String Functions

**Len Function-**-Returns the number of characters in a string or the number of bytes required  to store a variable.   
**Syntax:** Len(string | varname)  
**Example:**  
Str="Welcome to the World of QTP"  
Print **Len**(Str)            
Print  **Len**("Good Morning")  

**LCase Function**--Returns a string that has been converted to lowercase.  
**Syntax:** LCase(string)  
**Example:**   
Str="Welcome to the World of QTP"  
Print **LCase**(Str)   
Print  **Lcase**("Good Morning")   

**UCase Function**--Returns a string that has been converted to uppercase.  
**Syntax**: UCase(string)  
**Example**:   
Str="Welcome to the World of QTP"  
Print **Ucase**(Str)   
Print  **Ucase**("Good Morning")              

**Left Function**--Returns a specified number of characters from the left side of a string.  
**Syntax:**Left(string, length)  
**Example:**   
Str="Welcome to the World of QTP"  
print  **Left**(Str,3)                                                     
Print **Left**("Good Morning",4)                              

**Right Function**--Returns a specified number of characters from the right side of a string.  
**Syntax:** Right(string, length)  
**Example:**   
Str="Welcome to the World of QTP"  
print  **Right**(Str,12)

**Mid Function**--Returns a specified number of characters from a string.  
**Syntax:** Mid(string, start[, length])  
**Example:**   
Str="Welcome to the World of QTP"  
print  **Mid**(Str,9,12)                                              
Print **Mid**("Good Morning",6,7)                          
Print **Mid**("Good Morning",6)                        

**Replace Function--**Returns a string in which a specified substring has been replaced  with  another substring a specified number of times.  
**Syntax:** Replace(expression, find, replacewith[, start[, count[, compare]]])  
**Example:**  
Str="Welcome to the World of QTP"  
print **Replace**(Str, "to","into")                                                      
Print **Replace**("Good Morning","Morning","Night")                  
Print **Replace**("Quick Test ProfeSsional","s","x",5,3,0)              
Print **Replace**("Quick Test ProfeSsional","s","x",5,3,1)                    

**Space Function--**Returns a string consisting of the specified number of spaces.  
**Syntax:** Space(number)  
**Example:**   
Str="Welcome to the World of QTP"  
Str1="Welcome"  
Str2="to the World"  
Str3="of QTP"  
Print  Str1 & **Space**(2) & Str2 & **Space**(2) &Str3                    

**Split Function**--Returns a zero-based, one-dimensional array containing a specified number of substrings.  
**Syntax:** Split(expression[, delimiter[, count[, compare]]])  
**Example:**   
Str="Welcome to the World of QTP"  
SArray=**Split**(Str," ",-1)  
**For**  i= 0 **to** **UBound**(SArray)  
    Print  SArray(i)  
**Next**

**StrComp Function**--Returns a value indicating the result of a string comparison.   
**Syntax:** StrComp(string1, string2[, compare])  
**Example:**  
Str="Welcome to the World of QTP"  
Str1 = "QTP"  
Str2 = "qtp"                                     
Print **StrComp**(Str1, Str2, 1)                             
Print **StrComp**(Str1, Str2, 0)                             
Print **StrComp**(Str2, Str1)                                 
Print **StrComp**(Str1, Str2)                                 

**StrReverse Function**--Returns a string in which the character order of a specified  string  is  reversed.  
**Syntax:** StrReverse(string1)  
**Example:**   
Str="Welcome to the World of QTP"  
print  **StrReverse**(Str)                                        
Print  **StrReverse**("Quick Test ProfeSsional")                 

**LTrim; RTrim; and Trim Functions**--Returns a copy of a string without leading spaces  (LTrim), trailing spaces (RTrim), or both leading and trailing spaces  (Trim).  
**Syntax:**  LTrim(string)  
       RTrim(string)  
       Trim(string)  
**Example:**Print **Ltrim**("       Welcome to QTPWorld.com                 ")   
Print **Rtrim**("           Welcome to QTPWorld.com                          ")                      
Print     **trim**("              Welcome to QTPWorld.com               ")                

