⇐ STABLE-INDEX ⇒ tutorial glossar ⇒ essay ⇒ cook book

STABLE - GLOSSARY

an extreme small an fast FORTH-VM

For a more acurate documentation of the newst OP-Codes look at block 9

```
arithmetic
+  ( a b--a+b) addition
-  ( a b--a-b) subtraction
*  ( a b--a*b) multiply
/  ( a b--a/b) division
%  ( a b--a%b) modulo (division reminder)
_  (   n-- -n) negate

bit manipulation
& ( a b--a&b)  32 bits and
|  ( a b--a|b) 32 bits or
~  (  n -- n') not, all bits inversed (0=>1 1=>0)

stack
#  ( a--a a) duplicate top of stack
\  ( a b--a) drop top of stack
$  ( a b--b a) swap top of stack
@  ( a b--a b a) (over) copy next of stack on top

register
x  ( --)  select register x (x: a..z)
;  ( --value) fetch from selected register
:  ( value--) store into selected register
?  ( --value) selected register contains an address. Fetch value from there
!  ( value--) selected register contains an address. Store value there.
```

```
+  ( --) immediately after register, increment content by 1
-  ( --) immediately after register, decrement content by 1
```

functions

```
{X ( --)  start function definition for function X (A..Z)
}  ( --)  end of function definition
X  ( --)  call function X (A..Z)
```

I/O

```
.  ( value--) display value as decimal number on stdout
,  ( value--) send value to terminal, 27 is ESC, 10 is newline, etc.
^  ( --key)   read key from terminal, don't wait for newline.
"  ( --)      read string until next " put it on stdout
0..9 ( --value) scan decimal number until non digit. to push two values
              on stack separete those by space (4711 3333)
              to enter negative numbers call _ (negate) after the number
0..9.0 ( --value) to enter float numbers digits must contain one . (only
              available if float module is active, see 0`)
```

conditions

```
< ( a b--f)  true (-1) if b is < a, false (0) otherwise
> ( a b--f)  true (-1) if b is > a, false (0) otherwise
= ( a b--f) true (-1) if a is qeal to b, flase (0) otherwise
( ( f--) if f is true, execute content until ), if false
         skip code until )
[ ( f--f) begin while loop only if f is true. keep flag on stack
          if f is false, skip code until ]
] ( f--)  repeat the loop if f is true. If f is false, continue
          with code after ]
```

extensions (expermiental)

```
` ( n--) call extension function n
         0  ... switch to floating point mode
                 + - * / . _ < >

         1  ... switch back to interger mode
         2  ... dbg, function dbg() to set breakpoint for debugging
         3  ... switch traceing on (IP, TOKEN, SP, STACK) (not in stable_fast)
         4  ... switch traceing off. Tracing int file trace
         5  ... < = > without dropping their 2nd operand (NOS). This
                is the behavior of Santors original virtual engine.
         6  ... mstime, time in milliseconds, for timing
         7  ... ( n--) edit block number n
         8  ... ( n--) load block number n. Data segment remains. So this
                               is a kind of shared memory. Registers could be used as arguments.
                                         After leaving the application and 0 is on TOS, STABLE will be
                                         terminated. A value not equal 0 on TOS will load this block.
                                         If the block is not valid, the command block will be loaded
                                         Use block 0 as an index for all your block numbers
    9  ... ( n 9--) copy block n (1000 cells) into memory begining of 1000.
           write back the old content before. At exit write back current
           data block. STABLE is starting with block 0 loaded.
   10  ... trace only current state (ip, rp, sp, ..) on stdout
   11  ... quit VM ( n--) n is exit code to os
   12  ... ( --n) push current data block number on stack
   13  ... ( --)  copy 1000 cells from address 1000 to 2000
   14  ... ( --)  copy 1000 cells from address 2000 to 1000
```