# QTP experience (调试太不方便，看不到内部，考虑开源的产品)

## OR

Object Repository

QTP的对象识别

TO就是被添加到对象库中的对象  
RO其实就是被测试软件在运行时实际所运行的那个对象。  
GetTOProperty(): 获取对象库中某个对象的某个属性的值  
GetTOProperties(): 获取对象库中某个对象的所有属性的值。  
SetTOProperty(): 设置对象库中某个对象的某个属性的值  
GetROProperty(): 获取实际在运行时的某个对象的某个属性的值(不是从对象库里面获取)

## DP

描述性编程

不如OR强大。就是属性赋值的一个过程。

有一定的灵活性。在项目中少用。

## VBScript

### 调试脚本工具

打开命令行，进入到脚本的父文件夹，执行命令：

Wscript.exe /X test.vbs

选择调试器

最好有VS

设置断点

<http://www.cnblogs.com/killmyday/archive/2009/09/13/1565729.html>

用Visual Studio 2008进行调试vbs，如果QTP中的脚本又调用了外部的vbscript脚本，需要使用PowerDebug

### 脚本调试技巧

如果QTP脚本需要执行并调用vbs，不能使用ExecuteFile filepath，这样的话，不能调试vbs中类方法，要用LoadFunctionLibrary。

但是如果vbs中有类存在的话，LoadFunctionLibrary不能进入类中。需要用一个类之外的方法实例化这个类，再从qtp脚本中执行。这样就能调试到vbs的类中方法。

另外一个难题是watch下不能看到一个object内部的内容。考虑使用Visual Studio 2008，但是如何引用呢。

### Design Patterns 设计模式

#### Singleton 单例模式

#### Factory 工厂模式

#### Command 命令包装模式

#### Callback 回调模式

#### Function Pointer 函数指针模式

#### Class Inheritance 类的继承模式

## Framework

### 自动化测试框架的理解

#### 对每个脚本写法定义严格的规范，定制共享函数库

#### 使用共享对象库，让对象库不重复混乱

#### 对部分关键字分离出测试脚本

#### 脚本实行可配置

#### 测试执行管理

#### 状态监控

#### 报表管理

### 用例解析测试

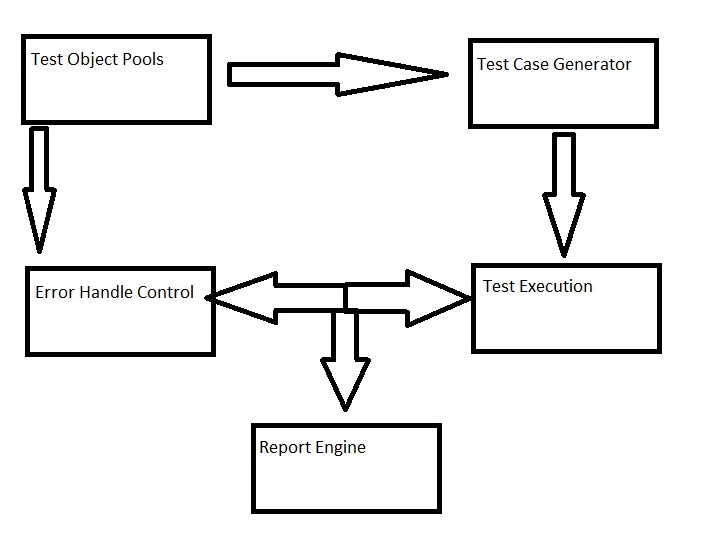
把测试脚本抽象到测试用例层面，让测试用例直接驱动自动化测试脚本，而不是每一次更改都需要重新转化测试脚本。

把用例的标准规范进行统一化，把用例的每个步骤进行标准化的拆分:

* 测试对象(具体的对象关键字)
* 测试行为(具体的行为函数名)
* 测试数据(具体的测试数据)



#### 框架设计总图



Test Object Pools:

测试对象池，用于存放所有的测试对象，并供用例层进行读取

Test Case Generator:

