

# Forth Starter

Gerald Wodni  
gerald.wodni@gee.at

[forth.wodni.at/files/starter.pdf](http://forth.wodni.at/files/starter.pdf)

13.02.2012



# Crash

```
0 0 !
```

```
here execute
```

```
' catch >body 20 erase abort
```

```
' (quit) >body 20 erase
```

# Roots



Figure: Charles H. Moore



Figure: Elizabeth Rather

National Radio Astronomy Observatory (NRAO)

## Name

- Programming Language for 4<sup>th</sup> generation hardware
- Successor to “compile-link-go”
- FOURTH, IBM filename-resriction (1968) allowed only 5 letters → FORTH

# Name

- Programming Language for 4<sup>th</sup> generation hardware
- Successor to “compile-link-go”
- FOURTH, IBM filename-resriction (1968) allowed only 5 letters → FORTH
- Standards:
  - Forth-77
  - Forth-79
  - Forth-83
  - ANS Forth ('94)
  - *Forth 200x*



# Forth

- Structured
- Imperative
- Reflective
- Concatenative
- Extensible
- Stack-based

# Features

- No distinction between run-time and compile-time
- Macros are computed at compile-time (same language)
- Run-time code generation: dynamic optimizing
- Portability
- Untyped
- Interactive
- Namespace: static binding
- Arbitrary control functions (e.g. begin while repeat)

# Words

- Consist of previously defined words
- Extend the compiler
- Defining words

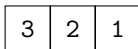


## Reverse Polish notation

Infix	Prefix (PN)	Postfix (RPN)
$1 + 2$	$(+1\ 2)$	$1\ 2+$
$(2 + 3) * 4$	$(*(+2\ 3)4)$	$2\ 3 + 4 *$
$1 * 2 + 3 * 4$	$(+(*1\ 2)(*3\ 4))$	$1\ 2 * 3\ 4 * +$

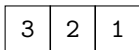
## Manipulation

- `drop ( x - )`

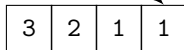
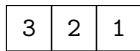


## Manipulation

- `drop ( x - )`



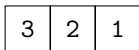
- `dup ( x - x x )`



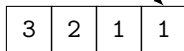
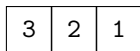


## Manipulation

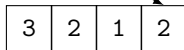
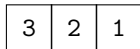
- drop ( x - )



- dup ( x - x x )

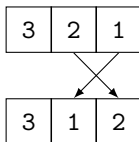


- over ( x1 x2 - x1 x2 x1 )



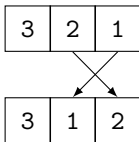
## Manipulation

- swap ( x1 x2 – x2 x1 )

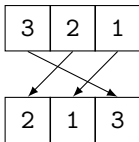


## Manipulation

- $\text{swap} ( x1 \ x2 - x2 \ x1 )$

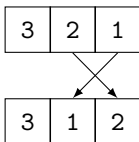


- $\text{rot} ( x1 \ x2 \ x3 - x2 \ x3 \ x1 )$

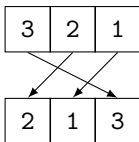


## Manipulation

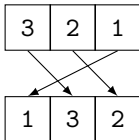
- swap ( x1 x2 – x2 x1 )



- rot ( x1 x2 x3 – x2 x3 x1 )



- -rot ( x1 x2 x3 – x3 x1 x2 )





# Stacks

- Data-stack
- Return-stack
- Floating-point-stack





## Control Flow

```
<flag> if
```

```
...
```

```
else
```

```
...
```

```
then
```

```
<value> case
```

```
  1 of ... endof
```

```
  2 of ... endof
```

```
...
```

```
endcase
```

# Loops

begin

...

again

begin

...

until

begin

...

while

...

repeat



## Counted Loops

```
10 0 do
```

```
...
```

```
loop
```

```
10 0 do
```

```
...
```

```
2 +loop
```



## Some Words

- Datastack-manipulation: dup over drop swap rot
- Returnstack-manipulation: >r r@ r>
- Integer: + - \* / /mod
- Floating-point: f+ f- f\* f/
- Memory: @ ! c@ c! f@ f!



## Links / References

Forth-Gesellschaft e.V. [www.forth-ev.de](http://www.forth-ev.de)

The Forth Net [theforth.net](http://theforth.net)

Anton Ertl [www.complang.tuwien.ac.at/anton/](http://www.complang.tuwien.ac.at/anton/)

Gforth [www.complang.tuwien.ac.at/forth/gforth/Docs-html/](http://www.complang.tuwien.ac.at/forth/gforth/Docs-html/)

Forth 200x [www.forth200x.org](http://www.forth200x.org)

My (old) Projects [forth.wodni.at](http://forth.wodni.at)