[表驱动法概念到实战(一) 原理及基本运用](http://www.cnblogs.com/shinn/archive/2008/04/16/1157141.html)

一:什么是表驱动法  
     1） 所谓表驱动法(Table-Driven Approach),简单讲是指用查表的方法获取值。

我们平时查字典就是典型的表驱动法。在数值不多的时候我们可以用逻辑语句(if 或case)的方法来获取值,但随着数值的增多逻辑语句就会越来越长,此时表驱动法的优势就显现出来了。

表驱动方法是一种使你可以在表中查找信息，而不必用很多的逻辑语句（if或Case）来把它们找出来的方法。

1. “表”是几乎所有数据结构课本都要讨论的非常有用的数据结构。表驱动方法出于特定的目的来使用表。  
     
   二:为什么要用表驱动。举例说明其优越性。获取月份及天数。

If(month==1){

days=31;

}else if(month==2){

days=28;

}else if(month==3){

days=31;

}else if(month==4){

days=30;

}else if(month==5){

days=31;

}else if(month==6){

days=30;

}else if(month==7){

days=31;

}else if(month==8){

days=31;

}else if(month==9){

days=30;

}else if(month==10){

days=31;

}else if(month==11){

days=30;

}else if(month==12){

days=31;

}

var days=[31,28,31,30,31,30,31,31,30,31,30,31];

var getDay=function(month){

var mon=month-1;

return days[mon];

}

getDay(1);

1. 提高了程序的可读性。一个消息如何处理，只要看一下驱动表就知道，非常明显。  
   2、减少了重复代码。  
   3、可扩展性。

4、程序有一个明显的主干。  
5、降低了复杂度。通过把程序逻辑的复杂度转移到人类更容易处理的数据中来，从而达到控制复杂度的目标。  
三:查表的方式  
   在使用表驱动法的时候必须要解决的一个问题就是如何查表.  
我们可以用非常直接的方式查一个表,用数组下标就好了,如月份天数的查询；

但你发现有谁查字典是直接依靠页码来查的吗?这是肯定行不通的。  
常用的查表方式有

* 直接查询
* 索引查询
* 分段查询

3.1 直接查询  
直接查询，是指无需绕圈子，用下标的方式就能顺利的获取到数据；  
实现同样的功能,我们可以将这些数据存储到一个表里  
  
  
  
只要一条语句就可以代替长长的if-else语句  
var dayNames=[“星期日”，“星期一”，“星期二”，“星期三”，“星期四”，“星期五”，“星期六”];

function getDayName(day){

return dayNames[day];

}  
3.2 索引查找

商品特征描述：100种商品；  
var table=[“50g,山东聊城，20170503”,“150g,山东青岛，20170809”，“100g,北京，20170810”......];(10条)

var discribeTable=[

{

“id":””apple1”,

“character”:table[0]

},

{

“id":””apple2”,

“character”:table[2]

}

]

function randerData(typeId){

$(discribeTable).each(function(idx,val){

If(typeId==val.id){

......

}

})

}  
  
使用索引查询的主要优点就是代码的**可读性大为增强,**可维护性也更好  
  
  
3.3 分段查找（阶梯访问表格）它适用于数据不是一个固定的值，而是一个范围的问题；比如将百分制成绩转成五级分制（我们用的优、良、中、合格、不合格，西方用的A、B、C、D和F），假定转换关系是当成绩在90-100区间，判为A，成绩在80-90区间，判为B，成绩在70-80区间，判为C，成绩在60-70区间，判为D，成绩在60以下，判为F（failure）  
var  rangeLimit = [  60.0, 75.0, 85.0, 95.0,100.0 ];

var grade = ["不合格","合乎要求", "良好","优秀" ,"卓越"];

function judge(score){

for(var i=0;i< rangeLimit.length;i++){

if(score<= rangeLimit[0] ){

return grade[0];

}else if(score<= rangeLimit[i]&&score>rangeLimit[i-1]){

return grade[i];

}

}

}