



Física Computacional

Escuela de Física

M.R.Fulla¹

¹Escuela de Física, Universidad Nacional de Colombia Sede
Medellín

marlonfulla@yahoo.com- Oficina:21-408

<https://sites.google.com/view/fiscomunalmed/>

August 21, 2023

Conjunto de Caracteres de Fortran

TABLE 2-1
The Fortran character set

Number of symbols	Type	Values
26	Uppercase letters	A-Z
26	Lowercase letters	a-z
10	Digits	0-9
1	Underscore character	_
5	Arithmetic symbols	+ - * / **
17	Miscellaneous symbols	() . = , ' \$: ! " % & ; < > ? and blank
11	Additional Fortran 2003 symbols	~ \ [] ` ^ { } # and @

Fortran es insensible a letras mayúsculas y minúsculas

Estructura de un Programa

Declaración



Ejecución



Finalización



```
ejemplo - ejemplo.f90

ejemplo.f90 x
G (Global Scope) - p ejemplo
1  !*****
2  !
3  ! PROGRAM: ejemplo.F90
4  !
5  ! PURPOSE: Mostrar las secciones de un programa.
6  !
7  !*****
8
9  PROGRAM ejemplo
10 ! Aquí comienza la sección de declaraciones
11   INTEGER :: i,j,k
12
13 ! Aquí comienza la zona de ejecución
14
15   WRITE(*,*) "Introduzca dos numeros enteros:..."
16   READ(*,*) i,j
17   k=i+j           ! se suman los dos números introducidos
18   WRITE(*,*) "Resultado = ",k
19
20 ! Aquí comienza la zona de finalización
21   STOP
22   END PROGRAM ejemplo
23
24
```

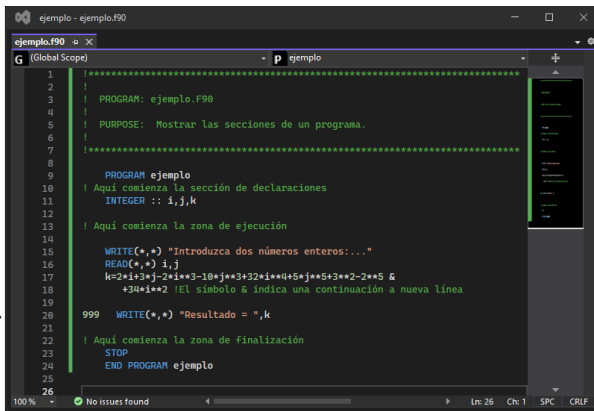
100% No issues found Ln: 24 Ch: 1 SPC CRLF

Estructura de un Programa

No ejecutable →

Nueva Línea & →

Etiqueta de
instrucción →



```
1  ! *****  
2  !  
3  ! PROGRAM: ejemplo.F90  
4  !  
5  ! PURPOSE:  Mostrar las secciones de un programa.  
6  !  
7  ! *****  
8  
9  PROGRAM ejemplo  
10 ! Aqui comienza la sección de declaraciones  
11   INTEGER :: i,j,k  
12  
13 ! Aqui comienza la zona de ejecución  
14  
15   WRITE(*,*) "Introduzca dos números enteros:..."  
16   READ(*,*) i,j  
17   k=2*i+3*j-2*i**3-10*j**3+32*i**4+5*j**5+3**2-2**5 &  
18     +34*i**2 !El símbolo & indica una continuación a nueva línea  
19  
20 999  WRITE(*,*) "Resultado = ",k  
21  
22 ! Aqui comienza la zona de finalización  
23   STOP  
24   END PROGRAM ejemplo  
25  
26
```

100 % No issues found Ln: 26 Ch: 1 SPC CRLF

- ▶ Palabras Reservadas (Keywords): Mayúsculas
- ▶ Constantes: Mayúsculas (PI)
- ▶ Variables: Minúsculas
- ▶ Nombres de programas y procedimientos: Minúsculas

Tipos de Datos Intrínsecos "Built-in" en FORTRAN



UNIVERSIDAD
NACIONAL
DE COLOMBIA

Numéricos

- ▶ INTEGER
- ▶ REAL
- ▶ COMPLEX

Lógicos

- ▶ LOGIC

Caracter

- ▶ CHARACTER

Precisión y Rango

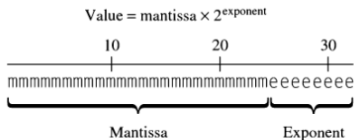
Enteros:

más pequeño: -2^{n-1} / más grande: $2^{n-1} - 1$

para 4 bytes = 32 bits

$$-2,147,483,648 \rightarrow 2,147,483,647$$

Reales:



mantisa 24 bits : $\pm 2^{23}$
exponente 8 bits : 2^{-128} y $2^{127} \Rightarrow 10^{-38} - 10^{38}$

$(-3.402823 \cdot 10^{39}, -1.175495 \cdot 10^{-39}), (1.175495 \cdot 10^{-39}, 3.402823 \cdot 10^{39})$: 7 dígitos

Declaración Explícita y por Defecto

Explícita:

INTEGER :: i,j,k

REAL :: tiempo

CHARACTER(len=9) :: resultado, mensaje

CHARACTER(9) :: resultado

El símbolo :: es opcional para compatibilidad con versiones anteriores de Fortran.

