

Line editor for interactive editing of text lines/pages - LINEEDIT.BAS

[lineedit.bas.v2.zip](#) [lineedit.bas.v3.zip](#) [lineedit.bas.zip](#)

The LineEdit function defined in the attached LINEEDIT.BAS is an MMBasic 4.5 function to edit lines of text or even pages of text, as these five examples demonstrate:

```
'  
' Edit a string until ENTER with LineEdit:  
'
```

```
Sub Example1  
  a$ = "text to edit"  
  Do: k = LineEdit(a$): Loop Until k = 13 ' until ENTER  
End Sub
```

```
'  
' Edit a string until ENTER or ESC with ESC to restore the old string value:  
'
```

```
Sub Example2  
  a$ = "text to edit"  
  b$ = a$ ' save old string value  
  Do: k = LineEdit(a$): Loop Until k = 13 Or k = 27 ' until ENTER or ESC  
  If k = 27 Then ' ESC pressed  
    Print b$;  
    If Len(a$) > Len(b$) Then Print Space$(Len(a$) - Len(b$));  
    a$ = b$ ' restore old value on ESC  
  EndIf  
End Sub
```

```
'  
' Interactive editing of 10 lines with up/down screen navigation:  
'
```

```
Sub Example3  
  Local a$(10) length 40  
  Cls  
  Print "Tiny edit | ESC to quit"  
  For row = 1 To 10  
    Print @(0,12*row) a$(row);Tab(11);"|"
```

```
  Next row  
  row = 1 ' first row to edit  
  cur = 0 ' cursor at begin of the line  
  max = 40 ' max size of a$()  
  win = 10 ' edit in window of 10 chars  
  Do  
    Print @(0,12*row); ' place cursor where we want to edit the line  
    k = LineEdit(a$(row), cur, -1, max, win) ' -1: get next key press  
    If k = 128 Then  
      If row > 1 Then row = row - 1 ' UP key: move up  
    ElseIf k = 13 Or k = 129 Then  
      If row < 10 Then row = row + 1 ' DOWN key: move down
```

```
EndIf
Print @(0,132) "row: ";row;" col: ";cur;" key: ";k;" ";
Loop Until k = 27 ' until ESC
End Sub
```

```
'
' Interactive editing with screen navigation while continuously updating the
screen with the current time:
'
```

```
Sub Example4
Local a$(10) length 120
Cls
Print "          | Editor | ESC to quit"
For row = 1 To 10
  Print @(0,12*row) a$(row);
Next row
row = 1 ' first row to edit
cur = 0 ' cursor at begin of the line
max = 120 ' max size of a$()
win = MM.HRes/6 ' edit in window up to window width
' show cursor
Font 1,,1: Print @(0,12*row) Mid$(a$(row)+" ", 1, 1);Chr$(8);: Font 1,,0
Do
  Print @(0,0) Time$; ' continuously show the current time
  k = Asc(Inkey$)
  If k Then
    Print @(0,12*row);
    k = LineEdit(a$(row), cur, k, max, win) ' pass in key to edit line
    If k = 128 Then ' UP key: move up
      If row > 1 Then row = row - 1
      Font 1,,1: Print @(0,12*row) Mid$(a$(row)+" ", 1, 1);Chr$(8);: Font
1,,0
      cur = 0
    ElseIf k = 13 Or k = 129 Then ' DOWN key: move down
      If row < 10 Then row = row + 1
      Font 1,,1: Print @(0,12*row) Mid$(a$(row)+" ", 1, 1);Chr$(8);: Font
1,,0
      cur = 0
    EndIf
  EndIf
  EndIf
  Loop Until k = 27 ' until ESC
End Sub
```

```
'
' Edit a string of digits, masking key presses to digits
'
Sub Example5
a$ = "1234"
Font 1,,1: Print Mid$(a$, 1, 1);: Font 1,,0: Print Mid$(a$, 2);Chr$(13);
Do
```

```
k = Asc(Inkey$)
If k Then
  If k < 32 Or k > 126 Or Instr("0123456789", Chr$(k)) Then
    k = LineEdit(a$, cur, k)
  EndIf
EndIf
Loop Until k = 13 Or k = 27 ' until ENTER or ESC
End Sub
```

