1. 上下文调用模式

普通调用 : this --> window

对象调用 : this --> obj

构造函数调用 : this --> 构造函数的实例

定时器中调用 : this --> window

事件调用 : this --> 事件源

一般情况下this是谁调用指向谁,通过上下文调用模式,可以让this想指向谁就指向谁.

1. call 语法:函数A.call(对象A,实参)

意思是 函数A里的this指向对象A,然后实参根据函数A的形参顺序一个个传.

Eg:function fn(x, y){

console.log(this); // ['a','b','c']

console.log(x + y); //8

}

fn.call(['a','b','c'], 3, 5);

1. apply 语法:函数A.apply(对象,数组)

意思是 函数A里的this指向对象A,然后数组里是函数A的实参,下标对应函数A的形参的顺序.

Eg:function fn(x, y){

console.log(this); // {a : 1}

console.log(x + y); //8

}

fn.apply({a : 1}, [3, 5]);

1. bind 语法:函数A.bind(对象A,实参)

同call

三者的差异

1. call和apply会调用函数A,bind不会调用函数,会克隆出一个新的函数.
2. apply需要把函数需要的实参放到数组中