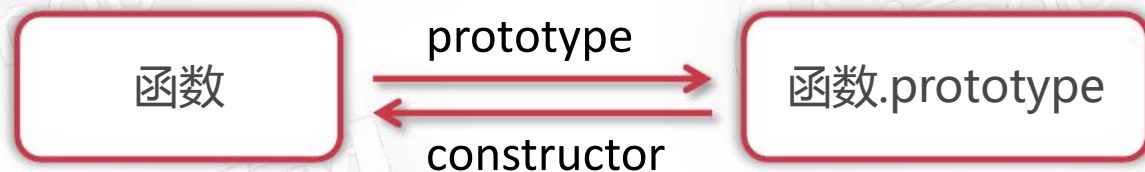


# prototype和原型链查找

## 什么是prototype

- ◆ 任何函数都有prototype属性，prototype是英语“原型”的意思
- ◆ prototype属性值是个对象，它默认拥有constructor属性指回函数

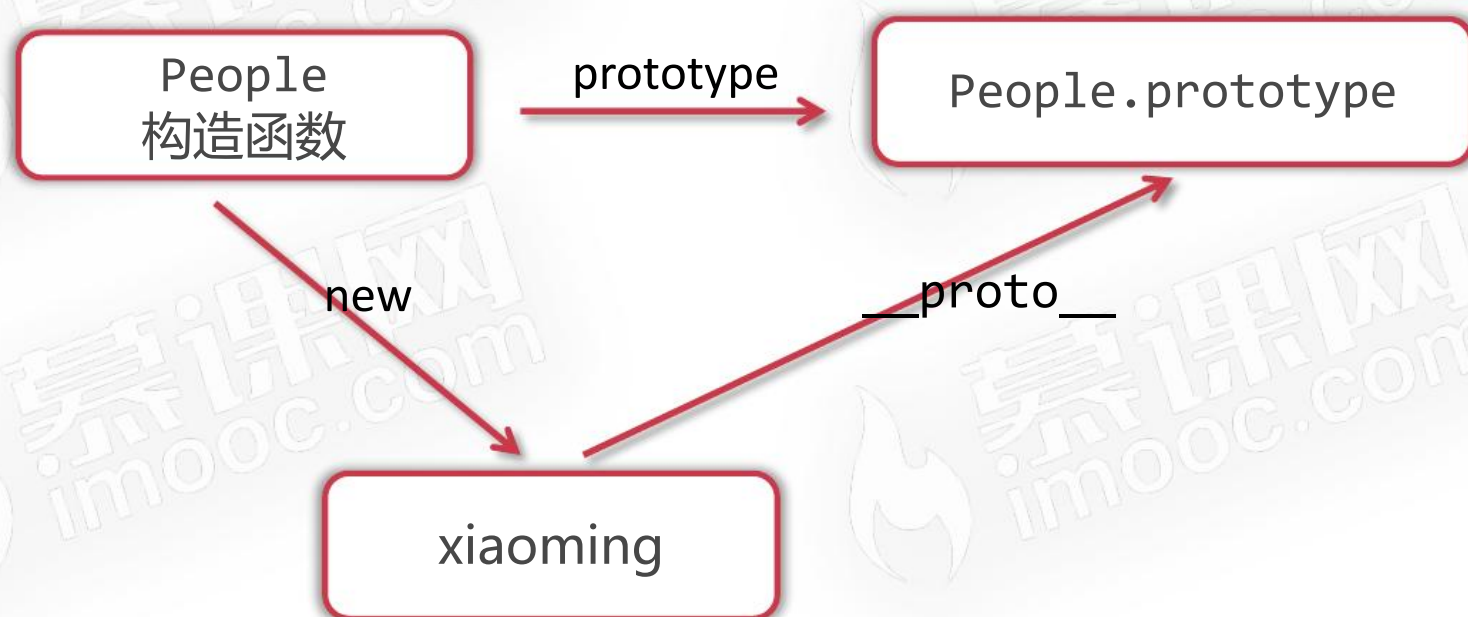
# 什么是prototype



# 什么是prototype

- ◆ 普通函数来说的prototype属性没有任何用处，而构造函数的prototype属性非常有用
- ◆ 构造函数的prototype属性是它的实例的原型

# 构造函数的prototype是实例的原型



## 原型链查找

- ◆ JavaScript规定：实例可以打点访问它的原型的属性和方法，这被称为“原型链查找”

```
function People(name, age, sex) {  
  this.name = name;  
  this.age = age;  
  this.sex = sex;  
}
```

