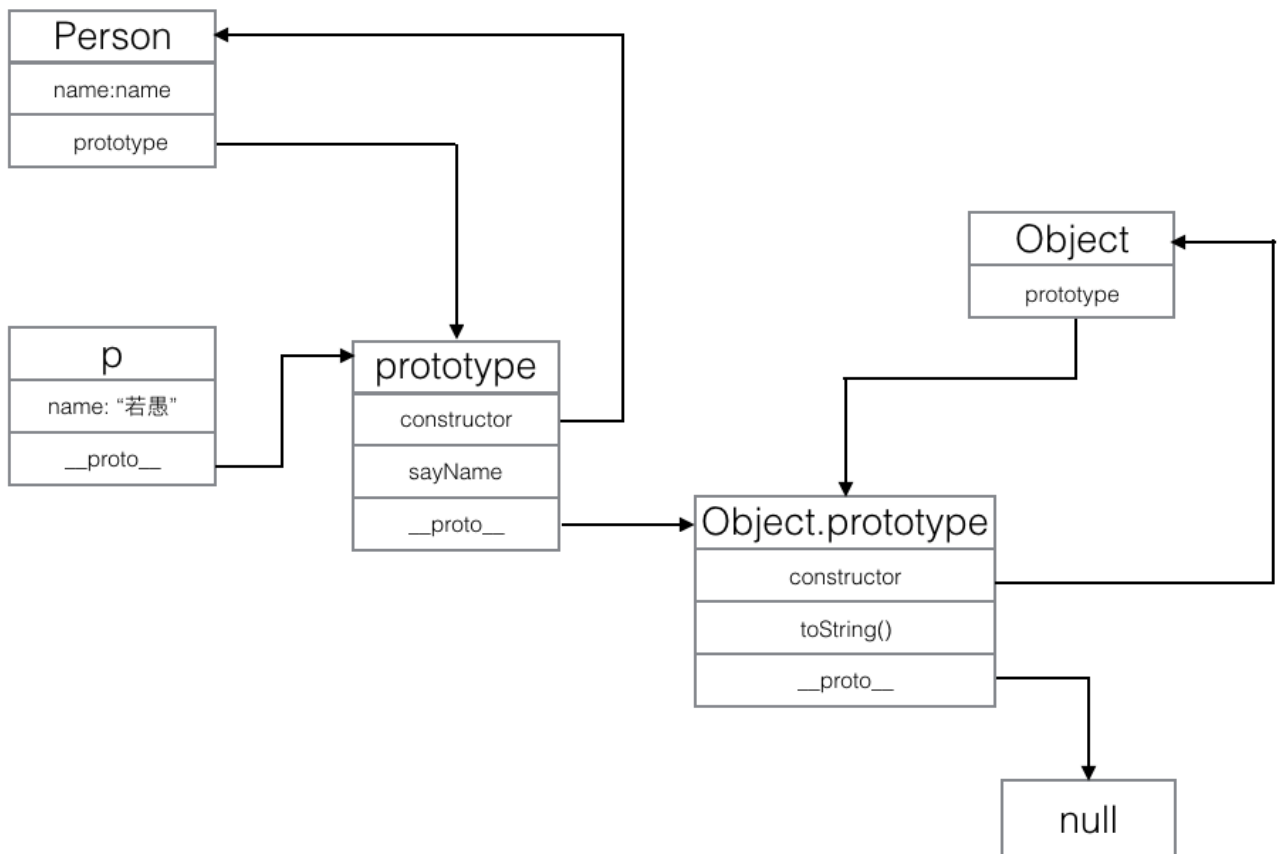


# 继承



## 构造函数的继承

### 1.Object.create法

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

Person.prototype.printName = function(){
    console.log(this.name);
};

function Male(name, sex, age){
    Person.call(this, name, sex);
    this.age = age;
}

function inherit(superType, subType){
    var _prototype = Object.create(superType.prototype);
    _prototype.constructor = subType;
    subType.prototype = _prototype;
}
```

```

}

//Male.prototype = new Person(); 原生的，兼容低版本IE

inherit(Person, Male);

// 在继承函数之后写自己的方法，否则会被覆盖
Male.prototype.printAge = function(){
    console.log(this.age);
};

var m = new Male('Byron', 'm', 26);
m.printName();

```

## 2.利用空对象作为中介

由于“直接继承prototype”存在上述的缺点，所以就有第四种方法，利用一个空对象作为中介。

```

var F = function(){};
F.prototype = Animal.prototype;
Cat.prototype = new F();
Cat.prototype.constructor = Cat;

```

F是空对象，所以几乎不占内存。这时，修改Cat的prototype对象，就不会影响到Animal的prototype对象。

```

alert(Animal.prototype.constructor); // Animal

```

我们将上面的方法，封装成一个函数，便于使用。

```

function extend(Child, Parent) {
    var F = function(){};
    F.prototype = Parent.prototype;
    Child.prototype = new F();
    Child.prototype.constructor = Child;
    Child.uber = Parent.prototype;
}

```

使用的时候，方法如下

```

extend(Cat,Animal);
var cat1 = new Cat("大毛","黄色");
alert(cat1.species); // 动物

```

这个extend函数，就是YUI库如何实现继承的方法。

另外，说明一点，函数体最后一行

```
Child.uber = Parent.prototype;
```

意思是子对象设一个uber属性，这个属性直接指向父对象的prototype属性。（uber是一个德语词，意思是“向上”、“上一层”。）这等于在子对象上打开一条通道，可以直接调用父对象的方法。这一行放在这里，只是为了实现继承的完备性，纯属备用性质。

## 非构造函数的继承

### 3.浅拷贝

```
function extendCopy(p) {  
    var c = {};  
    for (var i in p) {  
        c[i] = p[i];  
    }  
    c.uber = p;  
    return c;  
}
```

### 4.深拷贝

```
function deepCopy(p, c) {  
    var c = c || {};  
    for (var i in p) {  
        if (typeof p[i] === 'object') {  
            c[i] = (p[i].constructor === Array) ? [] : {};  
            deepCopy(p[i], c[i]);  
        } else {  
            c[i] = p[i];  
        }  
    }  
    return c;  
}
```