

参数

默认参数 default

普通写法:

```
function foo(x,y) {  
    x = x || 11;  
    y = y || 31;  
    console.log( x + y );  
}
```

问题: 当传入的参数为0或者false时, 会直接取到后面的值, 而不是传入的这个参数值。必须要加入一步判断是否传入参数, 很麻烦。

es6:

```
function foo(x = 11, y = 31) {  
    console.log( x + y );  
}
```

```
foo();    // 42  
foo( 5, 6 );    // 11  
foo( 0, 42 );    // 42  
foo( 5 );    // 36  
foo( 5, undefined );    // 36  
foo( 5, null );    // 5  
foo( undefined, 6 );    // 17  
foo( null, 6 );    // 6
```

只有传入undefined时才会触发默认值, 不传参数, 也相当于所有参数都传入undefined。传入null和false相当于传入0。

rest

可变参数, 使函数可以接受任意数量的参数。用三个点 ... 表示。

```
function animals(...types){  
    console.log(types)  
}  
animals('cat', 'dog', 'fish')    //["cat", "dog", "fish"]
```

如果不用ES6的话, 我们则得使用ES5的 arguments 。