Strings in Visual Basic

Words, Phrases, and Spaces
Strings are a series of characters.

- Constant strings never change and are indicated by double quotes.

- Examples: "Fleebo"
  "Here is a string."
Strings are a series of characters.

敲Variable strings are a special type of variable.

敲Example: Dim Fleeb as String

敲In any of the following operators and functions you can use either a variable or constant string.
Strings have their own operators and functions.

- Operators for putting strings together.
- Functions for pulling strings apart.
- Functions for creating strings.
- Functions for modifying strings.
- Operators for finding out things about strings.
Putting Strings Together

& concatenates two strings

Examples:

“This “ & “is fun!”” becomes “This is fun!”

“Fleebl norb” & “ski” becomes “Fleebnorbski”
The + sign can be used for the same purpose, but it is better to use the & since no one will mistake it for addition.
Functions for Pulling Strings Apart

Left$ - Pulls off the leftmost characters.

Takes two arguments,
- the string
- the number of characters to pull off

Examples:
- Left$("Here is another string",10) returns “Here is an”
- Left$("6 + 67 * 9",6) returns “6 + 67”
Pulling Strings Apart

- **Right$** - Pulls off the rightmost characters.
  - Takes two arguments:
    - the string
    - the number of characters to pull off
  - **Examples:**
    - Right$("Here is another string",10) returns “her string”
    - Right$("6 + 67 * 9",6) returns “67 * 9”
Mid$ - Pulls out characters from middle.

- Takes three arguments
  - the string
  - the position to start pulling from.
  - the number of characters to pull.

Examples:
- Mid$("Here is another string",10,4) returns "noth"
- Mid$("6 + 67 * 9",5,2) returns "67"
Creating Strings

- **Str$** - Creates a string from a number.
- **CStr** - Creates a string from a number.
- **Format$** - Allows very complicated formatting of numbers.
Creating Strings

- **Format$** - Used with numbers, gives precise formatting control over strings.
  - Takes two arguments
    - A number
    - A descriptor.
Descriptors for Format

- Descriptors come in two flavors: system defined and user defined.
  - System defined include: General Number, Currency, Fixed, Standard, Percent, Scientific, Yes/No, True/False, On/Off
  - User defined are generated using another string. The other string has special characters in it that define how the number should be displayed.
User Defined Format String

Characters

0    Display a number or zero.
#    Display a number or nothing.
.    Decimal placeholder
%    Percentage placeholder.
,    Thousands separator
E    Scientific Notation
:    Time separator
/    Date separator
\”   Literal character indicator.
User Defined Format String

Characters

- \( d \) Day
- \( m \) Month
- \( y \) Year
- \( h \) Hour
- \( n \) Minute
- \( s \) Second
- \( w \) Day of Week
- \( q \) Quarter
Some Examples of Format

\[
\text{format}(10201.2, "000000.00") \rightarrow 010201.20 \\
\text{format}(11.1, "#####.##") \rightarrow 11.1 \\
\text{format}(10010021, "#####,#####,#####") \rightarrow 10,010,021 \\
\text{format}(1000000, ".##E##") \rightarrow 1.00E6
\]
Dim curr_time as double
curr_time = Now
format(curr_time,"mm/dd/yy hh:nn") -> 03/06/95 09:15
format(curr_time,"m/d/yy h:n") -> 3/6/95 9:15
format(cur_time,""ddddd, mmm dd, yyyy")
Wednesday, Dec. 06, 1995
Functions for Modifying Strings

- **UCase$** - Make every letter upper case.
- **LCase$** - Make every letter lower case.
- **Trim$** - Removes leading and trailing spaces from the string.
Modifying Strings

£ UCase$ - Make every letter upper case.
£ Takes one argument, the string.
£ Examples
  £ UCase$("Here") returns "HERE"
  £ UCase$("Oh Boy") returns "OH BOY"
Modifying Strings

LCase$ - Makes every letter lower case.

- Takes one argument, the string.
- Examples
  - LCase$("Here") returns "here"
  - LCase$("Oh Boy") returns "oh boy"
Modifying Strings

\[ \text{Trim$ - Removes leading and trailing spaces from string} \]
\[ \text{Takes one argument, the string} \]
\[ \text{Examples Trim$}(\text{"Here is a string"}) \]
\[ \text{returns "Here is a string"} \]
\[ \text{Also comes in LTrim$ and RTrim$, for removing only one end or the other.} \]
Functions for Finding Things Out About Strings

- **Len** - Finds out how long the string is.
- **InStr** - Finding out if one string is in another.
- **StrComp** - Find out which string is “bigger”.
Finding Out about Strings

- **Len** - Finds out how long the string is.
  - Takes one argument
  - A string
- Returns the number of characters in the string.
Finding Out about Strings

**InStr** - Finding out if one string is in another.

- Takes two arguments
  - The string to look in.
  - The string to look for.
- Returns the location of the first occurrence of string two in string one.
- If the location equals zero, there is no occurrence of string one in string two.
Finding Out about Strings

StrComp - Which string is “bigger”.

Takes two arguments:

- string one
- string two

Returns:

- -1 if string one < string two
- 0 if they are equal
- 1 if string two < string one
Summary

- String operations can be very useful.
- VB gives particularly strong support for strings.