es5:

```
function sum(){
    // console.log(Array.from(arguments).reduce(function(item1,item2){
    //     return item1+item2
    // }))
    let num = 0;
    Array.prototype.forEach.call(arguments,function(item){
        num += item*1
    })
    console.log(num)
    return num
}
sum(3,5,4,12,7)
```

es6:

```
function sum(...nums){ // 参数前三个点代表所有参数
    console.log(nums.reduce(function(item1,item2){
        return item1+item2
    }))
}
sum(3,5,4,12,7)
```

————————————————————————————————

```
function sum(base,...nums){ // 参数前三个点代表剩余所有参数，参数收敛到一起
```

rest(剩余)

```
    let num = 0;
    let num1 = nums.reduce(function(item1,item2){
        return item1+item2
    })
    num += base*2+num1;
    console.log(num)
}
sum(3,5,4,12,7)
```

———————————————————————————————————

```
function sum(x=1,y=2,z=3){
    console.log(x+y+z)
}
let data = [4,5,6];
// sum.apply(this,data)
sum(...data) // 参数打散到函数spread(分发)
```