Summary of Forth Style Conventions

This is a Forth Style guide for Forth-83 placed in the public domain by Leo Brodie

Spacing and Indentation Guidelines

* 1 space between the colon and the name
* 2 spaces between the name and the comment *
* 2 spaces, or a carriage return, after the comment and before the definition *
* 3 spaces between the name and definition if no comment is used
* 3 spaces indentation on each subsequent line (or multiples of 3 for nested indentation)
* 1 space between words/numbers with a phrase
* 2 or 3 spaces between phrases
* 1 space between the last word and the semicolon
* 1 space between the semicolon and IMMEDIATE (if invoked)

No blank lines between definitions, except to separate distinct groups of defintions

* An often seen-alternative calls for 1 more space between the name and comment and 3 between the comment and the defintion. A more liberal technique uses 3 spaces before and after teh comment. Whatever you choose, be consistent.

Stack-Comment abbreviations

n	single-length signed number
d	double-length signed number
u	single-length unsigned number
ud	double-length unsigned number
t	triple-length
q	quadruple-length
c	7-bit character value
b	8-bit byte
?	boolean flag; or :
t=	true
f=	false
a or adr	address
acf	address of code field
apf	address of parameter field
'	(as prefix) address of
s d	(as a pair) source destination
lo hi	lower-limit upper-limit (inclusive)
#	count
o	offset
i	index
m	mask
x	don't care (data structure notation)

An "offset" is a difference expressed in absolute units, such as bytes. An "index" is a difference expressed in logical units, such as elements or records.

Input-Stream Comment Designations

c	single character, blank delimited
name	sequence of characters, blank delimited
text	sequence of characters, delimited by nonblank

Follow "text" with the actual delimiter required, e.g., text" or text).

Samples of Good Commenting Style

Here are two sample screens to illustrate good commenting style.

```
Screen #126
 0 \ Formatter          Data Structures -- p.2    06/06/83
 1  6 CONSTANT TMARGIN  \ line# where body of text begins
 2 55 CONSTANT BMARGIN  \ line# where body of text ends
 3
 4 CREATE HEADER  82 ALLOT
 5   ( 1left-cnt | 1right-cnt | 80header )
 6 CREATE FOOTER  82 ALLOT
 7   ( 1left-cnt | 1right-cnt | 80footer )
 8
 9 VARIABLE ACROSS   \ formatter's current horizontal position
10 VARIABLE DOWNWARD \ formatter's current vertical position
11 VARIABLE LEFT     \ current primary left margin
12 VARIABLE WALL     \ current primary right margin
13 VARIABLE WALL-WAS \ WALL when curr. line started being formatted
14
15
```

```
Screen #127
 0 \ Formatter          positioning -- p.1        06/06/83
 1 : SKIP  ( n -- )  ACROSS +! ;
 2 : NEWLEFT      \ reset left margin
 3    LEFT @  PERMANENT @ + TEMPORARY @ + ACROSS ! ;
 4 : \LINE        \ begin new line
 5    DOOR CR'   1 DOWNWARD +!  NEWLEFT  WALL @ WALL-WAS ! ;
 6 : AT-TO{? ( -- t=at-top )   TMARGIN DOWNWARD @ = ;
 7 : >TMARGIN     \ move from crease to TMARGIN
 8      0 DOWNWARD !  BEGIN \LINE  AT-TOP? UNTIL ;
 9
10
11
12
13
14
15
```

Naming Conventions

Meaning	Form	Example
Arithmetic		
integer 1	1name	1+
integer 2	2name	2*
takes relative input parameters	+name	+DRAW
takes scaled input parameters	*name	*DRAW
Compilation		
start of "high level" code	name:	CASE:
end of "high level" code	;name	;CODE
put something into dictionary	name,	C,
executes at compile time	[name]	[COMPILE]
(slightly different)	name' (prime)	CR'
internal form or primitive	(name)	(TYPE)
	or <name>	<TYPE>
compiling word run-time part:		
systems with no folding	lower-case	if
systems with folding	(NAME)	(IF)

defining word	:name	:COLOR
block-number where overlay begins	namING	DISKING

Data Structures

table or array	names	EMPLOYEES
total number of elements	#name	#EMPLOYEES
current item number (variable)	name#	EMPLOYEE#
sets current item	(n) name	13 EMPLOYEE
advance to next element	+name	+EMPLOYEE
size of offset to item from beginning of structure	name+	DATE+
size of (bytes per) (short for BYTES/name)	/name	/EMPLOYEE
index pointer	>name	>IN
convert address of structure to address of item	>name	>BODY
file index	(name)	(PEOPLE)
file pointer	-name	-JOB
initialize structure	0name	0RECORD

Note: The Forth Scientific Library is using a different style for arrays and structures.

Direction, Conversion

backwards	name<	SLIDE<
forwards	name<	CMOVE<
from	<name	<TAPE
to	>name	>TAPE
convert to	name>name	FEET>METERS
downward	\name	\LINE
upward	/name	/LINE
open	{name	{FILE
close	}name	FILE}

Logic, Control

return boolean value	name?	SHORT?
returns reversed boolean value	-name?	-SHORT?
address of boolean	'name?	'SHORT?
operates conditionally	?name	?DUP (maybe DUP)
enable	+name	+CLOCK
or, absence of symbol	name	BLINKING
disable	-name	-CLOCK -BLINKING

Memory

save value of (to stack)	@name	@CURSOR
restore value of	!name	!CURSOR
store into	name!	SECONDS!
fetch from	name@	INDEX@
name of buffer	:name	:INSERT
address of name	'name	'S
address of pointer to name	'name	'TYPE
exchange, especially bytes	>name<	>MOVE<

Numeric Types

byte length	Cname	C@
2 cell size, 2's complement integer encoding	Dname	D@
mixed 16 and 32-bit operator	Mname	M*
3 cell size	Tname	T*
4 cell size	Qname	Q*
unsigned encoding	Uname	U.

Output, Printing

print item	.name	.S
print numeric (name denotes type)	name.	D. U.
print right justified	name.R	U.R

Quantity

"per"	/name	/SIDE

Sequencing

start	<name	<#
end	name>	#>

Text

string follows delimited by "	name"	ABORT" text"
text or string operator (similar to $ prefix in BASIC)	"name	"COMPARE
superstring array	"name"	"COLORS"

How to pronounce the Symbols

!	store
@	fetch
#	sharp (or "number" as in #RECORDS)
$	dollar
%	percent
^	caret
&	ampersand
*	star
(	left paren; paren
)	right paren; paren
-	dash; not
+	plus
=	equals
{ }	faces (traditionally called "curly brackets")
[ ]	square brackets
"	quote
'	as prefix: tick; as suffix: prime
~	tilde
|	bar
\	backslash (also "under", "down" and "skip")
/	slash (also "up")
<	less than left dart
>	greater than right dart
?	question (some prefer "query")
,	comma
.	dot

Source:

Brodie, Leo, 1984; Thinking Forth, A Language and a Philosophy for Solving Problems, Prentice-Hall, Englewood Cliffs, N.J., 300 pages. ISBN 0-13-917568-7

This book has been out of print, and is now being republished by the Forth Interest Group.