

# LINGUAGEM FORTH

FORTH — Linguagem de programação imperativa, introspectiva e baseada em pilha, inventada por “Chuck” Moore em 1970.

Para quem programa na linguagem FORTH, ela segue o modelo de uma pilha, aonde se pode aplicar ações por meio de números e palavras (words).

```
8 1 +  
2 3 *  
/  
.
```

Imprime o valor 1 pois a pilha é de inteiros. O Resultado exato de  $9 \div 6$  é 1,5.

A palavra `.` remove o elemento do topo da pilha e imprime o valor. Para imprimir toda a pilha e sem remover, usa-se o `.s`

Bases numéricas.

- `DECIMAL` Dígitos 0123456789
- `HEX` Dígitos 0123456789ABCDEF
- `OCTAL` Dígitos 01234567

A pilha é uma fila em que o primeiro elemento a entrar é o primeiro a sair dela (First In, First Out – FIFO). Como modificar a pilha?

- `CLEAR` - Limpar a pilha
- `DUP` - Duplicar o topo da pilha (`n -- n n`)
- `DROP` - Remover o topo da pilha (`n --`)
- `SWAP` - Trocar a ordem de dois elementos (`a b -- b a`)

Como definir uma nova palavra? Exemplo: calcular o quadrado de um número inteiro.

---

```
: SQUARE DUP * ;
```

ou poderia rescrever assim:

```
: square ( n -- n*n, calcula o quadrado de n )  
  dup  
  *  
 ;
```

Entrada e Saída.

- `EMIT (char -- )` - Emite o caracter ASCII de número *char*
- `CR ( -- )` - Emite uma quebra de linha
- `SPACE ( -- )` - Emite um espaço em branco

E como se faz pra ler alguma coisa?

- `CHAR <char> ( -- char)` - Poe na pilha o ASCII da palavra que segue.
- `KEY ( -- char)` - Le um caracter poe na pilha o ASCII dele.

Como imprimir na tela uma mensagem?

```
." Ola mundo!"
```

Além da pilha, é possível ter constantes e variáveis.

```
1234 CONSTANT k  
k .  
  
VARIABLE v  
v .  
1234 v !  
v @ .
```

```
: ? @ . ;  
: BINARY 2 BASE ! ;
```

Booleanos.

```
TRUE  
FALSE
```

Operadores relacionais, retornam um booleano na pilha.

```
123 123 =  
123 456 <  
123 456 >  
123 0=  
123 0<  
123 0>  
123 321 = INVERT -- não são iguais
```

Condicional.

```
: ABS (n -- |n|)  
  DUP 0< IF  
    NEGATE  
  THEN  
;  
  
IF ELSE THEN
```

Loops.

```
10 0 DO CR ." Nao computa!" LOOP -- Loop definido  
10 0 DO CR I . SPACE ." nao computa!" LOOP
```

```
BEGIN ... f UNTIL -- loop indefinido  
BEGIN ." Nao computa!" 0 UNTIL -- Loop infinito
```

Exemplo: Função “Hailstone”.

Se o número é par (divisível por 2), então retorna o número dividido por 2. Se o número é ímpar (restou 1 na divisão por 2), então retorna o triplo do número mais 1.

```
: HAILSTONE (n -- hailstone(n))  
  DUP 2 MOD 0= IF  
    2 /  
  ELSE  
    3 * 1 +  
  THEN  
;
```

Referências:

- [Forth Tutorial](#)
- [Starting FORTH](#)
- [Open Firmware Quick Reference](#)
- [\(Conjectura de Collatz\) Por que um problema simples é um dos buracos negros da matemática](#)

© MMXVIII Rudá Moura