Source code of LineEdit:

```
' LineEdit(a$, [cur [, key [, max [, win ]]]) edit string a$ at position
cur,
' use key parameter (when provided) to make the next edit of a$ and update
cur, and
' return new key press, or if key press not provided (zero) keep editing
until
' ESC, ENTER or special key such as F1, UP/DOWN etc. where max is the
maximum
' string length and win is the "window" size. Use LEFT/RIGHT/HOME/END to
' navigate the line, BACKSPACE/DEL to delete a character.
'
' Supports three editing modes: key=0 (default) edit until ENTER or other
' special key, key=-1 wait for key and make one edit, returning the key
value,
' key=X to pass a key press to LineEdit to make one edit, returning the key
' value.
'
' Cursor position cur parameter is passed in and updated by LineEdit
'
' Use the max parameter when the string has a maximum length smaller than
255
'
' To edit within a smaller window, use the win parameter
'
```

```
Function LineEdit(a$, cur, key, max, win)
  Local x, y, size, from, done
  x = MM.HPos: y = MM.VPos
  If max <= 0 Then max = 255
  If win <= 0 Then win = 255
  size = Len(a$)
  If cur < 0 Then
    cur = 0
  ElseIf cur > size Then
    cur = size
  EndIf
  done = key
  Do
    If key >= 32 And key < 127 Then
      If size < max Then
        a$ = Left$(a$, cur) + Chr$(key) + Mid$(a$, cur + 1)
```

```
        cur = cur + 1
        size = size + 1
    EndIf
ElseIf key = 8 Then
    If cur > 0 Then
        cur = cur - 1
        a$ = Left$(a$, cur) + Mid$(a$, cur + 2)
        size = size - 1
    EndIf
ElseIf key = 127 Then
    If cur < size Then
        a$ = Left$(a$, cur) + Mid$(a$, cur + 2)
        size = size - 1
    EndIf
ElseIf key = 130 Then
    If cur > 0 Then cur = cur - 1
ElseIf key = 131 Then
    If cur < size Then cur = cur + 1
ElseIf key = 134 Then
    cur = 0
ElseIf key = 135 Then
    cur = size
ElseIf key > 0 Then
    ' done editing, so print a clean line
    Print @(x,y) Left$(a$, win);
    ' remove the cursor from the end of the line
    If cur = size And size < win Then Print " ";
    Exit
EndIf
' cursor cannot move beyond string max length
If cur >= max Then cur = max - 1
' cursor is offset from window start
If cur >= win Then from = cur - win + 2 Else from = 1
If cur = size Then
    ' print the line constrained to the window size
    Print @(x,y) Mid$(a$, from);
    ' put the cursor at the end
    Font 1,,1: Print " ";: Font 1,,0
    ' BACKSPACE may require removing the cursor from the end of the line
    If key = 8 Then
        If cur + 1 < win And size < win Then Print " ";
    EndIf
Else
    ' print the line constrained to window size
    Print @(x,y) Mid$(a$, from, cur - from + 1);
    ' invert character at the cursor
    Font 1,,1: Print Mid$(a$, cur + 1, 1);: Font 1,,0
    ' print the rest of the line
    Print Mid$(a$, cur + 2, win - cur + from - 2);
    If cur < win And size < win Then Print " ";
```

```
EndIf
' if this is a non-blocking call then exit
If done > 0 Then Exit
' wait on key press
Do: key = Asc(Inkey$): Loop Until key
done = -done
Loop
' move cursor back to the start of the line before returning
Print @(x,y);
LineEdit = key
End Function
```

From:

<http://fruitoftheshed.com/wiki/> - FotS

Permanent link:

http://fruitoftheshed.com/wiki/doku.php?id=mmbasic:line_editor_for_interactive_editing_of_text_lines_pages_lineedit_bas

Last update: **2024/01/19 09:30**

