

13_introduzione

June 4, 2023

```
[ ]: !fortran = formula translator; linguaggio compilato invece che interpretato
      ↳(come python) -> al posto
      !dell'interprete c'è il compilatore. Si scrive il codice sorgente e il
      ↳compilatore produce un file che
      !esegue questo programma direttamente. Una macchina è in grado di leggere il
      ↳programma solo se la
      !macchina è compatibile.

      !compilatore: wsl (windows subsystem for linux)
      !gfortran

      !SINTASSI

      !iniziare il programma -> program
      !finire il programma -> end program

program hello !(hello = nome)
    implicit none

    !numeri: innanzitutto bisogna dichiarare di che tipo la variabile è (vuol
    ↳dire alla stessa
    !variabile non si può assegnare un altro valore); tutte le dichiarazioni
    ↳devono precedere
    !l'uso stesso della variabile. Nonostante sia una limitazione, ciò permette
    ↳al programma di
    !compilare il programma molto velocemente.

    integer :: i !numeri interi
    double precision :: x !numeri reali
    double complex :: z !numeri complessi
    logical :: b !serve a definire i controlli di flusso (true, false, else
    ↳come in python)

    print *, "Hello World" !stampa nella riga di comando

    i = 5
```

```

print *, i

x = 5.0d0 !per 100 #invece che e^, e l'output non è precisissimo per i
↪numeri di cifre
print *, x

z = (1.0d0,2.0d0) !si deve dividere la parte reale da quella immaginaria,
↪che sono due numeri
! in doub.prec.

b = .true.
print *, b

end program hello !il nome può non ripetersi

!fortran non legge gli spazi, si potrebbe scrivere tutto senza
!per compilare il programma, comando nella shell: gfortran -o "nome del file"
↪eseguitibile (output).x"
!"nome del file con estensione .f90"
!cambiando il sorgente, continuo a far andare avanti l'eseguitibile di prima,
↪quindi devo compilare
!nuovamente ed eseguire il nuovo eseguitibile (ciclo di esecuzione del codice più
↪complicato)

open(unit=10, file="nome-out.txt", status="unknown") !stampare output in un file
write(10, "(F8.3)") x !formato di default = *; F8.3 = float con 8 cifre di cui
↪3 decimali
!E al posto della F funziona esattamente come su python
close(10)

```