

剩余参数

(... 名字) 获取的是一个数组

```
1  let sihai=(...all)=>{ //把 sihai中的所有参数放入 all里面
2      console.log(all);
3  }
4  sihai('jack',19)
```

匹配jack 剩下的给all

```
1  let sihai=(username ,...all)=>{ //匹配jack 剩下的给all
2      console.log(all);
3  }
4  sihai('jack',19 , 'dada')
```

在对象和数组中叫展开运算符通常用于合并对象和数组

合并两个对象到一个对象中

```
1  let obj1={
2      username:'jsck',
3      age:18,
4      id:1
5  }
6  let obj2={
7      commpy:'itcast',
8      salary:1000
9  }
10
11  let obgALL={...obj1,...obj2}
12
13  console.log(obgALL);
```

展开后的对象里面还有对象修改对象属性会互相影响（浅拷贝（改变了值但是地址没有变，所以相互影响了））~!!!! 重点 ✨

```
1  let obg3={
2      username:'jack',
3      hubby:[{
4          id:1,
```

```

5         name: 'tv'
6     }, {
7         id: 2,
8         name: 'tc'
9     }]
10 }
11 let obj4 = {
12     age: 1
13 }
14 let objall2 = { ...obj3, ...obj4 }
15 objall2.hubby[0].name = 'game'
16 console.log(objall2);
17 console.log(obj3);

```

对象的解构赋值

将对象当中的键分别解构给变量

username 和 sex 本身就是 obj 里面的属性

id 的也是 obj 里面的属性，如果全局还有一个 id 变量就必须重命名

对于嵌套的对象可以继续解构

剩余参数只能放在最后

```

1  let obj = {
2      username: 'jack',
3      age: 19,
4      id: 12,
5      sex: 'male',
6      score: 100
7  }
8  //let id = 999
9  let { username, id, sex } = obj
10 console.log(username, id, sex);
11 //id重命名
12 // let { username, id: user, sex, ...rest } = obj
13 // console.log(username, user, sex);
14 // console.log(username, user, sex, ...rest ); //...rest返回是一个对象

```

数组的解构赋值

```

let arr = [
  {username: 'jack'},
  {username: 'rose'},
  {username: 'zhangsan'},
  {username: 'lisi'},
  {username: 'wangwu'},
]
// rest2剩余的数组项和上面对象的不一样，这里是一个数组
let [obj1, obj2, ...rest2] = arr
console.log(obj1, obj2, rest2);

```

```

// 在函数的参数位置使用对象的解构
function printObj({ username }) {
  console.log(username);
}
printObj(obj)

```

控制合并运算符

??

当左侧面操作为空null或者为undefind时返回右边的操作否者返回左边

```

1 let app=null
2       let res=app ?? 'hello'
3       console.log(res);

```