Computación

Resumen de la clase anterior

- Un bit es un 0 un 1
- 8 bits son 1 Byte
- Los Bytes se usan para construir variables: Enteras, Reales, Complejas, Caracteres, Lógicas
- Los Bytes se usan para medir los tamaños de medidas de distintos tipos de memoria y almacenamiento. Además por unidad de tiempo se usan para medir velocidad de transmisión de la información-> tiene unidades de medida para grandes cantidades de bytes.

- Se usa la notación anglosajona . y , se usan al revés que nosotros estamos acostumbrados.

- La notación científica se escribe con la letra E. $1.23 x10^{-8}$ se escribe como 1.23E-8

- Complejos -> son dos reales, uno para la componente no imaginaria (real) y otro para la imaginaria

- Lógicas: Solo guardan un verdadero o un falso. Usan 1 Byte, pero de lo 8 bits solo necesitan 1.

- Caracteres: Usan un Byte por letra.

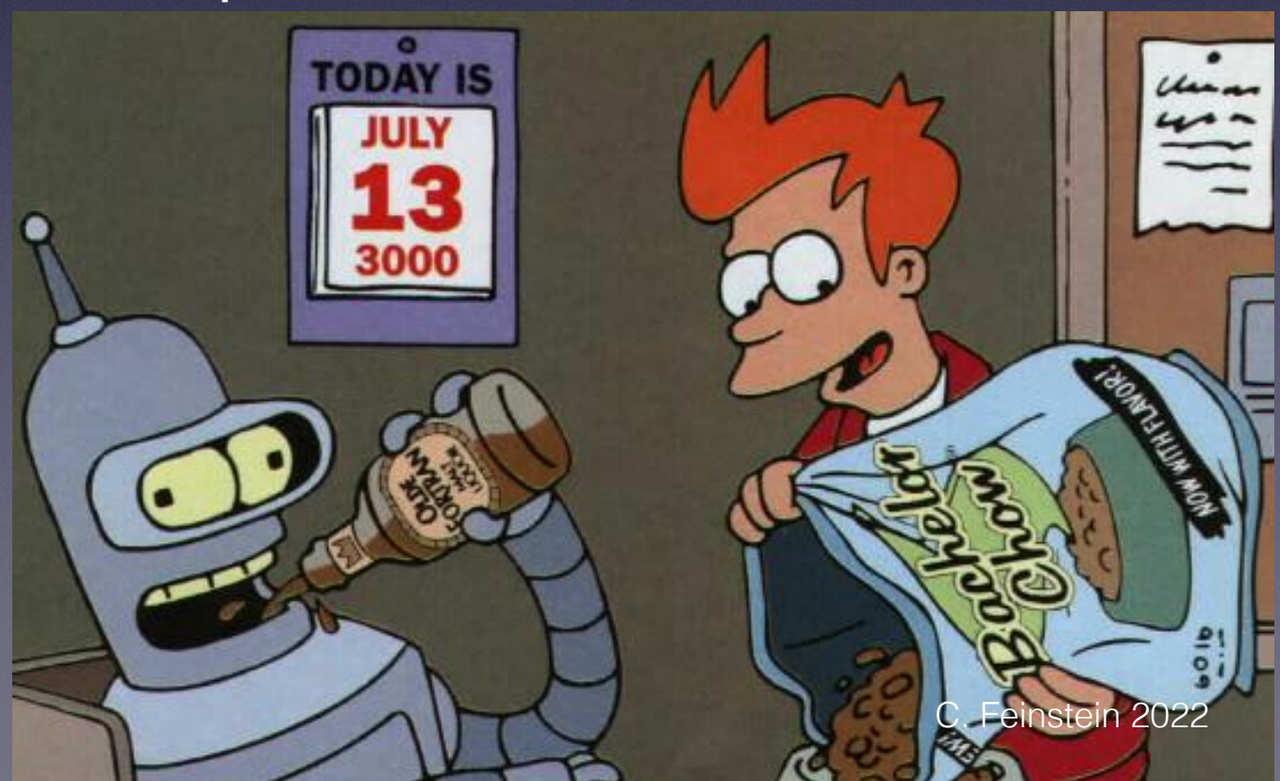
Fortran

- El nombre Fortran viene del inglés The IBM Mathematical Formula Translating System.

- Primeras versiones del año 1957.

- Es un lenguaje de cálculo, no es conveniente usar para otras actividades que no sea esa.

- Es muy (muy!!!) rápido.



C. Feinstein 2022

Puntos a favor

- Mucha experiencia y hacer compiladores extremadamente eficientes. Rápidos para hacer su trabajo y con resultados finales excelentes.
- Es un lenguaje simple, muy rápido de aprender. Pocas órdenes y concisas frente a lenguajes más modernos.
- Compatible en sus estructuras con otros programas (C o Python).
- Los compiladores optimizan el código para el hardware.
- Infinidad de algoritmos ya programados, que están en libros o en internet.
- Se pueden escribir en mayúsculas o minúsculas indistintamente.
- Se aprende rápido.

