

# Ausgabe in Strings

M. Anton Ertl  
TU Wien

# Probleme

- Man braucht einen String
- Viele Wörter geben auf die Standardausgabe aus ohne Pendant, das ein String baut  
Beispiel: F.  
Auch viele von Benutzern geschriebene Wörter
- Strings zusammenbasteln ist oft mühsam  
Mehr Werte auf dem Stack als bei Ausgabe

## Lösung

- `>string-execute ( ... xt -- ... addr u )`

- Beispiel

```
X f@ ['] f. >string-execute type
```

```
X f@ ['] f. >string-execute 2dup type drop free throw
```

# Großes Beispiel

```
: mix2bytes_ind, ( index n k i -- index n ) >r
  >r over r@ 64 +
  <<#
  s" );" holds r> 7 r@ - 0 #s 2drop >r
  s" ))], " holds $7 and 8 * 0 #s 2drop
  s" ]^(0xff&(t>>" holds 0 #s 2drop
  s" ^=ROL(rnds[states[" holds r@ 7 and 0 #s 2drop
  s" a" holds 0. #> \c, #>>
  rdrop rdrop ;
```

```
: .mix2bytes_ind ( index n k i -- index n ) >r
  >r over r@ 64 + swap
  ." a" r@ 7 and 0 .r ." ^=ROL(rnds[states[" 0 .r ." ]^(0xff&(t>>"
  $7 and 8 * 0 .r ." ))], " r> 7 r@ - 0 .r ." );" ;
```

```
: mix2bytes_ind, ( index n k i -- index n )
  ['] .mix2bytes_ind >string-execute 2dup \c, drop free throw ;
```

## Ohne ALLOCATE/FREE

- `>buffer-execute ( ... c-addr u1 xt -- ... u2 )`
- `u2` ist die Laenge der Ausgabe (auch wenn sie zu lang ist)
- Bei Bedarf zweimal laufen lassen:  
Ein Mess/Trainingslauf, Buffer Allozieren, ein Wertungslauf

## Allgemeiner

- In Forth läuft viel über implizite Parameter/Ausgaben  
Z.B. input stream, BASE, Ausgabe
- Das ist oft problematisch: nicht allgemein verwendbar
- Klassische Lösung  
Wörter mit expliziten Parametern/Resultaten  
Machen (leider?) viele nicht  
Zusätzliche Werte auf dem Stack

## Alternativlösung

- Lokalisierbarer Kontext
- Beispiele:
  - : hex. ( u -- ) ['] u. 16 base-execute ;
  - : \$create ( c-addr u -- ) ['] create execute-parsing ;
- Passen implizite Ein/Ausgaben besser zu Forth?
- Kontext-Verwaltung über EXECUTE-ähnliche Wörter  
Sichere Fehlerbehandlung

# Zusammenfassung

- `>string-execute` leitet Ausgabe in String um
- Verbreitert die Anwendungsmöglichkeiten von Ausgabewörtern
- Einfachere Programmierung durch weniger Stack-Items
- Kontext-EXECUTES als allgemeines Schema