



# programandala.net

English | [Esperanto-versio](#) | [Versión en español](#) | [Chef página in Interlingue](#)



[Programs](#) | [Computers](#) | [Texts](#) | [Pictures](#) | [Contact](#) | [About](#) | [Twitter](#) | [GitHub](#) | [New](#)

[Previous: z80dasm2tidySamForth](#)

[Up: Programs](#)

## ZX BASIC Vim syntax file

Vim syntax file for ZX BASIC.

Tags: [BASIC](#), [Vim](#), [Vim syntax file](#), [ZX BASIC](#), [ZX Spectrum](#).

I wrote the following [Vim](#) syntax file in order to write [ZX BASIC](#) code more comfortably with my favourite editor. I used the [FreeBASIC](#) syntax file already included as a base, because both languages are similar.

### HOMAN TRANSPORT KFT.

B kategóriás  
gépjárművezető

Regionális

Érdekel

The file is not finished yet, but most of the common keywords and operators are well highlighted. When it will be ready, I will submit it for a future release of [Vim](#).

## Source code

► [Gestión anuncios](#)

[Basic variables](#)

[Aplicaciones para diseñar](#)

[Bits bytes](#)

```
" zxbasic.vim
" Vim syntax file
" Language: ZX BASIC (http://www.zxbasic.net) for the ZX Spectrum computer
" Author:   Marcos Cruz (programandala.net)
" License:  Vim license (GPL compatible)
" URL:      http://programandala.net/en.program.zx_basic_vim_syntax_file.html
" Updated:  2015-02-26

" Note:

" This is a work in progress.

" -----
" History:

" 2010-04-18: First draft, based on the FreeBASIC version mantained by Mark
" Manning.
" 2010-07-06: Fixed: the aspect of the zxbasicLoops label was undefined.
```

```
" 2014-04-17: Code reorganized and improved; several issues are solved.
" 2014-04-26: New: 'elseif'.
" 2014-05-01: New: 'then'; unused methods are removed.
" 2014-05-02: New: 'float', 'ubound', 'lbound', 'band', 'bor', 'bxor', 'bnot'.
" Less methods.
" 2014-05-03: Fix: 'endif' removed. New: 'once'.
" 2014-05-05: New: '#pragma'.
" 2014-05-05: Fix: the String group contained the Todo group.
" 2014-05-05: Fix: single-quote strings are removed.
" 2014-05-09: New: Z80 source is highlighted between 'asm' and 'end asm';
" not perfectly working yet: the end of the region is not detected.
" 2014-05-10: New: '#require'.
" 2014-05-12: New: '#line'.
" 2015-02-26: Vim license.
```

```
" -----
" To-do
```

```
" Fix the block comments.
```

```
" -----
```

```
if exists("b:current_syntax")
    finish
endif
```

```
setlocal iskeyword=48-57,65-90,97-122,_,$,%,#
```

```
syn case ignore
```

```
syn keyword zxbasicConsole at
syn keyword zxbasicConsole bold
syn keyword zxbasicConsole border
syn keyword zxbasicConsole bright
syn keyword zxbasicConsole cls
syn keyword zxbasicConsole flash
syn keyword zxbasicConsole ink
syn keyword zxbasicConsole inverse
syn keyword zxbasicConsole italic
syn keyword zxbasicConsole over
syn keyword zxbasicConsole paper
syn keyword zxbasicConsole tab
syn keyword zxbasicFunction abs
syn keyword zxbasicFunction acs
syn keyword zxbasicFunction asn
syn keyword zxbasicFunction atn
syn keyword zxbasicFunction cast
syn keyword zxbasicFunction chr
syn keyword zxbasicFunction chr$
syn keyword zxbasicFunction cos
syn keyword zxbasicFunction exp
syn keyword zxbasicFunction in
syn keyword zxbasicFunction inkey
syn keyword zxbasicFunction inkey$
syn keyword zxbasicFunction int
syn keyword zxbasicFunction lbound
syn keyword zxbasicFunction len
syn keyword zxbasicFunction ln
syn keyword zxbasicFunction peek
syn keyword zxbasicFunction pi
syn keyword zxbasicFunction pos
syn keyword zxbasicFunction rnd
syn keyword zxbasicFunction screen
syn keyword zxbasicFunction screen$
syn keyword zxbasicFunction sgn
syn keyword zxbasicFunction sin
syn keyword zxbasicFunction sqr
syn keyword zxbasicFunction str
syn keyword zxbasicFunction str$
syn keyword zxbasicFunction tan
syn keyword zxbasicFunction ubound
syn keyword zxbasicFunction val
syn keyword zxbasicOperator <<
syn keyword zxbasicOperator >>
syn keyword zxbasicOperator and
syn keyword zxbasicOperator band
syn keyword zxbasicOperator bnot
syn keyword zxbasicOperator bor
```

```
syn keyword zxbasicOperator bxor
syn keyword zxbasicOperator mod
syn keyword zxbasicOperator not
syn keyword zxbasicOperator or
syn keyword zxbasicOperator shl
syn keyword zxbasicOperator shr
syn keyword zxbasicOperator xor
syn keyword zxbasicPreProcessor #define
syn keyword zxbasicPreProcessor #else
syn keyword zxbasicPreProcessor #elseif
syn keyword zxbasicPreProcessor #endif
syn keyword zxbasicPreProcessor #if
syn keyword zxbasicPreProcessor #ifdef
syn keyword zxbasicPreProcessor #ifndef
syn keyword zxbasicPreProcessor #includ
syn keyword zxbasicPreProcessor #include
syn keyword zxbasicPreProcessor #line
syn keyword zxbasicPreProcessor #pragma
syn keyword zxbasicPreProcessor #require
syn keyword zxbasicPreProcessor #undef
syn keyword zxbasicPreProcessor $include
syn keyword zxbasicPreProcessor once
syn keyword zxbasicProgramFlow continue
syn keyword zxbasicProgramFlow do
syn keyword zxbasicProgramFlow else
syn keyword zxbasicProgramFlow elseif
syn keyword zxbasicProgramFlow end
syn keyword zxbasicProgramFlow exit
syn keyword zxbasicProgramFlow fastcall
syn keyword zxbasicProgramFlow for
syn keyword zxbasicProgramFlow function
syn keyword zxbasicProgramFlow gosub
syn keyword zxbasicProgramFlow goto
syn keyword zxbasicProgramFlow if
syn keyword zxbasicProgramFlow loop
syn keyword zxbasicProgramFlow next
syn keyword zxbasicProgramFlow return
syn keyword zxbasicProgramFlow stdcall
syn keyword zxbasicProgramFlow step
syn keyword zxbasicProgramFlow stop
syn keyword zxbasicProgramFlow sub
syn keyword zxbasicProgramFlow then
syn keyword zxbasicProgramFlow to
syn keyword zxbasicProgramFlow until
syn keyword zxbasicProgramFlow wend
syn keyword zxbasicProgramFlow while
syn keyword zxbasicStatement beep
syn keyword zxbasicStatement circle
syn keyword zxbasicStatement clear
syn keyword zxbasicStatement code
syn keyword zxbasicStatement draw
syn keyword zxbasicStatement input
syn keyword zxbasicStatement let
syn keyword zxbasicStatement load
syn keyword zxbasicStatement out
syn keyword zxbasicStatement pause
syn keyword zxbasicStatement plot
syn keyword zxbasicStatement point
syn keyword zxbasicStatement poke
syn keyword zxbasicStatement print
syn keyword zxbasicStatement randomize
syn keyword zxbasicStatement save
syn keyword zxbasicStatement verify
syn keyword zxbasicType as
syn keyword zxbasicType byref
syn keyword zxbasicType byte
syn keyword zxbasicType byval
syn keyword zxbasicType const
syn keyword zxbasicType dim
syn keyword zxbasicType float
syn keyword zxbasicType integer
syn keyword zxbasicType long
syn keyword zxbasicType string
syn keyword zxbasicType ubyte
syn keyword zxbasicType uinteger
syn keyword zxbasicType ulong
```