自动化测试用例生成器，方便测试人员根据一定的标准来完成用例

Error Handle Control:

错误处理控制器，对错误实时监控，放生错误会进行相应的控制

Test Execution:

测试执行调度，利用AOM自动启动QTP执行脚本，并进行过程控制

Report Engine:

整个测试运行完毕，生成完整的测试报告

## Common function

### GUI层

Baidu Search page class

**Class** BaiduSearch

*' set scripting dictionary object*

**Private** scripting\_dictionary

*' Page elements object*

**Public** **Property** **Get** ChildObjects()

**Set** ChildObjects = scripting\_dictionary

**End** **Property**

**Public** **Property** **Let** ChildObjects(**ByVal** m\_objects)

**Set** scripting\_dictionary = m\_objects

**End** **Property**

**Public** **Function** Init()

ChildObjects = **CreateObject**("Scripting.Dictionary")

**With** ChildObjects

.Add "Browser", Browser("name:=百度一下，你就知道")

.Add "Page", ChildObjects.item("Browser").Page("title:=百度一下，你就知道")

.Add "SetSearchContext", ChildObjects.item("Page").WebEdit("html id:=kw")

.Add "Submit", ChildObjects.item("Page").WebButton("html id:=su")

**End** **With**

Init = **IsObject**(ChildObjects)

**End** **Function**

*' Enter words in search context*

**Public** **Function** setSearchContext(searchWords)

**If** **IsObject**(ChildObjects) **Then**

ChildObjects.item("SetSearchContext").Set searchWords

**End** **If**

**End** **Function**

*' Submit search form*

**Public** **Function** Submit()

**If** **IsObject**(ChildObjects) **Then**

ChildObjects.item("Submit").Submit

**End** **If**

**End** **Function**

**End** **Class**

**但是要QTP能够调用还需要创建对象，QTP不能实例化脚本中的类，只能调用对象中的方法和属性**

*' Initialize baiduSearch*

**Public** **Function** CreateBaiduSearch()

**Dim** init\_BaiduSearch

**Set** init\_BaiduSearch = **New** BaiduSearch

**Set** CreateBaiduSearch = init\_BaiduSearch

**End** **Function**

**这样就构成了baidu\_search\_page.vbs**

**QTP中写入这样的调用函数**

**Public** **Function** SearchContext()

**Dim** baidu

**Set** baidu = CreateBaiduSearch

baidu.Init()

baidu.setSearchContext("QTP")

baidu.Submit()

**End** **Function**

**再去Call这个函数执行就可以了。**

**这样也带来了问题：**

**Q1:每个脚本中都需要实例化类吗？**

**Q2：脚本中如果想调用其他脚本中的方法，该怎么做？**

**Q3：脚本如果不通过函数库手动加载，如何动态加载脚本？**

### 判断对象是否存在

*' \*\*\*\*\*\* Common function: Check all objects in GUI level are existed, and return the result \*\*\*\*\*\*\*\*\*\*\*\*\**

**Public** **Function** IsContextLoaded(**ByRef** htContext)

**Dim** allExist, ix, aItems, aKeys, strDetails, strAdditionalRemarks

**Dim** intStatus

allExist = **True**

aItems = htContext.Items

aKeys = htContext.Keys

**For** ix = 0 **To** htContext.Count - 1

IsContextLoaded = aItems(ix).Exist(0)

strDetails = strDetails & vbNewLine & "Object #" & ix + 1 & ": '" & aKeys(ix) & "' was"

**If** IsContextLoaded **Then**

strDetails = strDetails & ""

strAdditionalRemarks = ""

**Else**

strDetails = strDetails & " not"

strAdditionalRemarks = " Please check the object properties"

allExist = **False**

**End** **If**

IsContextLoaded = IsContextLoaded **And** allExist

**Select** **Case** IsContextLoaded

**Case** **True**:

intStatus = micPass

**Case** **False**:

intStatus = micWarning

**End** **Select**

strDetails = strDetails & " found." & strAdditionalRemarks

**Next**

Reporter.ReportEvent intStatus, "IsContextLoaded", strDetails

**End** **Function**

### 引用外部脚本文件

[**http://www.iquicktest.com/vbs-include-file.html**](http://www.iquicktest.com/vbs-include-file.html)

