COMMENTING YOUR PROGRAMS WITH STYLE

By John Harrison -
JCHarrison Mar 29, 2010

Much has been written about writing readable code, but little has been written about writing readable comments. The following Liberty Basic Newsletters talk about writing readable code and commenting it.

#4 - Towards Writing More Readable Code - Part 1
#46 - Block Structured Code and Development of Large Programs with LB
#105 - Documenting Your Code the Easy Way!
#119 - A Dozen Rules for Writing Code

Alyce Watson's Liberty Basic Encyclopedia article How to Write Well is a great source for guidance on how to write readable comments. The next step in commenting your code is to unleash your imagination and create comments with style. Stylish comments can enhance the readability of your code and can aid in making your programs easier to understand. After writing a program that you desire to share with others, impress them with your code and wow them with your stylish comments.

Some of the rules for writing readable code can be applied to writing stylish comments. The most important of these is white space (blank lines, indenting, and spacing within a line). No one likes to read through a large block of text to find the information they need. Use white space to break text into smaller pieces so it can be easily scanned for content or skipped to get to the desired information.

Using the rules for writing readable code, turn this:

```
'This stylebitsDemo6.bas is part of the
'ongoing series
'Liberty BASIC Newsletter: Stylebits Corner
'(c) 2005 Janet Terra
'http://babek.info/libertybasicfiles/lbnews/
'This stylebitsDemo6.bas accompanies
'The Liberty BASIC Newsletter Issue #134.
'stylebitsDemo1.bas and stylebitsDemo2.bas
'can be found in
'The Liberty BASIC Newsletter Issue #130.
'stylebitsDemo3.bas can be found in
'The Liberty BASIC Newsletter Issue #131.
'stylebitsDemo4.bas can be found in
'The Liberty BASIC Newsletter Issue #132.
'stylebits Demo5.bas can be found in
'The Liberty BASIC Newsletter Issue #133.
```

Into stylish comments like this:

```
' Program:  stylebitsDemo6.bas
'
' Purpose:  Part of the ongoing Liberty BASIC Newsletter series:
'              Stylebits Corner
'              (c) 2005 Janet Terra
'              http://babek.info/libertybasicfiles/lbnews/
'
' Files:  The following files can be found in:
'            stylebitsDemo1.bas - Liberty BASIC Newsletter Issue #130.
'            stylebitsDemo2.bas - Liberty BASIC Newsletter Issue #130.
'            stylebitsDemo3.bas - Liberty BASIC Newsletter Issue #131.
'            stylebitsDemo4.bas - Liberty BASIC Newsletter Issue #132.
'            stylebitsDemo5.bas - Liberty BASIC Newsletter Issue #133.
'            stylebitsDemo6.bas - Liberty BASIC Newsletter Issue #134.
```

By grouping comments into smaller related chunks and labeling them, the labels can be used as an index to quickly find the information you need. The use of white space makes this information easier to read and access.

In John M. Nevinson's book *The Little Book of BASIC Style*, How to write a program you can read, rule number 3 states, "Distinguish comment from code." One of the ways to distinguish comment from code is with white space. Another way to do it, with style, is to put a border around them. COBOL (a computer language designed for business applications) programmers would sometimes surround their comments with boxes made up of asterisks called flower boxes.

```
***********************************
*                                 *
*  Program:  Qubic.               *
*                                 *
*  Description:  3D Tic-Tac-Toe.  *
*                                 *
***********************************
```

I use flower boxes around the opening comments that introduce my programs. It really makes them stand out and is a great way to enhance their look. In the comments for my subroutines, functions, and gosubs, I use a comment line for my top and bottom border instead of making a box. Because I change these comments often when writing a program, the absence of side borders makes entering and changing them easier.

```
Function File_Exists(Path$, FileName$)
' ====================================================================
=
'
'   Purpose:  Use the File command to check for a file's existence.
'
'   Parms:
'       FileName$ - Name of the file to search for.
'       Path$     - Directory path to search for the file in.
'
'   Returns:
'       0 - File not found.
'       1 - File exists.
'
' ====================================================================
=
```

As you can see, using white space and borders makes the comments stand out. They are easier to read, easier to follow, and easier on your eyes.

Comment boxes can be used to section off and label parts of a program. These parts are then easier to find when looking for them on a computer screen or in a printed listing.

```
+------------------+   (!) (!) (!) (!) (!) (!)   <?> <?> <?> <?>
<?> <?>
|                  |
                            <?>                 <?>
|  INITIALIZATION  |   (!)  MAIN  PROGRAM  (!)   <?>  ERROR
 ROUTINE  <?>
|                  |
                            <?>                 <?>
+------------------+   (!) (!) (!) (!) (!) (!)   <?> <?> <?> <?>
<?> <?>

' **************
' * INITIALIZE *
' *****************************************************************

'                 +-------------+
'                 | MAIN PROGRAM |
' ---------------+-------------+-------------------------------

'                                    .============.
'                                    |  FUNCTIONS  |
' ================================================================

' # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # #
#
' #
#
' #                       S U B R O U T I N E S
#
' #
#
' # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # #
#
```