" XXX TODO finish

```

"syn keyword zxbasicPredefined __DATE__ __FB_DOS__ __FB_LINUX__ __FB_MAIN__
__FB_MIN_VERSION__

" XXX TODO how to do "go to" and "go sub"?:
syn match zxbasicProgramFlow "<\s+(\GO TO|GO SUB)\s+\>"

" Do the Basic variables names first. This is because it
" is the most inclusive of the tests. Later on we change
" this so the identifiers are split up into the various
" types of identifiers like functions, basic commands and
" such. MEM 9/9/2006

"syn match zxbasicIdentifier "<[a-zA-Z][a-zA-Z0-9]*$\>"
"syn match zxbasicGenericFunction "<[a-zA-Z][a-zA-Z0-9]*\>\s*(\"me=e-1,he=e-1

" Catch errors caused by wrong parenthesis

" XXX TODO
"syn region zxbasicParen transparent start='(' end=')'
contains=ALLBUT,@zxbasicParenGroup
"syn match zxbasicParenError ")"
"syn cluster zxbasicParenGroup contains=zxbasicParenError,zxbasicTodo

" Integer number, or floating point number without a dot and with "f".

" XXX TODO
"syn region zxbasicHex start="\$" end="\W"
"syn region zxbasicHexError start="\$\\x*[g-zA-Z]" end="\W"
"syn match zxbasicInteger "<\d+\(u\=l\=\|lu\|f\)\>"

" Floating point number, with dot, optional exponent

" XXX TODO
"syn match zxbasicFloat "<\d\+\\.d*\(e[-+]\=\d\+\)\=[f1]\=\>"

" Floating point number, starting with a dot, optional exponent

" XXX TODO
"syn match zxbasicFloat "\.d\+\(e[-+]\=\d\+\)\=[f1]\=\>"

" Floating point number, without dot, with exponent

" XXX TODO
"syn match zxbasicFloat "<\d\+e[-+]\=\d\+[f1]\=\>"

" Comments

" XXX TODO finish
syn keyword zxbasicTodo contained
syn keyword zxbasicTodo XXX
syn keyword zxbasicTodo TODO
syn keyword zxbasicTodo FIXME
syn keyword zxbasicTodo CHECKME
syn keyword zxbasicTodo OLD
syn keyword zxbasicTodo TMP
syn keyword zxbasicTodo INFORMER

syn region zxbasicLineComment start="^rem" end="$" contains=zxbasicTodo
syn region zxbasicLineComment start=":\$rem" end="$" contains=zxbasicTodo
"syn region zxbasicLineComment start="^'" end="$" contains=zxbasicTodo
"syn region zxbasicLineComment start="\$'" end="$" contains=zxbasicTodo
syn region zxbasicLineComment start="'" end="$" contains=zxbasicTodo,

" 2014-05-02 XXX FIXME '/'...'/ block comments don't work, why?
"syn region zxbasicBlockComment start="/" end="/" contains=zxbasicTodo
"syn region zxbasicBlockComment start="/" end="/"
contains=zxbasicTodo,zxbasicLineComment
"syn region zxbasicBlockComment start="\\" end="\\" contains=zxbasicTodo
"syn region zxbasicBlockComment start="\\" end="\\"
contains=zxbasicTodo,zxbasicLineComment

" String and character constants

syn region zxbasicString start="'" end="'"