**还没有应用到自己的脚本中**

**'\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

**'使VBSCRIPT支持include外部函数文件**

**'通过ExecuteGlobal载入外部文件内容**

**'\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

**Option Explicit**

**' Include函数，通过FSO组件读取外部函数文件内容**

**' 通过ExecuteGlobal载入**

**Sub include(file)**

**Dim fso, f, str**

**Set fso = CreateObject("Scripting.FileSystemObject")**

**Set f = fso.OpenTextFile(file, 1)**

**str = f.ReadAll**

**f.Close**

**ExecuteGlobal str**

**End Sub**

**'**

**'**

**' 通过include方式载入iquicktest.vbs文件类库**

**Function LoadVBS**

**'include "C:\QTP\Common\iquicktest.vbs"**

**include "C:\QTP\Common\IsContextLoaded.vbs"**

**End Function**

**Q1: IsContextLoaded.vbs 总是2行1列，与数据不符。再调试**

### FSO组件在VBS中的常用函数

**http://www.iquicktest.com/vbscript-com-fso.html**

**Dim objFso**

**' creating the file system object**

**Set objFso = CreateObject ("Scripting.FileSystemObject")**

**' Create a new txt file**

**' Parameters:**

**' FilePath - location of the file and its name**

**Function CreateFile (StrFilePath)**

**' variable that will hold the new file object**

**Dim NewFile**

**' create the new text File**

**Set NewFile = objFso.CreateTextFile(StrFilePath, True)**

**Set CreateFile = NewFile**

**End Function**

**' Check if a specific file exist**

**' Parameters:**

**' FilePath - location of the file and its name**

**Function CheckFileExists (StrFilePath)**

**' check if file exist**

**CheckFileExists = objFso.FileExists(StrFilePath)**

**End Function**

**' Write data to file**

**' Parameters:**

**' FileRef - reference to the file**

**' str - data to be written to the file**

**Function WriteToFile (byref FileRef,str)**

**' write str to the text file**

**FileRef.WriteLine(str)**

**End Function**

**' Read line from file**

**' Parameters:**

**' FileRef - reference to the file**

**Function ReadLineFromFile (byref FileRef)**

**' read line from text file**

**ReadLineFromFile = FileRef.ReadLine**

**End Function**

**' Read Entire Text File Test**

**' Parameters:**

**' FileRef - reference to the file**

**Function ReadTextFileTest(StrFilePath)**

**Const ForReading = 1, ForWriting = 2, ForAppending = 8**

**Dim fso, f, Msg**

**Set fso = CreateObject("Scripting.FileSystemObject")**

**Set f = fso.OpenTextFile(StrFilePath, ForReading)**

**ReadTextFileTest = f.Readall()**

**End Function**

**' Closes an open file.**

**' Parameters:**

**' FileRef - reference to the file**

**Function CloseFile (byref FileRef)**

**FileRef.close**

**End Function**

**' Opens a specified file and returns an object**

**' that can be used to read from, write to, or append to the file.**

**' Parameters:**

**' FilePath - location of the file and its name**

**' mode options are:**

**' ForReading - 1**

**' ForWriting - 2**

**' ForAppending - 8**

**Function OpenFile (StrFilePath,mode)**

**' open the txt file and retunr the File object**

**set OpenFile = objFso.OpenTextFile(StrFilePath, mode, True)**

**End Function**

**' Copy file.