

Comparison of FORTRAN - MATLAB - MAPLE - C syntax

Structure	FORTRAN	MATLAB	MAPLE	C
=====	=====	=====	=====	=====
case sensitive	no	yes	yes	yes
comment line	c or * in 1st column	%	#	/* ... */
comment inline	! comments	% comments	# comments	/* comments */
special	spaces ignored	; suppress output : next statement	: suppress output ; end statement	; next statement
Variables	REAL A,B INTEGER I,J CHAR*5 Label	all variables complex, double precision	types set by assignment i:=2; x:=2.0; Label:='abc';	int, float, double, char static void, struct
Dimensions	DIMENSION A(10) REAL B(10,5) REAL*8 C(-2:5) (any index)	auto dim'd as used; up to two dimensions (index >=1)	A:=array(1..10); B:=array(1..10,1..5); C:=array(-2..5); (any index)	float A[10] float B[10][5] double C[7] (index >=0)
Array Elements	A(3), B(3,2), C(-1)	A(3), B(3,2)	A[3], B[3,2], C[-1]	A[3], B[3][2]
Arithmetic	+ - * / **	+ - * / ^ .* ./ .^ (element-wise)	+ - * / ** (or ^) &* (matrix product)	+ - * / **
Do Loop	DO I = 1,10,2 ENDDO	for i = 1:2:10 ; end	for i from 1 to 10 by 2 do; od;	for (i=1; i<=10; ++2) { ; }
While Loop	DO WHILE (A.EQ.0) ... ENDDO	while a==0 ... ; end	while a=0 do ...; od;	while (a=0) { ... ; }
If-Then-Else	IF (A.EQ.0) THEN ELSEIF(a.GT.2) THEN ELSE ENDIF	if a==0 elseif a>2 else end	if a=0 then; elif a>2 then; else; fi	if (a==0) { } elseif (a>2) then { } else { }
Comparisons	.EQ. .NE. .LT. .LE. .GT. .GE.	== ~= < <= > >=	= <> < <= > >=	== != < <= > >=
read variable	READ*, a	a=input('a: ')	readline readstat sscanf	scanf
Output	WRITE PRINT	<var>, disp fprintf	printf write, writeto, appendto	printf fprintf
global vars	COMMON /foo/ a, x(3)		global a,x[3]	declared before main() extern float x[];

Functions	FUNCTION fn(x,y) fn=... return	function y=fn(x,y) y=...	f:= (x,y) -> fn(x,y)	<rtn> fn(int n, float x) {...; } rtn = void int float
Subroutines	SUBROUTINE foo(N,a,x) ... return end		foo:=proc() ...; end:	
program files	code.f	code.m	code (code.m)	code.c
compile	f77 code.f	(interpreted)	(interpreted) mint code (debug)	cc -c code.c cc -c code.c -lm (math lib)
execute	a.out	>> code or: matlab < code.m	> read(code); or: maple < code > out	a.out
I/O to files	a.out < dat > out	>> load dat >> diary out		a.out < dat > out