**InStr Function**--Returns the position of the first occurrence of one string within another.  
**Syntax:** InStr([start, ]string1, string2[, compare])  
**Example:**   
Str="How do you DO ?"  
Print **Instr**(1, Str, "DO", 1)                  
Print **Instr**(1, Str, "DO", 0)                  
Print **Instr**(6, Str, "do", 0)

**InStrRev Function**--Returns the position of an occurrence of one string within another, from the end of string.   
**Syntax:** InStrRev(string1, string2[, start[, compare]])  
**Example:**   
Str="How do you DO ?"  
Print **InStrRev**(Str,"DO",-1,1)                  
Print **InStrRev**(Str,"do",-1,0)              
Print **InStrRev**(Str,"DO",13,0)              
Print **InStrRev**(Str,"DO",10,1)            

**Control Flow Functions**

**Do...Loop Statement**-- Repeats a block of statements while a condition is True or until a condition becomes True.  
**Syntax:**  **Do** [{**While** | **Until**} condition]  
   [statements]  
   [**Exit** **Do**]  
   [statements]  
**Loop**  
**OR**  
**Do**  
   [statements]  
   [**Exit** **Do**]  
   [statements]  
**Loop** [{**While** | **Until**} condition]

**Example:**Counter = 1  
 **Do** **While** Counter < 4  
 Total =**Inputbox**("Please enter the total marks in numbers")  
                         **If**   Total < 50 **Then**  
                                    **MsgBox**  "Fail"  
                               **ElseIf**  Total >=50 **and** Total <=60   **then**  
                                    **Msgbox** "Second Class"  
                    **ElseIf**  Total >60 **and** Total <80   **then**  
                                    **Msgbox** "First Class"  
                        **ElseIf**  Total >=80  **then**  
                                    **Msgbox** "Distinction"  
                         **Else**  
                        **Msgbox** "Invalid Marks"  
                               **End** **If**

Counter = Counter + 1  
**Loop**  
  
**Example:**  
Counter = 1  
 **Do**    
 Total =**Inputbox**("Please enter the total marks in numbers")  
                         **If**   Total < 50 **Then**  
                                    **MsgBox**  "Fail"  
                               **ElseIf**  Total >=50 **and** Total <=60   **then**  
                                    **Msgbox** "Second Class"  
                    **ElseIf**  Total >60 **and** Total <80   **then**  
                                    **Msgbox** "First Class"  
                        **ElseIf**  Total >=80  **then**  
                                    **Msgbox** "Distinction"  
                         **Else**  
                        **Msgbox** "Invalid Marks"  
                               **End** **If**

Counter = Counter + 1  
**until** Counter < 4

**Example:   
Do** **While** Counter < 10  
 Counter=counter + 1  
 **If**  counter = 6 **Then**

**Exit** **Do**  
 **End** **If**  
 Print Counter  
  **Loop**  
  
**Example:   
Do**  
    Counter=Counter + 1  
 **If**  Counter = 6 **Then**  
**Exit** **Do**  
 **End** **If**  
 Print Counter  
**Loop** **Until** counter >11

**For...Next Statement**--Repeats a group of statements a specified number of times.  
**Syntax:** **For** counter = start **To** **end** [**Step** **step**]  
    [statements]  
    [**Exit** **For**]  
    [statements]  
**Next**  
**Example:**Counter = 1  
**For**  Counter = 1 **to** 4  
    Total =**Inputbox**("Please enter the total marks in numbers")  
                         **If**   Total < 50 **Then**  
                                    **MsgBox**  "Fail"  
                               **ElseIf**  Total >=50 **and** Total <=60   **then**  
                                    **Msgbox** "Second Class"  
                    **ElseIf**  Total >60 **and** Total <80   **then**  
                                    **Msgbox** "First Class"  
                        **ElseIf**  Total >=80  **then**  
                                    **Msgbox** "Distinction"  
                          **End** **If**

**Next**

**Example:** Counter = 1  
 Result  = 0  
    **For**  Counter = 1 **to** 10  **step** 3   
     Result=Result + Counter  
     Print Result  
    **Next**  
      
**Example:**   
**For** Counter=1 **to** 10  
 **If**  Counter=6 **Then**  
**Exit** **for**  
     **End** **If**  
 Print Counter  
**Next**