**

**' Parameters:**

**' FileRef - reference to the file**

**Sub FileCopy ( StrFilePathSource,StrFilePathDest)**

**' copy source file to destination file**

**objFso.CopyFile StrFilePathSource, StrFilePathDest**

**End Function**

**' Delete a file.**

**' Parameters:**

**' FileRef - reference to the file**

**Sub FileDelete ( StrFilePath)**

**' copy source file to destination file**

**objFso.DeleteFile ( StrFilePath)**

**End Function**

**' Create Folder if Not Exists.**

**Function ReportFolderStatus(fldr)**

**Dim fso, msg**

**Set fso = CreateObject("Scripting.FileSystemObject")**

**If Not (fso.FolderExists(fldr)) Then**

**Set f = fso.CreateFolder(fldr)**

**End If**

**End Function**

**ReportFolderStatus("C:\iquicktest")**

**' Count Number of Lines in txt file**

**Function NumberOfLines(FileName)**

**i = 0**

**Set objFSO = CreateObject("Scripting.FileSystemObject")**

**Set objTextFile = objFSO.OpenTextFile(FileName, ForReading)**

**Do Until objTextFile.AtEndOfStream**

**Redim Preserve arrFileLines(i)**

**arrFileLines(i) = objTextFile.ReadLine**

**i = i +1**

**Loop**

**NumberOfLines = UBound(arrFileLines)**

**objTextFile.Close**

**End Function**

**msgbox NumberOfLines("C:\iquicktest\vbs.txt")**

## 工具

### Test Design Studio

创建QTP工程

在Startup Script下创建函数库

在Tests下创建QTP脚本

Build Solution或者是单个Test，都能自动部署到UFT(QTP)中

### 脚本编写工具：TDS不能智能感知自定义变量

# Selenium 2 – Web Driver

## 开源应用架构之​Selenium WebDriver

http://www.infoq.com/cn/news/2011/06/selenium-arch

<http://www.infoq.com/cn/news/2011/07/selenium-arch-2>

## Chromedriver

下载ChromeDriver.exe，在System.setProperty("webdriver.chrome.driver",

"/Program Files (x86)/Google/Chrome/Application/26.0.1410.64/chromedriver.exe");因为chrome是依赖这个exe启动的。

## DataProvider

需要申明Object[][]这样的多维数组做provider，然后再数据方法上使用@DataProvider(name=”DP1”)，在调用方法上使用@Test(dataProvider=”DP1”)

## 用xpath查找元素

webElement.findElement(By.xpath(“path”))

## 执行javascript

String js=”alert(“hello”)”;

((JavascriptExecutor) driver).executeScript(js);也可以调用比较复杂的js

## WebDriverWait & driver.manage()等待的重要性

<http://wenku.baidu.com/view/6d0aa57e31b765ce05081427.html###>

<http://my.oschina.net/u/928852/blog/98885>

近期在回放时，总是出现找不到element的情况。这样促使我在找webElement之前先等待页面加载完

### 显式等待

WebDriverWait wait = new WebDriverWait(driver, 100);

wait.until(new ExpectedCondition<WebElement>() {

@Override

public WebElement apply(WebDriver driver) {

return driver.findElement(By.xpath(fpath));

}

}).click();

直接wait的方式是显式的，比较个体，特别在用于frame加载的时候，像login执行了js，需要frame的加载。比较好用

### 隐式等待

driver.manage().timeouts().implicitlyWait(1, TimeUnit.MINUTES);

在driver上普遍加入timeout时间，在timeout之内而没有找到的情况给予等待。

### 让隐式等待发挥更大作用

public boolean isByElementDisplayed(WebElement element, int time) {

boolean status = false;

if (element.isDisplayed() == false) {

driver.manage().timeouts().implicitlyWait(time, TimeUnit.SECONDS);

} else {

status = true;

}

return status;