Puntos en contra

- Alguna de la sintaxis de las ordenes provienen de la época que se perforaban tarjetas
- No es orientado a objetos.
- En los lenguajes modernos hay muchas más funciones y algoritmos pre-programados (librerías).
- No incluye un sistema para hacer dibujos o gráficos.
- No es interactivo y no tiene mucha utilidad fuera de los requerimientos de científicos o de ingenieros

Asignaciones

- I=5
- IMA = 23
- FE4 = 484.24

- La asignaciones implican que el número que está a la derecha se guarda en la variable que está a la izquierda.
- El uso del “=“ fue muy cuestionado, algunos lenguajes en virtud de esto utilizan otros símbolos u órdenes para asignar.
- Las variables comienzan con una letra pero pueden seguir números después de esa letra.


Global

Variables y constantes

- Enteras (Integer)
- Reales (float)
- Complejas (Complex)
- Booleanas lógicas (Logical)
- Caracteres (character)

Implementación en Fortran Variables Enteras

- Pueden ser de 2,4 u 8 Bytes (las de 2 se usan en astronomía)
- Las Variables que comienzan con las letras I, J, K, L, M y N son enteras, a menos que de una orden contraria
- Se usa la orden “Integer” -> INTEGER Cuenta
- Si se escribo CUENTA = 3.14159 en cuenta sólo se guardará el número “3”. Los decimales se pierden

Reales o flotantes

- Son de 4 Bytes u 8 Bytes, en algunas computadoras hay de 16 Bytes
- Las variables que comiencen con las letras de la A hasta la H, y de la O hasta la Z, son reales de 4 Bytes, salvo orden contraria.
- Se puede escribir: REAL*8 Mag, Mag1, Mag2
- Los real*8 también se los denomina doble precisión y existe la orden: DOUBLE PRECISION

Variables complejas

- Son el conjunto de dos reales
- El número complejo puede ser REAL*4 o REAL*8 (ambas componentes del número son del mismo tipo)
- Para activar una variable de este tipo tengo que dar la orden COMPLEX
- COMPLEX A1, A2, A3, A4 <- desde ahora en adelante guardan números complejos

Variables Lógicas

- Utilizan 1 Byte (de estos sólo un bit)
- Guardan un Verdadero (True) o un Falso (False)
- Se activan con la orden LOGICAL
- LOGICAL L1, L2, L3
- En Fortran el Verdadero es .true. y el false .false. (notar los puntos)
- L1 = .true.
- L2 = .false.
- L3 = L2

Caracteres

- Son para guardar textos
- Se activan con la orden CHARACTER*n, donde n es la cantidad de Bytes, y es un Byte por letra (ASCII).
- ASCII —> American Standard Code for Information Interchange (1 Byte). —> UNICODE (hasta 4 Bytes)
- CHARACTER*20 CARTEL
- CARTEL= ‘FACULTAD DE CIENCIAS’

VECTORES, MATRICES, CUBOS Y OTROS…

- Se puede convertir una variable (de cualquier tipo) en un vector, matriz, cubo o dimensiones mayores)
  - Se hace DIMENSION A(10) y de ahí en adelante existen A(1),A(2),A(3)…A(9) y A(10)
  - Puedo escribir DIMENSION B(100,100)
  - También se puede poner
    REAL*8 B(100,100),C(100,100,100).
    INTEGER*8 IJ(1000000)

Resumen

- REAL*4 MAGNITUD
- REAL*8 IXAG, JXAG, KXAG2, JAXG4
- INTEGER XA, XB, XF5, SER
- COMPLEX*8 A,B,C
- COMPLEX*8 X(100)

Variables Vectoriales

- Real*8 A(10)
- A(8)= 5
- I= 5
- A(9)=A(8)+A(I)
- A(I+1)= sqrt(A(9))

Fortran 90/95

```
Son de la forma:
Tipo específico:: Lista de Variables
```

- INTEGER, REAL, COMPLEX, LOGICAL y CHARACTER (como siempre!!! + otras)

```
Ejemplos:
```

Las variables ZIP, Media and Total quiero que sean del tipo INTEGER

INTEGER :: ZIP, Media, Total

Las variables promedio, error, sum and ZAP quiero que sean del tipo REAL

REAL :: Average, error, sum, ZAP

Y las de tipo “*character*”

CHARACTER(LEN=15) :: Name, Street

LEN=15 significa que se usan 15 lugares (Bytes) para las letras

Variable vectoriales (o arreglos)

Dos formas:

```
Defino reales y la dimensión es individual

 real:: v(4), A(4,6)

Y lo hago en grupo

real, dimension(50):: x, y
```