20170805

复习原型链

```
//易车面试题
//用友面试题
```

函数三种角色

```
//BAT面试题
```

call/apply/bind

js中this的几种情况

> 和函数在哪儿定义以及在哪儿执行都没有任何的关系
> 你以为你以为的就是你以为的

```
//->1、自执行函数中的this是window
~function(){
    //this:window
}();

//->2、给元素的某个事件绑定方法，事件触发，方法执行，方法中的this是当前元素本身
function fn(){
    document.body.onclick=fn;//->点击body，触发fn执行，fn方法中的this:d
ocument.body
}

//->3、构造函数执行，函数体中出现的this都是当前类的一个实例
function Fn(){
    //->this:f
    this.x=100;
    this.y=200;
}
Fn.prototype.sum=function(){
    return this.x+this.y;
}
var f=new Fn;

//->4、方法执行，看方法前是否有“.”，有的话，“.”前面是谁this就是谁，没有“.”，t
his就是window
f.sum();//->this:f
Fn.prototype.sum();//->this:Fn.prototype
f.__proto__.sum();//->this:f.__proto__

//->5、可以使用call/apply/bind强制改变某一个方法中的this（这个规律的优先级最
高）
var obj={};
function fn(num){}
document.body.onclick=fn.bind(obj,100);//->点击的时候执行fn，fn中的thi
s：obj
```

但是以上都是在非严格的js模式下，js还有严格模式，严格模式下和非严格模式有点区别

```
"use strict";//->开启js的严格模式（严格模式：js代码更加严谨）
//->1、自执行函数中的this是undefined

//->2、函数执行，如果没有“.”，this是undefined而不是window

//->3、使用call/apply/bind的时候，如果第一个参数不传递，this不是window而是undefined，如果传递的是null/undefined，this指向的就是null/undefined，而不是之前所谓的window了

==>特点：严格模式下，如果没有确定执行的主体，this都是undefined，而不是非严格模式下的window了
```