}

## 加入log4j

<http://www.blogjava.net/rickhunter/articles/28133.html>

导入jar包log4j.jar

在classpath下面创建log4j.properties文件copy

log4j.rootLogger=debug, stdout, R

log4j.appender.stdout=org.apache.log4j.ConsoleAppender

log4j.appender.stdout.layout=org.apache.log4j.PatternLayout

# Pattern to output the caller's file name and line number.

log4j.appender.stdout.layout.ConversionPattern=%5p [%t] (%F:%L) - %m%n

log4j.appender.R=org.apache.log4j.RollingFileAppender

log4j.appender.R.File=example.log

log4j.appender.R.MaxFileSize= 1024K

# Keep one backup file

log4j.appender.R.MaxBackupIndex=1

log4j.appender.R.layout=org.apache.log4j.PatternLayout

log4j.appender.R.layout.ConversionPattern=%p %t %c - %m%n

要在某个class加入自己的日志信息：

**private** **static** Logger *LOGGER* = Logger.*getLogger*(testOrder.**class**);

用法*LOGGER.debug*

*LOGGER.error*

*…*

# SQL

## 复杂语句

### Declare @name varchar(50)

别忘了@

### If @name is not null

别忘了is

### Return

结束整个函数

### CONVERT(varchar(10), @InquiryId)

把数字转换为Text

### While

循环

### Waitfor delay ’00:03’

在达到指定时间或时间间隔之前，或者指定语句至少修改或返回一行之前，阻止执行批处理、存储过程或事务。

Refer: <http://msdn.microsoft.com/zh-cn/library/ms187331.aspx>

Waitfor time ’22:00’

### Isnull()

<http://database.51cto.com/art/201009/224323.htm>

ISNULL

使用指定的替换值替换 NULL。

Isnull(@name, ‘’)

如果@name是null，则返回空字符串。如果@name有值，则返回值

## [SQL字符串操作汇总](http://www.cnblogs.com/lovelace821/archive/2009/11/05/1596691.html)

--将字符串中从某个字符开始截取一段字符，然后将另外一个字符串插入此处  
select stuff('hello,world!',4,4,'\*\*\*\*')   --返回值hel\*\*\*\*orld!

--返回从指定位置开始指定长度的字符串  
select substring('Hello,World!',2,10)   --返回值ello,World

--将字符串中某段字符替换为指定的字符串  
select replace('hello,world!','ll','aa') --返回值heaao,world!

--去除字符串中左边的空格  
select ltrim('   hello,world!')    --返回值hello,world!

--去除字符串中左边的空格  
select ltrim('hello,world!   ')    --返回值hello,world!

--去除字符串中左边和右边的空格  
select ltrim('    hello,world!   ')   --返回值hello,world!

--将NULL值替换为指定字符  
select isnull('a',null)     --返回值a

--转换数据类型  
select cast('2007-10-11' as datetime)   --返回值2007-10-11 00:00:00.000  
select convert(datetime,'2007-10-11')   --返回值2007-10-11 00:00:00.000

--获取字符串长度  
select len('hello,world!')    --返回值12

--获取字符串的前3个字符  
select left('hello,world!',3)    --返回值hel

--获取字符串的后3个字符  
select right('hello,world!',3)    --返回值ld!

--去除字符串的前3个字符  
select right('hello,world!',(len('hello,world!')-3)) --返回值lo,world!

--去除字符串的后3个字符  
select left('hello,world!',(len('hello,world!')-3)) --返回值hello,wor

--获取在该字符串中某字符串的位置(返回数字)   
select charindex('e','hello,world!')   --返回值2

--返回从第2个字符开始前4个字符  
select left(right('[哈哈哈哈]aaa',len('[哈哈哈哈]aaa')-1),4) --返回值哈哈哈哈

--返回字符的小写形式  
select lower('HELLO,WORLD!')    --返回值hello,world!

--返回字符的大写形式  
select UPPER('hello,world!')    --返回值HELLO,WORLD!

--用第三个表达式替换第一个字符串表达式中出现的所有第二个指定字符串表达式的匹配项  
（如果其中有一个输入参数属于 nvarchar 数据类型，则返回 nvarchar；否则返回 varchar。如果任何一个参数为 NULL，则返回 NULL。）  
SELECT REPLACE('Hello,World!','l','a')   --返回值Heaao,Worad!  
SELECT REPLACE('Hello,World!','l','')   --返回值Heo,Word!  
SELECT REPLACE('Hello,World!','l',null)   --返回值NULL

--以右边参数数值次数复制字符表达式  
select REPLICATE('Hello,World!',4)   --返回值Hello,World!Hello,World!Hello,World!Hello,World!

