

# Input / Output in Volks-FORTH#

## Table of Contents

- [Input / Output in Volks-FORTH](#)
- [Input / output commands in VolksFORTH](#)
- [Input / output terminal on](#)
- [Input / output of numbers](#)
- [Input / output via a port](#)
- [Enter characters](#)

## Input / output commands in VolksFORTH#

All input and output words (**KEY EXPECT EMIT TYPE** etc.) are vectorized in the people-FORTH, ie when called, the code field address of the associated command is taken from a table and executed. It is included in the system table wine named DISPLAY, which provides for the output to the screen terminal.

This method of vectorization offers significant benefits:

- Nit-vectorization of the input can e.g. switch with one stroke of the keyboard on the input of a modem.
- The output can vectorization with a new table, the total expenditure to another device (eg a printer) will be conducted without having to change the output commands themselves.
- In a word (**DISPLAY, PRINT**) can all be changed spending habits. Is there such a (((print a list display))) will be issued a screen on a printer, and then falls back to the screen. So you need a new word, such PRINTERLIST to define.

A new table is created with the word **OUTPUT:** . The definition can output with (((view:)) look).

**OUTPUT:** Expected to issue a list of words, with must, be completed.

Beipsiel:

```
Output:> PRINTER
  pemit pcr ptype pdel PPAG pat pat? ;
```

For a new table named > **PRINTER** is created. With a later call to PRINTER > is the address of this table in the Uservariable **OUTPUT** written. From now leads **EMIT** from a **PEMIT**, a **PTYPE TYPE** etc.

The order of words after **OUTPUT:** userEMIT userCR userType userdel Userpage userAT userAT? must necessarily be met. Accordingly, the input-vectorization is handled.

## Input / output terminal on#

The state-FORTH has a number of constants which serve to improve readability:

- [C / row](#)
- [C / col](#)
- [C / dis](#)
- [C / l](#)
- [L / s](#)
- [Bl](#)
- [# Esc](#)
- [# Cr](#)

- [# Lf](#)
- [# Bel](#)
- [# Bs](#)
- [Standardized / o](#)
- [Inputkol](#)
- [Outputkol](#)
- [Area](#)
- [Areakol](#)
- Terminal
- [Window](#)
- [Full](#)
- [Curat?](#)
- [Cur!](#)
- [Setpage](#)
- [@ Video](#)
- [Savevideo](#)
- [Restorevideo](#)
- [Catt](#)
- [List](#)
- [\(Page](#)
- [Page](#)
- [\(Del](#)
- [Del](#)
- [\(Cr](#)
- [Cr](#)
- [? Cr](#)
- [\(At](#)
- [\(At?](#)
- [At](#)
- [At?](#)
- [Col](#)
- [Row](#)
- Curoff
- [Curon](#)
- [Curshape](#)
- [Printer](#)
- [Print](#)
- [+](#)
- [Ls!](#)

### **Input / output of numbers#**

The input of numbers is made in the interpretive mode via the keyboard, and basic input words are defined with **number numbers** and related words. For the issue of numbers again is the lack of typing of FORTH observed - for a specific data format (integer, unsigned, double) is appropriate in each case the operator to select.

- [.](#)
- [U.](#)
- [D](#)
- [.R](#)
- [U.r](#)
- [D.r](#)

## Input / output via a port#

### MS-DOS

- [Pc @](#)
- [PC!](#)

## Enter characters#

In FORTH you will always designate a storage area, incorporated into the characters and strings. To do this you usually use a small, 80-character memory area called **PAD**. This note pad - so the German translation of pad used - no fixed memory area and is both the FORTH system and the programmers.

Then I liked you with the text input buffer **TIB** another important Speicherbereich imagine that ensures the reasonable use of the connected devices. Because the text input via the keyboard vorsichtig relatively slow, the characters are collected here only in a free space, the buffer **TIB**, and then processed.

- [tib](#)
- [#tib](#)
- [>tob](#)
- [>in](#)
- [pad](#)
- [input](#)
- [keyboard](#)
- [empty-keys](#)
- [\(key?](#)
- [key?](#)
- [\(key](#)
- [key](#)
- [\(decode](#)
- [\(expect](#)
- [expect](#)
- [span](#)
- [>expect](#)
- [nullstring?](#)
- [stop?](#)
- [source](#)
- [word](#)
- [parse](#)
- [name](#)
- [find](#)
- [execute](#)
- [perform](#)
- [query](#)
- [interpret](#)
- [output](#)
- [display](#)
- [\(emit](#)
- [emit](#)
- [charout](#)
- [tipp](#)
- [\(type](#)
- [type](#)

- [ltype](#)
- [space](#)
- [spaces](#)