Por Defecto:

Si el tipo de variable no es explícitamente especificada entonces por defecto las variables que comiencen por las letras I,J,K,L,M o N, se asumirán como INTEGER, las demás se asumirán como REAL.

REAL, PARAMETER :: PI=3.141593

CHARACTER, PARAMETER :: mensaje = "resultado ="

Las constantes en un programa nunca deben ser modificadas dentro del programa para guardar consistencia con el modelo físico en estudio.

Operadores:

- = Asignación (no confundir con igualdad matemática, $i=i+1$)
- + Adición
- Sustracción
- * Multiplicación
- / División
- ** Potenciación

El computador automáticamente trunca la parte fraccionaria de la división de dos enteros.

$$\frac{3}{4} = 0 \quad \frac{6}{4} = 1 \quad \frac{8}{4} = 2 \quad \text{etc...}$$

La precisión limitada de los reales, a menudo presentan resultados ligeramente diferentes.

$$3. * (1./3.) \neq 1 \quad 2. * (1./2.) = 1$$

Jerarquía entre operaciones

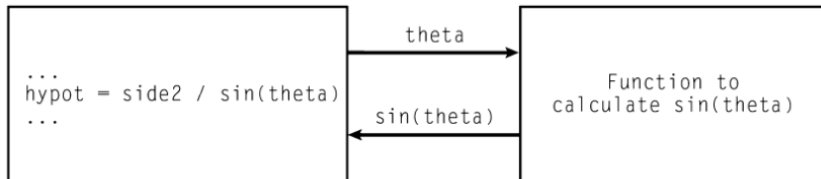
1. Primero se evalúan las expresiones entre paréntesis.
2. Se evalúan todas las operaciones con potencias de derecha a izquierda.
3. Se evalúan todas las operaciones de multiplicación y división de izquierda a derecha.
4. Se evalúan todas las operaciones de suma y diferencia de izquierda a derecha.

$a = 3, b = 2, c = 3$ entonces $a * * b * * c = 3. * * 8. = 6561.$

Funciones Intrínsecas

Funciones construidas dentro del lenguaje Fortran. Sus valores de entrada se definen como **argumentos**.

$$y = \text{SIN}(\text{theta})$$



Funciones Intrínsecas



UNIVERSIDAD
NACIONAL
DE COLOMBIA

Function name and arguments	Function value	Argument type	Result type	Comments
SQRT(X)	\sqrt{x}	R	R	Square root of x for $x \geq 0$
ABS(X)		R / I	*	Absolute value of x
ACHAR(I)		I	CHAR(I)	Returns the character at position I in the ASCII collating sequence.
SIN(X)	$\sin(x)$	R	R	Sine of x (x must be in <i>radians</i>)
COS(X)	$\cos(x)$	R	R	Cosine of x (x must be in <i>radians</i>)
TAN(X)	$\tan(x)$	R	R	Tangent of x (x must be in <i>radians</i>)
EXP(X)	e^x	R	R	e raised to the x th power
LOG(X)	$\log_e(x)$	R	R	Natural logarithm of x for $x > 0$
LOG10(X)	$\log_{10}(x)$	R	R	Base-10 logarithm of x for $x > 0$
IACHAR(C)		CHAR(I)	I	Returns the position of the character C in the ASCII collating sequence.
MOD(A,B)		R / I	*	Remainder or Modulo function
MAX(A,B)		R / I	*	Picks the larger of a and b
MIN(A,B)		R / I	*	Picks the smaller of a and b
ASIN(X)	$\sin^{-1}(x)$	R	R	Inverse sine of x for $-1 \leq x \leq 1$ (results in <i>radians</i>)
ACOS(X)	$\cos^{-1}(x)$	R	R	Inverse cosine of x for $-1 \leq x \leq 1$ (results in <i>radians</i>)
ATAN(X)	$\tan^{-1}(x)$	R	R	Inverse tangent of x (results in <i>radians</i>)

Notes:

* = Result is of the same type as the input argument(s).