--返回反转后的字符串  
select REVERSE('Hello,World!')    --返回值!dlroW,olleH

--使用DIFFERENCE时，两个字符串发音越相似（仅限于英文字符），返回值越大（返回值在0-4之间）  
DIFFERENCE('sun','san')    --返回值4  
DIFFERENCE('sun','safdsdf')   --返回值3  
DIFFERENCE('sun','dgffgfdg')   --返回值0

--将带小数点的数字类型转换为可设定长度可设定小数位的四舍五入后的字符串  
SELECT STR(123.34584, 7, 3)   --返回值123.346  
--当设定长度值小于整数部位长度时，字符串将返回设定长度个\*  
SELECT STR(123333.34584, 5, 4)   --返回值\*\*\*\*\*

--===================================================================================

--=====================================数字操作汇总==================================

--返回指定数字的最大整数  
select floor(123456.1234)   --返回值123456

--返回不带小数部分并且不小于其参数的值的最小数字。如果参数是一个空序列，则返回空序列  
select ceiling(123.010)    --返回124  
select ceiling(null)    --返回NULL

--返回四舍五入后的最接近该数值的数值  
select round(126.018,2)    --返回126.12

--返回一个0-1之间的FLoat类型的随机数  
select rand()     --返回0.94170703697981

--返回圆周率PI的值  
SELECT PI()     --返回3.14159265358979

# VBA for excel

## Connect Database

### Tool->Reference Microsoft ActiveX Data Objects 6.1 Library

If Dim Conn As ADODB.Connection not possible, you should add ADODB reference.

You need to set a reference to "Microsfot ActiveX Data Objects" (ADO for short) first.  
  
In the VBE select Tools -> References... From the dialog box that pops up, scrolll down until you find an entry that reads something like Microsfot ActiveX Data Objects 2.7 Library (the number might be different - pick the highest number you can see). Check the tick box next to this entry and then click ok. You should be good to go now.

<http://p2p.wrox.com/excel-vba/39128-adodb-connection-user-defined-type-not-defined.html>

### Recordset.getRows()

获取结果个数UBound(Recordset.getRows(), 2)

http://www.cnblogs.com/spring4/archive/2007/04/30/2483818.html

## FSO

### Tool->Reference Microsoft Scripting Runtime

<http://www.thevbprogrammer.com/Ch06/06-09-FSO.htm>

## Dictionary

VBA中Dictionary对象使用小结

Dim dict

' 创建Dictionary  
Set dict = CreateObject("Scripting.Dictionary")

' 增加项目  
dict.Add "A", 300  
dict.Add "B", 400  
dict.Add "C", 500

' 统计项目数  
n = dict.Count

' 删除项目  
dict.Remove ("A")

' 判断字典中是否包含关键字  
dict.exists ("B")

' 取关键字对应的值，注意在使用前需要判断是否存在key，否则dict中会多出一条记录  
Value = dict.Item("B")

' 修改关键字对应的值,如不存在则创建新的项目  
dict.Item("B") = 1000  
dict.Item("D") = 800

' 对字典进行循环  
k = dict.keys  
v = dict.Items  
For i = 0 To dict.Count - 1  
  key = k(i)  
  Value = v(i)  
  MsgBox key & Value  
Next

' 删除所有项目  
dict.Removeall

<http://abaper.blogbus.com/logs/1596438.html>

## 时间函数

Now         返回一个 Variant (Date)，根据计算机系统设置的日期和时间来指定日期和时间。

Date        返回包含系统日期的 Variant (Date)。

Time        返回一个指明当前系统时间的 Variant (Date)。

Timer       返回一个 Single，代表从午夜开始到现在经过的秒数。

TimeSerial(hour, minute, second) 返回一个 Variant (Date)，包含具有具体时、分、秒的时间。

DateDiff(interval, date1, date2[, firstdayofweek[, firstweekofyear]]) 返回 Variant (Long) 的值，表示两个指定日期间的时间间隔数目

