

# I tipi di dato del Fortran

In Fortran sono disponibili 5 tipi di dato (tipi primitivi o predefiniti):

- **INTEGER**
  - **REAL**
  - **COMPLEX**
  - **CHARACTER**
  - **LOGICAL**
- 
- tipi numerici
- tipi non numerici

Non considereremo il tipo COMPLEX

## Il tipo INTEGER

È costituito da un sottinsieme limitato dei numeri interi, la cui dimensione dipende dal calcolatore e/o dal traduttore usati (perché ?).

Per il compilatore g77:

valore intero minimo: -2.147.483.648

valore intero massimo: +2.147.483.647

### Operazioni ammesse

assegnazione =

somma +

sottrazione -

moltiplicazione \*

divisione intera /

potenza \*\*

confronto >,<,>=,<=,==,./=

## Il tipo REAL

È costituito da un sottinsieme limitato e discreto dei numeri reali, la cui dimensione dipende dal calcolatore e/o dal traduttore usati.

Per il compilatore g77:

valore reale minimo:  $-1.70141183E+38$  circa

valore reale massimo:  $+1.70141183E+38$  circa

precisione:  $1.40129846E-45$  circa

### Operazioni ammesse

assegnazione =

somma +

sottrazione -

moltiplicazione \*

divisione /

potenza \*\*

confronto >,<,>=,<=,==,./=

**Nota:**

**sebbene siano formalmente le stesse operazioni, hanno una realizzazione diversa rispetto al tipo INTEGER**

## **Il tipo CHARACTER**

Consiste in un insieme di caratteri, alcuni stampabili (caratteri alfabetici, cifre, caratteri di punteggiatura, ecc.) ed altri non stampabili tramite i quali si gestisce il formato dell'input/output (caratteri di controllo).

I sottinsiemi delle lettere e delle cifre sono ordinati e coerenti.

Per la rappresentazione interna, viene tipicamente usato il codice ASCII, che mette in corrispondenza ogni carattere con un numero intero compreso tra 0 e 255.

## Il tipo LOGICAL

È un tipo costituito da due soli valori (indicati con `.FALSE.` e `.TRUE.`), necessario per gestire le informazioni di tipo logico (es. il risultato di un confronto).

### Operazioni ammesse

assegnazione	=
disgiunzione	<code>.OR.</code>
congiunzione	<code>.AND.</code>
negazione	<code>.NOT.</code>
equivalenza	<code>.EQV.</code>
diversità	<code>.NEQV.</code>

x	y	x .AND. y	x .OR. y	x .EQV. y	x .NEQV.y
.FALSE.	.FALSE.	.FALSE.	.FALSE.	.TRUE.	.FALSE.
.FALSE.	.TRUE.	.FALSE.	.TRUE.	.FALSE.	.TRUE.
.TRUE.	.FALSE.	.FALSE.	.TRUE.	.FALSE.	.TRUE.
.TRUE.	.TRUE.	.TRUE.	.TRUE.	.TRUE.	.FALSE.

x	.NOT. x
.FALSE.	.TRUE.
.TRUE.	.FALSE.

## Struttura di un programma Fortran

```
PROGRAM prova1
  ! Sezione dichiarativa
  INTEGER :: n
  REAL :: x
  LOGICAL :: trovato
  REAL,PARAMETER :: pigreco=3.141593

  ! Istruzioni

END PROGRAM
```

## Definizione di variabili

Per definire una nuova variabile, la sintassi è:

*nome\_tipo :: nome\_variabile*

Il nome della variabile contiene al massimo 31 caratteri. I caratteri ammessi sono lettere, cifre e carattere di sottolineatura (underscore `_`), messi in qualunque ordine, purchè il primo carattere del nome sia una lettera

La definizione può contenere più variabili dello stesso tipo:

`REAL :: x, y, z`

Nella definizione, una variabile può essere anche inizializzata:

`INTEGER :: n=0`

Nello scegliere il nome per le variabili, è consigliabile orientarsi verso nomi significativi del ruolo della variabile nel programma.



## Dichiarazione implicita

Il Fortran ammette anche la “dichiarazione implicita” delle variabili: tutte le variabili il cui nome inizia con i,j,k,l,m,n sono automaticamente ritenute di tipo intero, mentre sono ritenute di tipo reale tutte le altre.

Ciò può portare seri problemi alla leggibilità, manutenzione e correttezza del programma. Per questo motivo, è preferibile rimuovere questa possibilità con l'istruzione

```
IMPLICIT NONE
```

da inserire dopo l'istruzione PROGRAM e prima di ogni istruzione di definizione di variabili.

## Costanti

### Costanti di tipo intero

Sono definite come sequenze di cifre decimali, eventualmente precedute da un segno (+ o -):

0 -1 3256 +34

### Costanti di tipo reale

Sono definite come sequenze di cifre decimali, eventualmente precedute da un segno (+ o -), strutturate in virgola fissa o in virgola mobile (floating point):

0.1 -3.7 0.0001 1.0E-4 -7.6E12

### Costanti di tipo carattere

Sono definite come sequenze di caratteri racchiusi tra singoli apici (') o doppi apici ("):

'x' "x" 'Valore di n: ' '123' "Il risultato e': "

### Costanti di tipo logico

Sono solo due: .FALSE. e .TRUE.

## La dichiarazione **PARAMETER**

È possibile assegnare un nome ad una costante tramite l'attributo **PARAMETER** in una dichiarazione:

```
REAL, PARAMETER :: pigreco=3.141593
```

Questa è la sintassi del Fortran 90. La sintassi richiesta dal compilatore g77 è invece:

```
REAL :: pigreco
```

```
PARAMETER (pigreco=3.141593)
```