//通过分析原生的new方法可以看出，在new一个函数的时候，

// 会返回一个func同时在这个func里面会返回一个对象Object，

// 这个对象包含父类func的属性以及隐藏的\_\_proto\_\_

function New(f) {

//返回一个func

return function () {

var o = {"\_\_proto\_\_": f.prototype};

f.apply(o, arguments);//继承父类的属性

return o; //返回一个Object

}

}

function Person(name,age){

this.name = name;

this.age = age;

}

通过自定义New方法创建一个实例对象p2:

var p2 = New(Person)("Jack",25);

p2.name;//Jack

p2.age;//25

此时p2 instanceof Person 返回的是true；

Person.prototype.gender ="male";

p1.gender//male

p2.gender//male

ES6的reflect.construct(target,args)也可以替代new的功能

function Greeting(name) {

this.name = name;

}

const instance = Reflect.construct(Greeting, ['张三']);