Second(time) 返回一个 Variant (Integer)，其值为 0 到 59 之间的整数，表示一分钟之中的某个秒

Minute(time) 返回一个 Variant (Integer)，其值为 0 到 59 之间的整数，表示一小时中的某分钟

Hour(time)  返回一个 Variant (Integer)，其值为 0 到 23 之间的整数，表示一天之中的某一钟点

Day(date)   返回一个 Variant (Integer)，其值为 1 到 31 之间的整数，表示一个月中的某一日

Month(date) 返回一个 Variant (Integer)，其值为 1 到 12 之间的整数，表示一年中的某月

Year(date) 返回 Variant (Integer)，包含表示年份的整数。

Weekday(date, [firstdayofweek]) 返回一个 Variant (Integer)，包含一个整数，代表某个日期是星期几

## Format Format(date, “yyyy-dd-mm”)

<http://msdn.microsoft.com/en-us/library/office/gg251755.aspx>

Dim MyTime, MyDate, MyStr

MyTime = #17:04:23#

MyDate = #January 27, 1993#

' Returns current system time in the system-defined long time format.

MyStr = Format(Time, "Long Time")

' Returns current system date in the system-defined long date format.

MyStr = Format(Date, "Long Date")

MyStr = Format(MyTime, "h:m:s") ' Returns "17:4:23".

MyStr = Format(MyTime, "hh:mm:ss AMPM") ' Returns "05:04:23 PM".

MyStr = Format(MyDate, "dddd, mmm d yyyy") ' Returns "Wednesday,

' Jan 27 1993".

' If format is not supplied, a string is returned.

MyStr = Format(23) ' Returns "23".

' User-defined formats.

MyStr = Format(5459.4, "##,##0.00") ' Returns "5,459.40".

MyStr = Format(334.9, "###0.00") ' Returns "334.90".

MyStr = Format(5, "0.00%") ' Returns "500.00%".

MyStr = Format("HELLO", "<") ' Returns "hello".

MyStr = Format("This is it", ">") ' Returns "THIS IS IT".

## Regular expression

<http://blog.csdn.net/yan11cn/article/details/5004279>

http://iregex.org/blog/regex-in-excel-vba.html

## Array

<http://www.excelpx.com/home/show.aspx?id=1399>

### UBound 其值为指定的[数组](http://baike.baidu.com/view/209670.htm)维可用的最大下标

返回一个 Long 型数据，其值为指定的[数组](http://baike.baidu.com/view/209670.htm)维可用的最大下标。  
UBound(arrayname[, dimension])

arrayname 必需的。[数组变量](http://baike.baidu.com/view/1349297.htm)的名称，遵循标准变量[命名约定](http://baike.baidu.com/view/553722.htm)。

dimension 可选的；Variant (Long)。指定返回哪一维的上界。1 表示第一维，2 表示第二维，以此类推。如果省略 dimension，就认为是 1。

<http://www.w3school.com.cn/vbscript/func_ubound.asp>

### <http://www.cpearson.com/excel/vbaarrays.htm> 数组常用函数和用法

## 退出函数或者子过程

Exit Sub

Exit Function

## 字符串常用函数列表

http://www.excelpx.com/home/show.aspx?id=583

Trim(string)            去掉string左右两端空白

Ltrim(string)           去掉string左端空白

Rtrim(string)           去掉string右端空白

Len(string)             计算string长度

Left(string, x)         取string左段x个字符组成的字符串

Right(string, x)        取string右段x个字符组成的字符串

Mid(string, start,x)    取string从start位开始的x个字符组成的字符串

Ucase(string)           转换为大写

Lcase(string)           转换为小写

Space(x)                返回x个空白的字符串

Asc(string)             返回一个 integer，代表字符串中首字母的字符代码

Chr(charcode)           返回 string,其中包含有与指定的字符代码相关的字符



## VBA集锦<http://www.360doc.com/content/11/0214/12/3660125_92910704.shtml>

## 状态比较结果

http://winofsql.jp/pinpoint/vbscripttruefalse.htm