R = REAL, I = INTEGER, CHAR(I) = CHARACTER(len = 1)

Instrucciones de entrada y salida

Entrada: READ(*,*) *lista – variables – entrada*

Salida: WRITE(*,*) *lista – variables – salida*

(*,*)
↑↑
Unidad de Entrada/Salida Formato

```
INTEGER :: i,j  
REAL :: a  
CHARACTER(len=12) :: chars  
READ (*,*) i,j a, chars  
...
```

Program

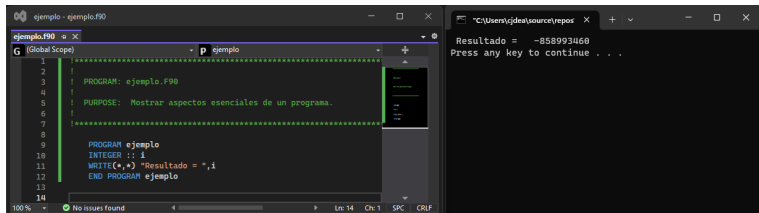
1, 2, 3., 'This one.'

Input data

i	1
j	2
a	3.
chars	'This one.'

Results

Inicialización de Variables



The screenshot shows an IDE with two windows. The left window, titled 'ejemplo.f90', displays the following Fortran code:

```
1 |  
2 |  
3 | PROGRAM: ejemplo.F90  
4 |  
5 | PURPOSE:  Mostrar aspectos esenciales de un programa.  
6 |  
7 | *****  
8 |  
9 | PROGRAM ejemplo  
10 | INTEGER :: i  
11 | WRITE(*,*) "Resultado = ",i  
12 | END PROGRAM ejemplo  
13 |  
14 |
```

The right window, titled 'C:\Users\jidea\source\repos', shows the program's output:

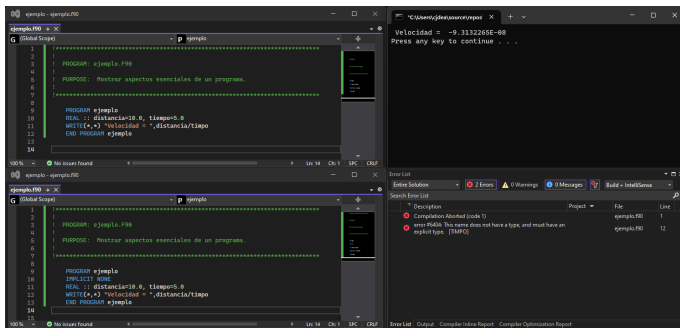
```
Resultado = -858993460  
Press any key to continue . . .
```

tipo :: var1=val1, var2=val2...

Siempre inicializar todas las variables en el programa antes de utilizarlas.

IMPLICIT NONE

Cuando se incluye la instrucción **IMPLICIT NONE** después de **PROGRAM** y antes de la sección de declaración, cualquier variable que no aparezca en forma explícita, se considerará como un error.



The screenshot displays an IDE with two windows. The left window shows a Fortran program named 'ejemplo.f90'. The code is as follows:

```
1 .....  
2 .....  
3 PROGRAM: ejemplo.F90  
4 .....  
5 PURPOSE: Ilustrar aspectos esenciales de un programa.  
6 .....  
7 .....  
8 .....  
9 .....  
10 PROGRAM ejemplo  
11 REAL :: distancia=10.0, tiempo=5.0  
12 WRITE(*,*) 'Velocidad = ', distancia/tiempo  
13 END PROGRAM ejemplo  
14 .....
```

The right window shows the output of the program, displaying 'Velocidad = -9.31322695E-08' and 'Press any key to continue . . .'. Below the output, the 'Error List' panel shows two errors:

Description	Project	File	Line
Compilation Aborted (code 1)		ejemplo.F90	1
error #6406: This name does not have a type, and must have an explicit type. [TIMPO]		ejemplo.F90	12

Depuración de Programas (Debugging)

- ▶ Errores de Sintaxis
- ▶ Errores en tiempo de Ejecución (run-time errors)
- ▶ Errores Lógicos (respuestas incorrectas o no deseadas)

¿Quiénes contribuyen a disminuirlos ?

- ▶ Compilador
- ▶ Buenas prácticas de programación
- ▶ IMPLICIT NONE, etc.

- ▶ Fraccionar expresiones largas (&)
- ▶ Uso de Paréntesis
- ▶ Verificar el tipo de argumentos introducidos en funciones
- ▶ Aritmética de enteros y reales
- ▶ Usar WRITE para desplegar valores intermedios
- ▶ Un revisor externo o Debugger

Actividad

Escribir un programa que permita despliegue en consola las dos raíces de una ecuación cuadrática de la forma $ax^2 + bx + c = 0$.

Entradas: a , b y c .

Salidas: x_1 y x_2 .