**For Each...Next Statement**--Repeats a group of statements for each  element in an  array  or  collection.  
**Syntax:**  
 **For** **Each** element **In** group  
   [statements]  
   [**Exit** **For**]  
   [statements]  
**Next** [element]  
   
**If...Then...Else Statement--**Conditionally executes a group of statements, depending on the value of an expression.  
**Syntax:  
If** condition **Then**  
   [statements]  
[**ElseIf** condition-n **Then**  
   [elseifstatements]] . . .  
**End** **If**   
**Example:**Total = **Inputbox**("Please enter the total marks in number ")  
 **If**   Total >= 50 **Then**  
            **MsgBox**  "Pass"  
**Else**   
         **Msgbox** "Fail"  
**End** **If**  
  
**Select Case statement**--Executes one of several groups of statements, depending on the value of an expression.   
**Syntax:  
Select** **Case** testexpression  
   [**Case** expressionlist-n  
      [statements-n]] . . .  
   [**Case** **Else** expressionlist-n  
      [elsestatements-n]]  
**End** **Select**  
**Example:**Colour =**Ucase**( **Inputbox**("Please enter the colour of your choice"))  
 **Select** **Case** Colour  
        **Case**   "RED"  
        **MsgBox**  "Colour selected is Red"  
         **Case**   "BLUE"  
         **MsgBox**  "Colour selected is Red"  
      **Case**   "GREEN"  
         **MsgBox**  "Colour selected is Red"  
         **Case**   **ELSE**  
         **MsgBox**  "Invalid Colour"  
           **End** **Select**

**While...Wend statement**--Executes a series of statements as long as a given condition is True.  
**Syntax:** **While** condition  
   Version [statements]  
**Wend**  
**Example:**    Counter =1  
    **While** Counter < 4  
 Total =**Inputbox**("Please enter the total marks in numbers")  
                         **If**   Total < 50 **Then**  
                                    **MsgBox**  "Fail"  
                               **ElseIf**  Total >=50 **and** Total <=60   **then**  
                                    **Msgbox** "Second Class"  
                    **ElseIf**  Total >60 **and** Total <80   **then**  
                                    **Msgbox** "First Class"  
                        **ElseIf**  Total >=80  **then**  
                                    **Msgbox** "Distinction"  
                          **End** **If** Counter=counter + **Wend**

**Date** **Function** Returns the current system **date**.  
Syntax: **Date**   
Example:   
print date  
  
**Time** **Function** Returns a Variant of subtype **Date** indicating the current system **time**.  
Syntax: **Time**   
Example:   
Print **Time**                                    
  
**DateAdd** **Function** Returns a **date** **to** which a specified **time** interval has been added.   
Syntax: **DateAdd**(interval, **number**, **date**)  
Example:   
Print  **DateAdd**("yyyy",1,"November 01, 2010")                  
  
**DateDiff** **Function** Returns the **number** of intervals between two dates.   
Syntax: **DateDiff**(interval, date1, date2 [,firstdayofweek[, firstweekofyear]])  
Example:   
Print **DateDiff** ("yyyy","November 01, 2008","November 01, 2009")    

**DatePart** **Function** Returns the specified part of a given **date**.  
Syntax: **DatePart**(interval, **date**[, firstdayofweek[, firstweekofyear]])  
Example:   
Print **DatePart**("yyyy","November 01, 2008")          

**Day** **Function** Returns a whole **number** between 1 **and** 31, inclusive, representing the **day** of the **month**.  
Syntax: **Day**(**date**)  
Example:   
Print  **Day**("11-01-2010")                                          
Print  **Day**("11/1/2010")                                                  
Print  **Day**("November 01, 2010")                            

**Month** **Function** Returns a whole **number** between 1 **and** 12, inclusive, representing the **month** of the **year**.  
Syntax: **Month**(**date**)  
Example:   
Print  **Month**("11-01-2010")                                          
  
**MonthName** **Function** Returns a **string** indicating the specified **month**.  
Syntax: **MonthName**(**month**[, abbreviate])  
Example:   
Print  **MonthName**(11,**true**)                                                    
Print  **MonthName**(12,**false**)                                                  
Print  **MonthName**(9)                                                                