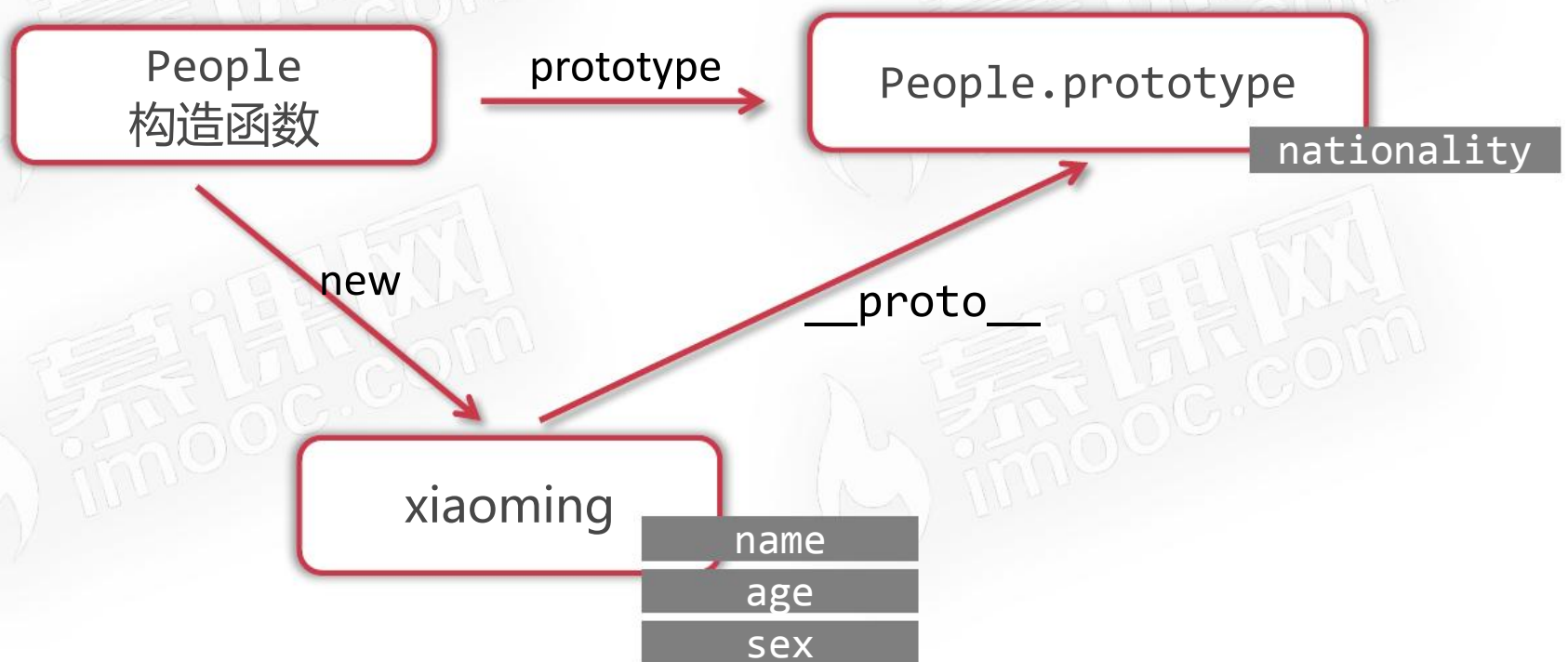
```
People.prototype.nationality = '中国';
```