```

```

" Labels and line numbers

" XXX TODO finish
"syn match zxbasicLabel "\d+\>"
syn match zxbasicLabel "\w+:"
syn region zxbasicLineNumber start="\d" end="\s"

" Create the clusters

syn cluster zxbasicNumber contains=zxbasicHex,zxbasicInteger,zxbasicFloat
syn cluster zxbasicError contains=zxbasicHexError

" Math operators

" XXX TODO finish
syn match zxbasicMathOperator "[\+|-|=|*|/|>|<|%\(|\)|\]]" contains=zxbasicParen

" Include Z80 source
" See ':help syn-include' for info.

"syntax include @Z80Source syntax/z80.vim

" XXX FIXME start and end are highlighted as Z80! and the end is not detected!
" 'hs' and 'he' don't work. "asm", "end asm" and the ZX BASIC code after "end
" asm" are highlighted as Z80!

"syntax region zxbasicZ80 start="\s*asm\s*$"hs=e end="\s*end asm\s*$"he=s
contains=@Z80Source

syn keyword zxbasicProgramFlow asm

" The default methods for highlighting. Can be overridden later

hi def link zxbasicBlockComment Comment
hi def link zxbasicConsole Statement
hi def link zxbasicType Type
hi def link zxbasicError Error
hi def link zxbasicFunction Function
hi def link zxbasicLabel Label
hi def link zxbasicLineComment Comment
hi def link zxbasicLineNumber Label
hi def link zxbasicMathOperator Operator
hi def link zxbasicNumber Number
hi def link zxbasicOperator Operator
hi def link zxbasicPreProcessor PreProc
hi def link zxbasicProgramFlow Special
hi def link zxbasicStatement Statement
hi def link zxbasicString String
hi def link zxbasicTodo Todo

let b:current_syntax = "zxbasic"

" vim: ts=2:sts=2:sw=2:et:smartcase:

```

## Downloads

- [zxbasic.vim \(9.66 KiB\)](#)

**MAGYAR POSTA ZRT.**  
Kiemelt gépjárműve

Regionális

Érdekel

---

[Previous: z80dasm2tidySamForth](#)

[Up: Programs](#)

---

© [Marcos Cruz](#)

URL: [http://programandala.net/en.program.zx\\_basic\\_vim\\_syntax\\_file.html](http://programandala.net/en.program.zx_basic_vim_syntax_file.html)

Document created on 2010-06-17; last modified on 2010-08-11.

Built by [Fendo](#).