工作中应该使用的代码技巧

1. 函数传递默认参数值

```javascript
volume = (l, w = 3, h = 4 ) => (l * w * h);

volume(2) //output: 24
```

如果不传入参数的话按照值为默认参数，传入参数为传入的值

l=2,w=3,h=4

2. 解构赋值时分配变量名：

```javascript
const { store, form, loading, errors, entity:contact } = this.props;
//最后一个变量名为contact
```

相当于为contact赋值

3. 扩展运算符简写

```javascript
// joining arrays
const odd = [1, 3, 5 ];
const nums = [2 ,4 , 6, ...odd];
console.log(nums); // [ 2, 4, 6, 1, 3, 5 ]

// cloning arrays
const arr = [1, 2, 3, 4];
const arr2 = [...arr];
```

相对于concat()来说更加灵活，可以在数组的任意处插入
const nums = [2, ..odd, 4, 6];
// [2, 1, 3, 5, 4, 6]

这里也可以使用Object.assign()
Object.assign()和扩展运算符都是浅拷贝，
如果对应的数组或对象的属性对应的属性值
还是对象的话最好使用深拷贝

使用扩展运算符进行解构

```javascript
const { a, b, ...z } = { a: 1, b: 2, c: 3, d: 4 };
console.log(a) // 1
console.log(b) // 2
console.log(z) // { c: 3, d: 4 }
```

剩余属性组成的对象

4. 使用 `Array.find()` 从数组中查找某个值

```
var array1 = [5, 12, 8, 130, 44];

var found = array1.find(function(element) {
  return element > 10;
});

console.log(found);
// expected output: 12
```

Array.filter():返回true项组成的数组，这里输出：[12, 130, 44]
Array.find():返回数组中满足条件的第一个元素的值，这里输出：12