函数参数的默认值

**function** **log**(x, y = 'World') {

console.log(x, y);

}

log('Hello') *// Hello World*

log('Hello', 'China') *// Hello China*

log('Hello', '') *// Hello*

* 函数内声明的私有变量不能与形参相同
* 传值的优先级高于默认值的优先级
* 使用参数默认值时，函数不能有同名参数
* 参数默认值是惰性求值
* 参数默认值可以与解构赋值的默认值，结合起来使用
* 如果传入undefined，将触发该参数等于默认值，null则没有这个效果。
* 指定了默认值后，length属性将失真。
* 默认值为...reset, length属性将失真

**function** **sum**(...n){

console.log(n);

*//[10,11,12,13]*

};

sum(10,11,12,13)

注意，rest 参数之后不能再有其他参数（即只能是最后一个参数），否则会报错。 函数的length属性，不包括 rest 参数。

ES2016 做了规定只要函数参数使用了默认值、解构赋值、或者扩展运算符，那么函数内部就不能显式设定为严格模式，否则会报错

name属性

函数的name属性，返回该函数的函数名

**function** **foo**() {}

foo.name *// "foo"*

箭头函数

* ES6 允许使用“箭头”（=>）定义函数。

**var** f = v => v;

*//====================*

**var** f = **function**(v) {

**return** v;

};

* 多个参数用逗号隔开

**var** sum = (num1, num2) => num1 + num2;

*//====================*

**var** sum = **function**(num1, num2) {

**return** num1 + num2;

};

* 如果箭头函数直接返回一个对象，必须在对象外面加上括号，否则会报错。

不报错

**let** getTempItem = id => ({ id: id, name: "Temp" });

* 函数体内的this对象，就是定义时所在的对象，而不是使用时所在的对象。
* 箭头函数无自己的this，导致内部的this就是指向的外层代码块的this
* 不可以使用arguments对象
* 不可以当作构造函数

this对象的指向是可变的,但是在箭头函数中,它是固定的

*  箭头函数可以嵌套使用