

4 - 23

包装对象

- 当简单类型去使用某个属性或者方法的时候，内部会偷偷的转成对象（new 内置类）把属性或者方法提供给使用者，然后再悄悄地销毁；——这个过程就叫包装对象；
- 红领巾|| 活雷锋
- **for in** 不但会枚举本对象，还会枚举它的原型；
- 如果不要原型下的属性或者方法，只想枚举对象本身的属性或者方法可以用 `hasOwnProperty` 判断
- **obj.hasOwnProperty(属性名)**：判断该属性名是不是属于这个obj对象，是返回true。不是就false；

```
for(attr in obj){  
    if(obj.hasOwnProperty){  
        console.log(obj[attr])  
    }  
}
```

call和apply改变this指向

call： 有无数个参数

第一个参数，改变this指向（写啥this是啥）

null和undefined改变不了this指向

第二个参数之后：就是实参

apply 有两个参数

第一个参数： 改变this指向（写啥是啥）

第二个参数： 数组 [] (数组中放参数)；

bind：

有无数个参数

第一个参数:

改变this指向 (写啥是啥)

null和undefined为window

第二个参数之后

就是实参

使用bind不能立马执行函数, 会返回一个新函数, 这个函数的this是改变了的, 得执行这个新函数才能输出代码;

类数组转换数组的方法:

类数组转数组的方法:

```
Array.prototype.slice.call(arr);  
Array.from(arr);
```

取数组大小值:

```
console.log(Math.max.apply(null, ary));  
console.log(Math.max(...ary))
```

JSON -> [] | {}

JSON.parse()

能够把json转成对象或者数组

json必须是一个标准格式的json, 不然转不出来
json中不能放函数、不能为undefined

JSON.stringify()

把对象或者数组转成json 对象中不能放函数、不能为undefined

继承

定义子类继承了父类的一些特征, 然后自己还有一套自己的特征

作用就是为了代码能够更好复用, 组合起来生成一个新的类别

属性继承:

- 类式继承(构造函数继承)
把父类看做一个函数，调用这个父类并且通过call去改变this指向，把指向改为子类。
- 方法继承:
扩展式继承
`{...父类原型}`
 - 拷贝继承
深层克隆

```
function deepClone(obj){
    //先声明一个数组，去存克隆出来的内容
    //判断obj是否为数组，是数组就o就为[],否则为{}
    let o = obj.push?[]: {};
    //循环传进来的对象
    for(let attr in obj){
        // for(let i=0;i<arr.length;i++){
        //判断对象中的某个值是否为引用类型
        //如果是，就继续调用deepClone把引用值传到函数中
        if(typeof obj[attr] === 'object'){
            o[attr] = deepClone(obj[attr])
        }else{
            //如果是简单类型就直接赋值
            o[attr] = obj[attr];
        }
    }
    return o;
}
```

```
for in
    child.prototype = deepClone(parent.prototype)
```