[关于angularJS的$watch的 一些小用法](http://blog.csdn.net/u010451286/article/details/50635839)

$watch方法，它可以帮助我们在每个scope中监视其中的变量。

## $watch 单一的变量

对于普通的变量时，如数字，字符串等，直接如下写是可以监视到变量的变化，并执行相应的函数的。

$scope.count=1;

$scope.$watch('count',function(){

...

});

## $watch 多个变量

对于多个变量的监视变化，执行同一函数的话，可以将这几个变量转为字符串，以‘+’号隔开来进行监视

//当count或page变化时，都会执行这个匿名函数

$scope.count=1;

$scope.page=1;

$scope.$watch('count + page',function(){

...

});

## $watch对象或数组

发现用上面两种方法去监视数组时，会发现即使数组的内容改变了，也没有触发到这个匿名函数。之后发现watch函数其实是有三个变量的，第一个参数是需要监视的对象，第二个参数是在监视对象发生变化时需要调用的函数，实际上watch还有第三个参数，它在默认情况下是false。   
当第三个参数是false时，其实watch函数监视的是数组的地址，而数组的内容的变化不会影响数组地址的变化，所以watch函数失灵了。   
解决办法，就是在后面添加第三个参数为true就好（当然，也可以将这监听返回结果为JSON字符串形式的该对象或数组的的匿名函数。）

$scope.items=[

{a:1},

{a:2}

{a:3}];

$scope.$watch('items',function(){...},true);

或者将监听返回结果为JSON字符串形式的该对象或数组的的匿名函数

$scope.items=[

{a:1},

{a:2}

{a:3}];

$scope.$watch(function(){

return JSON.stringify($scope.items);

},function(){...});

## $watch 函数的返回结果

在写代码的时候，有时会遇到要监视一个函数返回的结果是否变化的情况，所以查了一下$watch 监视函数的情况。

方法1：监视对象为“函数名()”的字符串，记得加“()”!

//未完成的任务个数

$scope.unDoneCount = function() {

var count = 0;

angular.forEach($scope.todoList, function(todo) {

count += todo.done ? 0 : 1;

});

return count;

};

//单选影响全选部分

$scope.$watch('unDoneCount()', function(nv) {

$scope.isDoneAll = nv ? false : true;

});

方法2：在监视对象中设置为匿名函数，返回要监视的函数的返回值（绕晕了…）

$scope.$watch(function(){

return $scope.unDoneCount();//不要忘了(),要执行的啊~

}, function(nv) {

$scope.isDoneAll = nv ? false : true;

});

## 取消$watch

watch的性能消耗好像蛮大的，所以对于已经不需要监视的watch，记得定时取消掉。   
至于怎么取消了…查了好久才找到的   
其实每个watch函数返回的结果就是这个watch的deregisterWatch()函数

//在chrome的控制台上，断点得到的$watch的返回值

function deregisterWatch() {

arrayRemove(array, watcher);

lastDirtyWatch = null;

}

所以啊，要取消watch的话，一开始将$watch的返回值保存就好啦，要取消watch的时候，在调用。

var count=1;

var unbingWatch=$scope.$watch('todoList',function(){

console.log('todoList change');

count++;

//相当于在todoList变化了4次之后，就调用unbingWatch()取消这个watch

//在第5次todoList改变的时候,就不会输出todoList change了。

if(count>4){

unbingWatch();

}

});