Try creating your boxes with different line styles, like the ones below, or make up your own. Your imagination (and the font you use) is your only limitation.

```
/*\*/*\*/*\    <*><*><*><*><*>    :-) :-) :-) :-) :-)
```

Comments can be used as line headings to sub divide your program into smaller pieces. They can be used to group and identify related items.

```
1) Windows and their controls.

'               -------- WINDOWS ---------

'         ========== Adventure Window ==========
StaticText #Adv.ST1, "A D V E N T U R E", 220, 20, 120, 20

'        ========== Command Entry Textbox ==========
TextboxColor$ = "White"
TextBox #Adv.tbCmd, 20, 560, 925, 45


'          #-#-#-#-#- About This Program Window -#-#-#-#-#
StaticText #Abt.ST1, "About This Program", 220, 20, 120, 20

'           #-#-#-#-#- Close About Window Button -#-#-#-#-#
Button #Abt.btnClose, "Close", [Close_ATP], UL, 670, 385, 90, 35


2) Variable groups.
```

```
'              ------- CONSTANTS --------

'            ----- Special Characters -----
CR$ = Chr$(13)    ' Carriage Return
DQ$ = Chr$(34)    ' Double Quote Mark (")

'                ----- File Status -----
Open   = 0        ' File Open
Closed = 1        ' File Closed

'             ----- Boolean Constants -----
True  = 1
False = 0

'                   -------- TABLES --------
Dim Conversations$(25)    ' Conversations of passengers.
Dim ConvNo(25)            ' Number of conversations per trip segment.


'                -------- VARIABLES --------

'                 ----- Flags -----
Bear = 0    ' 0-Ferocious  1-Friendly  2-Freed
Bird = 0    ' 0-Loose      1-In Cage

' ---------- Enterprise ----------
Shields      = 0
Photon_Torps = 10
Energy       = 1000
```

3) Data being read in.

```
'              ---------- Read In Treasure Locations ----------
For I = 1 to TotTreas
    Read N
    TreasLoc(I) = N
Next

Data 18, 25, 23, 24, 21, 52, 00, 71, 74, 58, 59, 69, 66, 82, 10
, 07, 07
```

These are some of the ways you can use line comments to make your code more understandable. Try creating them with different eye catching line styles.

```
+-+-+-+-+-+    /-/-/-/-/-/    =0= =0= =0= =0=    [#]-[#]-[#]-[#]
```

Last, but not least, enliven your comments with a little ASCII art. ASCII is a numeric code used by PC's to identify the characters you type in. ASCII art is pictures created with characters typed in from the keyboard. If you are artistically challenged, like me, Google ASCII Art to find web sites where you can get ideas for pictures to include in your comments. Google is a search engine program used to find information on the Internet. You can use it or any other search engine program to find ASCII Art sites. Here is a Star Trek example:

```
' ********************************************************************
*
' *
*
' *
*
' *       #####  #####  ###  ####       ##### ####   ##### #    #
*
' *     #         #   #   # #   #        #   #   #  #     #   #
*
' *      ###     #   ##### ####         #   ####   ####  ####
*
' *         #   #   #   # #   #        #   #   #  #     #   #
*
' *    #####   #   #   # #    #       #   #    # ##### #     #
*
' *
*
' *             ___________________            _-_
*
' *             \==============_=_/   ____.---'---`---.____
*
' *                          \_\      \----._________.----/
*
' *                           \ \     /  /    `-_-'
*
' *                      __,---`.`---'..'-_
*
' *                     /_                _|(
*
' *                       `-._____________/
*
' *
```

```
*
' *
*
' * - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
*
' *
*
' *  Program:  Star Trek v1.0
*
' *
*
' *  Description:  Save the Federation from invading Klingons.
*
' *
*
' *  Hardware:
*
' *     Computer:          IBM Compatible PC.
*
' *     Hard Disk Space:  128 K
*
' *     Memory:            512 MB
*
' *
*
' *  Software:
*
' *     Liberty Basic 4.3
*
' *     Microsoft Windows XP
*
' *
*
' *  Reference:
*
' *     Creative Computing 101 Basic Computer Games
*
' *
*
' *  Reference:
*
' *     Windows is a registered trademark of Microsoft Corp.
*
' *     Liberty Basic is a registered trademark of Shop Talk.
*
' *
```

```
*
' ********************************************************************
*
```

You can use icons, needlepoint designs, art books, or anything else as inspiration for pictures to include in your comments.

```
     Open        Unlock     Lock          Torpedo

                  .***.
+---+    +---+    *   *     .***.    **\   /=======\
|   |\   |   |    *         *   *    *  ***         \
|   | |  |   |   *******   *******   *  ***         /
+---+ |  +---+   *     *   *     *   **/   \=======/
     \|          *     *   *     *
      \          *******   *******
```

As you can see, you don't have to have an elaborate picture. The simple outline of an object can convey your message and make your comments more interesting.

Commenting your code does not have to be a boring but necessary evil. You can have fun by using your creativity to make comments stand out with style. Don't go overboard by putting too many items in your comments. Sometimes simplicity and restraint have a beauty and style all of their own. Now when someone looks at your program listing, they can be impressed with your code, marvel at your comments, and appreciate your imagination and creativity.