在构造函数的prototype上  
添加nationality属性

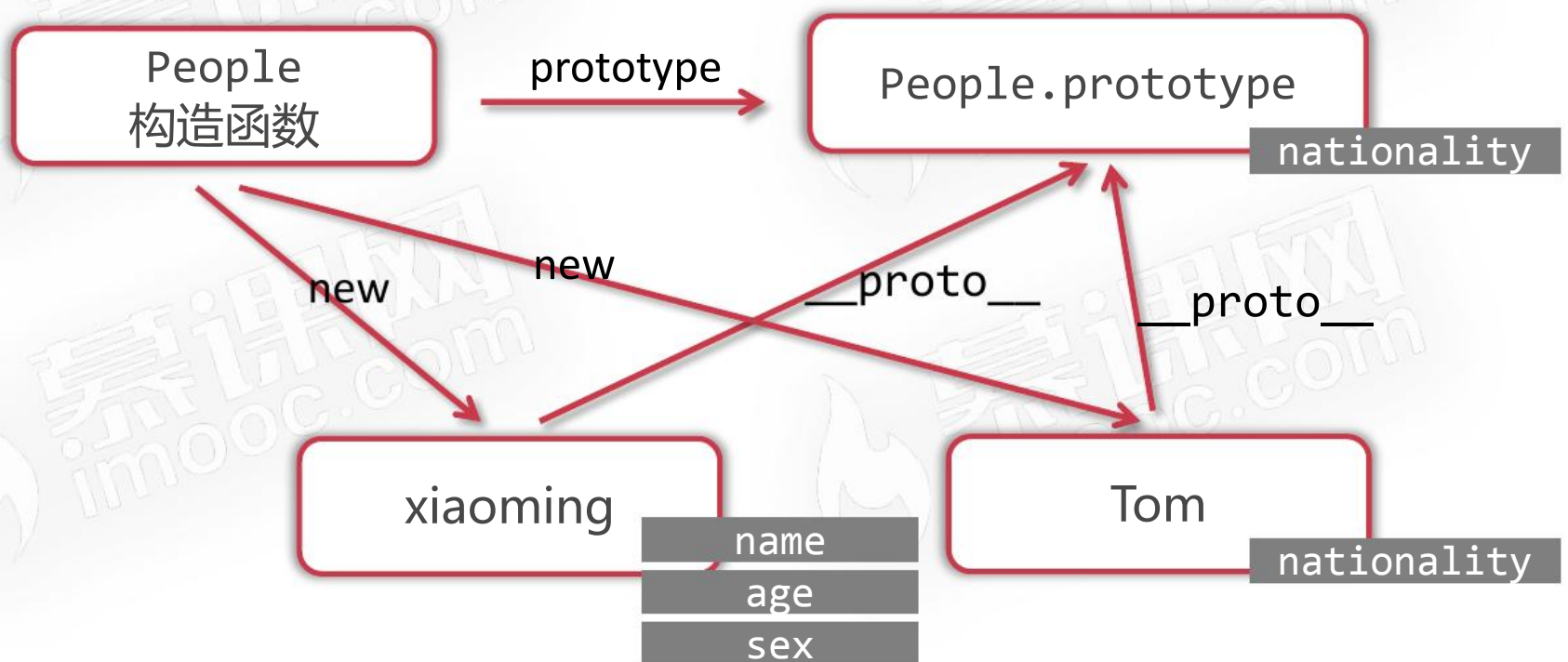
```
var xiaoming = new People('小明', 12, '男');  
console.log(xiaoming.nationality);
```

实例可以打点访问  
原型的属性和方法

## 原型链查找



## 原型链查找





# hasOwnProperty

- ◆ hasOwnProperty方法可以检查对象是否真正“自己拥有”某属性或者方法

```
xiaoming.hasOwnProperty('name');           // true
xiaoming.hasOwnProperty('age');             // true
xiaoming.hasOwnProperty('sex');             // true
xiaoming.hasOwnProperty('nationality');     // false
```

## in

- ◆ in运算符只能检查某个属性或方法是否可以被对象访问，不能检查是否是自己的属性或方法

```
'name' in xiaoming           // true
'age' in xiaoming            // true
'sex' in xiaoming            // true
'nationality' in xiaoming    // true
```



谢谢！

