Strings (strings) in state-FORTH#

Here are the basic routines for string processing. Above all, words were added that allow the handling of those required by some operating systems 0-terminated strings. FORTH has here over C has the disadvantage that FORTH-strings start with a default count byte that contains the length of the string. A final mark (eg a null byte) is therefore unnecessary. If the operating system but was written in C (Atari TOS, MS-DOS), have strings are converted accordingly.

By default FORTH uses counted strings, which are only identified by an address. The byte at that address contains the words, how long the string is. On this "byte count" then the string itself follows a result, the Läunge standard strings in FORTH is limited to 255 characters. The shortest string is a string of length NULL, for its review, the command is **NULLSTRING?** Available.

This is what the string FORTH at the address addr in memory at FORTH.

- . "
- "
- , "
- Null string?
- 'Lit
- . (
- (
-) - this is not a Forth word, but a stop sign

String manipulation#

Here in the glossary, the following comment Stack (string -) the address of a counted string, len against (addr -) to characterize the start address of the string and its length.

No string variable? - Use:

```
: String: Create dup, 0 c, DOES> allot 1 + count;
```

- Caps
- Capital
- Upper
- Capitalitze
- / String
- -Trailing
- Scan
- Skip
- "
- Bounds
- Type
- > Type
- Place
- Attach
- Append
- Detract
- Match
- Search

The Dictionary#

- (Find
- Find

0-terminated strings#

There is another form of representation for strings, which is suitable for example for MS-DOS. These strings are indeed also characterized by an address, this address does not include a byte count. Instead, these strings are terminated with a null byte.

- asciz
- >asciz
- counted

Conversions: Strings - Numbers#

String to convert numbers#

- Digit?
- Accumulate
- Convert
- Number?
- Number
- Dpl

FORTH in the input of numbers is often realized with the general text and the commands for converting strings to numbers. In the literature it is often the solution with **QUERY** available:

```
: In # (string - d n tf tf addr ff)
   query bl word number? ;
```

This solution is unfavorable because **QUERY** clears the **TIB**. At the same time, the definition of **NUMBER?** An unhappy place in the people-FORTH dar. It is in Laxen & Perry F83-word the same name, the very different (better deals!), With the parameters. Here is the definition of the F83-NUMBER?, Based on the state-FORTH **NUMBER?** :

```
: F83-NUMBER? (String - d f)
  number? ? IF dup 0 <IF true THEN extend exit THEN
  drop false 0 0;
```

This represents the word **INPUT #** an inexpensive option for entering numbers 16/32Bit-Zahlen:

```
\ Input #
: Input # (string - d f)
  pad c / l 1 -> expect \ get a maximum of 63 char
  F83-pad number? ; \ Convert string-> number
```

So the user can evaluate the given flag and use the double-exact number as he sees fit to make in the simplest case of dropping a Single precision number.

Convert numbers to strings#

- #

- # S
- Hold
- Sign
- #>