

3 "Bench" Programming Language

Monday, September 8, 2025

11:57 AM

Basic Components
of a programming
language.

- literals
- Operators
- Expressions
- Variables
- Assignment.
- Conditionals
- Repetitions
- "functions"

The PUCK 25.3 Programming Language

```
% This is a comment
DEFUN ( main )
  PRINT ( "Hello" , "World" ) .
  x <- 2 + 2 .
  y <- 3 * 12 / 7.5 .
  p <- ( x > 0 ) AND ~ ( y <= 30 ) .
  PRINT ( x * 100 )
END.
```

- keywords are UPPERCASE

- ← assignment

- space separation

- Literals

- operators

AND

~ negation.

- • is a "separator" not a "terminator"

```
% FizzBuzz
DEFUN ( FizzBuzz , n )
  IF ( n MOD 3 = 0 )
    IF ( n MOD 5 # 0 )
      PRINT ( "Fizz" )
    ELSE
      PRINT ( "FizzBuzz" )
    FI
  ELSE
    IF ( n MOD 5 = 0 )
      PRINT ( "Buzz" )
    FI
  FI
FI
END.
```

- function "name"

- MOD Modulus

- = equals

- # not equals.

- IF ... FI

```
% Fibonacci and Loops
```

```
DEFUN ( fibo , n )
```

```
  x <- 1
```

```
  y <- 2
```

```
  z <- 3
```

```
  loop
```

```
  loop
```

```
% Fibonacci and Loops
```

```
DEFUN ( fibo , n )
```

```
  x <- 1 .  y <- 2 .  c <- 3 .
```

```
  LOOP ( c < n )
```

```
    x <- x + y .
```

```
    y <- x - y .
```

```
    c <- c + 1
```

```
  POOL .
```

```
  RET x
```

```
END.
```

• LOOP POOL

```
% Greatest Common Denominator
```

```
DEFUN ( gcd , a , b )
```

```
  LOOP ( a > b )
```

```
    a <- a - b
```

```
  ELIF ( b > a )
```

```
    b <- b - a
```

```
  POOL .
```

```
%  $7(a > b) \wedge 7(b > a) \equiv a = b$ 
```

```
  RET a
```

```
END.
```

• LOOP has an ELIF

• Conditions are tested sequentially until one of them is true.

— the corresponding block is executed

— the next iteration proceeds

LOOP c1

b1

ELIF c2

b2

ELIF c3

b3

==

==

==

POOL.

```
% Newlines and just whitespace
```

```
DEFUN ( foo , s )
```

```
res <- True
```

```
  IF ( ( s = "A" ) OR
```

```
        ( s = "B" ) )
```

```
    PRINT
```

```
    (
```

```
      "Hello" & "World"
```

```
    )
```

```
  FI
```

```
END.
```

• & string concatenation.

a + b

a * b

b + a

—o— EOF