

# this 指向

2018年7月5日 星期三

08:27

## 1、昨日内容回顾

### 1、递归函数

递归函数：自己调用自己的函数。

递归函数的第一要素：临界条件，（函数循环退出的条件）。如果

### 2、打印对象

### 3、复制对象

为了让新对象和原来对象没有关系。

伪代码

```
function f(obj){
```

先判断这个参数是不是对象。如果不是对象，就直接返回

再判断参数是数组还是对象啊？为啥？如果参数是数组，

```
var newObj = Obj instanceof Array ? [] : {}
```

```
for(var k in obj){
```

判断属性值 是不是对象？为什么？

```
if(obj[k] 是一个引用数据类型){
```

把这个引用数据再次遍历

```
newObj[k] = f(obj[k]);
```

```
}
```

```
newObj[k] = obj[k];
```

```
}
```

没有临界条件，就会死循环。

1。

，那么返回的是数组还是对象？

}

## 2、模拟块级作用域

作用域

全局作用域

局部作用域 ( 函数作用域 )

## ★ 3、this 指向

全局模式下 :

非严格模式 : this 指向 window

严格模式 : this 指向 undefined

★ 在函数内部 :

- 1、事件处理函数中 , this 指向的是 触发事件的那个元素。
- 2、对象调用方法时 , this 指向的是当前对象。
- 3、函数直接执行时 , this 指向的是 window
- 4、回调函数 ( setTimeout、forEach ) , this 指向 window
- 5、使用 call 和 apply 函数 , 可以修改 this 指向。你传什么 , this 就指向什么。

## ★ 4、call 和 apply 函数

Call和 apply 的作用是改变还是内部的 this 指向。传谁就指谁 , 一般情况

call和 apply 的区别

call 向函数内传递参数 , 只能一个一个传递 ,

apply 向函数内传递参数 放在一个数组里面 一起传递 数组今天

就指向谁。

兄下都是传对象。

函数内部解压 参数由右几个元素 arguments

apply 向函数传递参数，放在 1 数组里， 起传递。数组元素里就有几个元素。而不是把整个数组当成一个元素。

★ 5、javascript 高阶函数。

forEach	遍历数组	参数1 : 数三，遍 只是简单
map	对数组中每一项元素进行操作，返回一个新的数组，数组中放着修改后的每一项元素。	参数同上 返回值是 想对数组 map。m
filter	过滤数组，返回一个新数组，数组中放着满足过滤条件的元素	参数同上 返回值是 过滤数组

课堂练习

有五个 li ， 文字是 1 , 2 , 3 , 4 , 5 , 点击每个 li , 弹出 li 的文字。  
使用 for 循环给 li 添加事件。模拟块级作用域。

练习高阶函数  
对对象进行操作

四、双指针解法。双指针有以下几种，arguments

数组中的每一个元素，参数2：数组的下标，参  
遍历的数组  
的遍历

。  
新数组。和原数组无关  
中每一项元素都进行操作的时候，就用  
map 遍历时除了报错，并不能中断遍历

。  
新数组。和原数组无关  
时使用。