**Weekday** **Function** Returns a whole **number** representing the **day** of the week.   
Syntax: **Weekday**(**date**, [firstdayofweek])  
Example:   
Print  **Weekday**("November 03, 2010")                                  

**WeekdayName Function** Returns a **string** indicating the specified **day** of the week.  
Syntax: **WeekdayName**(**weekday**, abbreviate, firstdayofweek)  
Example:   
Print  **WeekdayName**(4)                                  

**Year** **Function** Returns a whole **number** representing the **year**.  
Syntax: **Year**(**date**)  
Example:   
Print  **Year**("11-01-2010")                                          
Print  **Year**("11/1/2010")                                                  
Print  **Year**("November 01, 2010")                            

**Hour** **Function** Returns a whole **number** between 0 **and** 23, inclusive, representing the **hour** of the **day**.   
Syntax: **Hour**(**time**)  
Example:   
Print  **Hour**(**Now**)                                 

**Minute**  **Function** Returns a whole **number** between 0 **and** 59, inclusive, representing the **minute** of the **hour**.  
Syntax: **Minute**(**time**)  
Example:   
Print  **Minute**(**Now**)                                 

**Second** **Function** Returns a whole **number** between 0 **and** 59, inclusive, representing the **second** of the **minute**.   
Syntax: **Second**(**time**)  
Example:   
Print  **Second**(**Now**)                                 

**Now** **Function** Returns the current **date** **and** **time** according **to** the setting of your computer*'s* system date and time.  
Syntax: **Now**  
Example:   
Print  **Now**                                     

**TimeSerial** **Function** Returns a Variant of subtype **Date** containing the **time** **for** a specific **hour**, **minute**, **and** **second**.  
Syntax: **TimeSerial**(**hour**, **minute**, **second**)  
   
**minute**:    Any numeric expression.  
**second**:    Any numeric expression.  
Example:   
Print  **Timeserial**(13,30,00)                    

**TimeValue**  **Function** Returns a Variant of subtype **Date** containing the **time**.  
Syntax: **TimeValue**(**time**)  
Example:   
Print  **TimeValue**("16:30:00")            

**Array** **Function** Returns a Variant containing an **array**.   
Syntax: **Array**(arglist)  
Example:  
A = **Array**(10,20,30)  
Print  A(0)                  
Print  A(1)                  
Print  A(2)                  
  
  *'Single dimensional Array with  five elements*  
**Dim**  Num(5)  
Num(0)=10  
Num(1)=20  
Num(2)=30  
  
Print  Num(0)                  
Print  Num(1)                  
Print  Num(2)                 
  
*'multidimensional array*  
**Dim**  DNum(3,2)  
DNum(0,0)=10  
DNum(0,1)=20  
DNum(1,0)=30  
DNum(1,1)=40  
  
Print  DNum(0,0)                  
Print  DNum(0,1)                  
Print  DNum(1,0)                  
Print  DNum(1,1)                

**IsArray** **Function** Returns a Boolean value indicating whether a variable **is** an **array**.  
Syntax: **IsArray**(varname)  
Example:   
**Dim** Arr  
**Dim**  NArr  
Arr = **Array**(10,20,30)  
Print  **IsArray**(Arr)              
Print  **IsArray**(NArr)         

**LBound** **Function**  Returns the smallest available subscript **for** the indicated dimension of an **array**.  
Syntax: **LBound**(arrayname[, dimension])  
Example:   
**Dim** LArr  
LArr = **Array**(10,20,30)  
Print  **Lbound**(LArr,1)                 

**UBound** **Function**  Returns the largest available subscript **for** the indicated dimension of an **array**.  
Syntax: **UBound**(arrayname[, dimension])  
Example:   
**Dim** UArr  
UArr = **Array**(10,20,30)  
Print  **Ubound**(UArr,1)      

**Dim**  Umarr(3,2,4)  
Print  **UBound**(Umarr,1)         
Print  **UBound**(Umarr,2)         
Print  **UBound**(Umarr,3)