

Entrada y salida estándar en Pascal

- **Instrucciones de entrada y salida:**
 - Sirven para que el programa se comuniquen con un periférico del ordenador tal como una terminal, una impresora o un disco.
 - Las instrucciones de entrada estándar, sirven para leer caracteres desde el teclado, y las instrucciones de salida estándar muestran caracteres en la pantalla.
 - En Pascal todas las operaciones de entrada/salida se realizan ejecutando procedimientos que forman parte del compilador y sus nombres son identificadores predefinidos:
 - Procedimientos de entrada: read, readln
 - Procedimientos de salida: write, writeln

Procedimientos `read` y `readln`

- Los procedimientos predefinidos `read` y `readln` permiten introducir datos durante la ejecución de un programa.
 - Las instrucciones para llamar a los procedimientos `read` y `readln` son de la siguiente forma :
 - `read(lista_de_variables);`
 - `readln(lista_de_variables);` donde :
lista_de_variables : es una lista de identificadores de variables separados por comas.
Los datos que se pueden leer son : enteros, reales, caracteres, o cadenas.
 - La acción de la instrucción es obtener, del teclado, tantos valores de datos como elementos hay en *lista_de_variables*. Los datos deberán ser compatibles con los tipos de las variables correspondientes en la lista.
 - La diferencia entre las instrucciones `read` y `readln` consiste en que `read` permite que la siguiente instrucción continúe leyendo valores en la misma línea; mientras que con `readln` la siguiente lectura se hará después de que se haya tecleado el carácter de fin de línea.

Procedimientos `write` y `writeln` (1)

- Los procedimientos predefinidos `write` y `writeln` permiten mostrar datos durante la ejecución de un programa.
 - Las instrucciones para llamar a los procedimientos `write` y `writeln` son de la siguiente forma :
 - `write(lista_de_variables);`
 - `writeln(lista_de_variables);` donde :
lista_de_variables : es una lista de identificadores de variables separados por comas.
Los datos que se pueden mostrar son : enteros, reales, caracteres, o cadenas.
 - La diferencia entre las instrucciones `write` y `writeln` consiste en que `write` permite que la siguiente instrucción continúe en la misma línea; mientras que, con `writeln` se hará en la siguiente

Procedimientos `write` y `writeln` (2)

- **Formatos de salida**
 - Cada dato que se escribe con `write` y `writeln` aparece con un determinado número de caracteres denominado longitud de campo
 - Cuando un dato se escribe sin una especificación de longitud de campo, se utilizará la especificación de campo por defecto
 - Es posible también especificar la longitud de campo para cada dato mediante:
 - `writeln (valor:anchura...);` con
anchura : expresión entera (literal, constante, variable o llamada a función) que especifica la anchura total del campo en que se escribe el valor.
 - `writeln (valor:anchura:dígitos ...);`
dígitos: dígitos decimales de un número real, *anchura*: Total de dígitos del número real contando parte entera, punto decimal y dígitos decimales.
 - Ejemplos:
 - `valor:= 25.0776;`
`writeln(valor);2.5077600000E+01` `writeln(valor:2);2.5E+01`
`writeln(valor:2:1);25.1` `writeln(valor:2:8);25.07760000`
 - `writeln('Tec':3);Tec` `writeln('Tec':5);□□Tec`