

TI-74 Learn Pascal Quick Reference Card: Elements and Syntax of Pascal

Commands

BREAK
 BYE
 CONTINUE
 DEL
 FORMAT
 LIST
 NEW
 NEW ALL
 NUMBER
 OLD
 RENUMBER
 RUN
 SAVE
 UNBREAK
 VERIFY

Declaration Statements

CONST
 FORWARD
 FUNCTION
 LABEL
 PROCEDURE
 TYPE
 VAR

String Functions

CHR(65)
 CONCAT(A, '&', B)
 COPY('attire', 3, 4)
 DELETE('attire', 1, 2)
 INSERT('at', T, 1)
 LENGTH(C)
 POS('plus', 'surplus')
 STR(8422737, T)

Numeric Functions

ABS(-14)
 ATAN(1)
 COS(3.141592654/2)
 EXP(2.302585093)
 LN(2.718281828)
 LOG(10)
 ODD(21)
 PWROFTEN(1)
 ROUND(89.5)
 SIN(3.141592654)
 SQR(5280)
 SQRT(43560*640)
 TRUNC(89.5)

Character Functions

ORD
 PRED
 SUCC

Operator Precedence

unary minus, unary plus, NOT
 *, /, DIV, MOD, AND
 +, -, OR
 =, <>, <, <=, >, >=, IN

System Values

FALSE
 MAXINT
 MEMAVAIL
 TRUE

Interpreter Options

(*\$w + *)
 (*\$w - *)
 (*\$a + *)
 (*\$a - *)
 (*\$i + *)
 (*\$i - *)

Ordinal Types

ARRAY OF type
 BOOLEAN
 CHAR
 INTEGER
 PACKED ARRAY OF type
 REAL
 STRING
 TEXT

File Types

INPUT
 INTERACTIVE
 KEYBOARD
 OUTPUT

Declarations for Syntax Examples

```
10 PROGRAM PROG(INPUT, OUTPUT);
20 LABEL 200;
30 VAR FILE1, FILE2: TEXT;
40     I, J, N      : INTEGER;
50     X, Y         : REAL;
60     A, B         : BOOLEAN;
70     CH           : CHAR;
80     ST1, ST2    : STRING;
90     AR           : ARRAY[1..50]
                OF INTEGER;
```

CASE

```
100 CASE I OF
110   1:WRITE(' I IS 1');
120   2:WRITE(' I IS 2');
130   3:WRITE(' I IS 3');
140 END; (* OF CASE*)
```

CLOSE

```
100 CLOSE(FILE1);
100 CLOSE(FILE1, LOCK);
100 CLOSE(FILE1, PURGE);
```

EOF

```
100 A:=EOF;
100 IF EOF THEN WRITE('EOF');
```

EOLN

```
100 A:=EOLN;
100 IF EOLN THEN WRITE('EOLN');
```

TI-74 Learn Pascal Quick Reference Card: Syntax of Pascal (Continued)

EXIT

```
100 EXIT(TEST);  
100 EXIT(PROGRAM);
```

FILE

```
100 VAR FILE1: FILE OF CHAR;
```

FILLCHAR

```
100 FILLCHAR(ST1,10,'-');  
100 FILLCHAR(ST1[5],10,CH);
```

FOR DOWNTO DO

```
100 FOR I:=10 DOWNTO 1 DO WRITE(I);
```

FORTO DO

```
100 FOR I:=1 TO 10 DO WRITE(I);
```

FUNCTION

```
100 FUNCTION FACTORIAL(N: INTEGER):  
REAL; (*USES VALUE PARAMETER*)  
100 FUNCTION TEST(VAR CH: CHAR):  
BOOLEAN; (*USES REFERENCE  
PARAMETER*)
```

GOTO

```
100 GOTO 200;  
110 200: WRITE(I);
```

GOTOXY

```
100 GOTOXY(0,0); GOTOXY(30,0);
```

HALT

```
100 HALT;
```

IF THEN ELSE

```
100 IF I=1 THEN WRITE('I=1');  
100 IF A=TRUE THEN WRITE('A IS  
TRUE') ELSE WRITE('A IS  
FALSE');  
100 IF (A=TRUE) OR (B=TRUE)  
THEN WRITE('A AND/OR B IS  
TRUE')  
120 ELSE WRITE('A AND B ARE  
FALSE');
```

MOVELEFT

```
100 MOVELEFT(ST1,ST2[10],5);
```

MOVERIGHT

```
100 MOVERIGHT(ST1[10],ST1[5],10);
```

PAGE

```
100 PAGE(FILE1);
```

READ/READLN

```
100 READ(N);  
100 READ(I,J,X,Y);  
110 READ(FILE1,CH);  
100 READLN(N);  
100 READLN(CH);
```

REPEAT

```
100 I:=0;  
110 REPEAT  
120 I:=I+1;  
130 WRITE(ST1[I])  
140 UNTIL I=10;  
110 REPEAT I:=I+1 UNTIL I=10;
```

RESET

```
100 RESET(FILE1,'1.DAT');
```

REWRITE

```
100 REWRITE(FILE1,'1.DAT');  
110 REWRITE(FILE2,'12');
```

SCAN

```
100 I:=SCAN(10,'A',ST1);  
100 WRITE(SCAN(10,<>CH,ST1));
```

SIZEOF

```
100 I:=SIZEOF(ST1);  
100 WRITE(SIZEOF(ST1));
```

WHILE DO

```
100 I:=1;  
110 WHILE I<=10 DO  
120 BEGIN  
130 WRITE(AR[I]);  
140 I:=I+1  
150 END; (* OF WHILE *)
```

WRITE/Writeln

```
100 WRITE(I:5,X:6:2);  
100 WRITE('N IS EQUAL TO ',N);  
100 Writeln(FILE2,CH);  
100 Writeln('I * J =',I*J);
```