

借助函数参数解构，即可完成复杂参数不按顺序传参并且能够正常接收

```
1 function getFullName(a, b, { firstName, lastName }) {  
2   console.log(a, b)  
3   console.log(firstName, lastName)  
4 }  
5  
6 let obj = {  
7   firstName: '张三',  
8   haha: 'dasda',  
9   dd: 'fff',  
10  lastName: '李四',  
11  ff: 'ddd'  
12 }  
13  
14 getFullName('1', '2', obj)
```

1 2

张三 李四

便于扩展，不影响后面追加其他参数：

```
1 function getFullName(a, b, { firstName, lastName }, c) {  
2   console.log(a, b, c)  
3   console.log(firstName, lastName)  
4 }  
5  
6 let obj = {  
7   firstName: '张三',  
8   haha: 'dasda',  
9   dd: 'fff',  
10  lastName: '李四',  
11  ff: 'ddd'  
12 }  
13  
14 getFullName('1', '2', obj, {a: 1})
```

```
1 2 ▶ {a: 1}
```

```
张三 李四
```

可以对解构参数设置默认值：

```
1 function getFullName(a, b, { firstName = 'first', lastName = 'last' }, c)
2 {
3   console.log(a, b, c)
4   console.log(firstName, lastName)
5 }
6
7 let obj = {
8   firstName: undefined,
9   haha: 'dasda',
10  dd: 'fff',
11  lastName: undefined,
12  ff: 'ddd'
13 }
14 getFullName('1', '2', obj, {a: 1})
```

```
1 2 ▶ {a: 1}
```

```
first last
```

如果函数返回多个值，优先使用对象的解构赋值，而不是数组的解构赋值。这样便于以后添加返回值，以及更改返回值的顺序。

```
1 // bad
2 function processInput(input) {
3   return [left, right, top, bottom];
4 }
5
6 // good
7 function processInput(input) {
8   return { left, right, top, bottom };
9 }
```

10

11 `const { left, right } = processInput(input);`