4));
```

## // another method

```
function addNumber(num1, num2) {  
  return num1 + num2;  
}
```

```
console.log(addNumber(3, 4));
```

→ Isko hm function ke pehle  
bhi likh sakte hai aur baad me  
bhi likh sakte hai.

## // Arrow function :-

```
const addNumber = () => {  
  console.log("Hello world");  
}
```

```
addNumber();
```



05  
const addNumber = (num1, num2) => {  
 return num1 + num2;  
}

console.log(addNumber(3, 4));

Note:-

If hmare paas single parameter hai tab () brackets ki koi jarurt nhi.

eg: const squareNumber = num => num \* num;

console.log(squareNumber(6));

06 IIFE Function.

```
(function greeting() {  
  console.log("Hello ji");  
})();
```

```
06 (() => {  
  console.log("hi");  
})();
```

## A callback function:-

```
function greet () {  
  console.log("Hello ji, Kaise ho")  
}
```

```
function meet (callback) {  
  console.log("I am going to meet someone.")  
  callback ();  
}
```

```
meet (greet);
```