FUNDAMENTOS DE LA INFORMÁTICA

Apuntes

**Leyenda:** las funciones van en naranja (**type**)

**Tipos:**

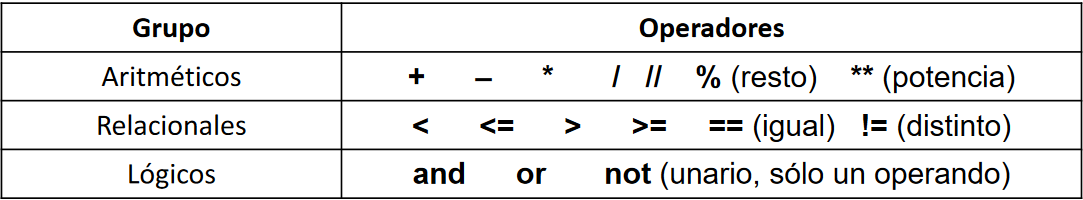
**int** (integer, entero)

**float** (decimal con coma flotante)

**str** (cadena de caracteres)

**bool** (verdadero o falso)

**type()** ->devuelve el tipo de cualquier valor o expresión ()



**/** división real (coma flotante)

**//** división entera

**+** permite concatenar cadenas

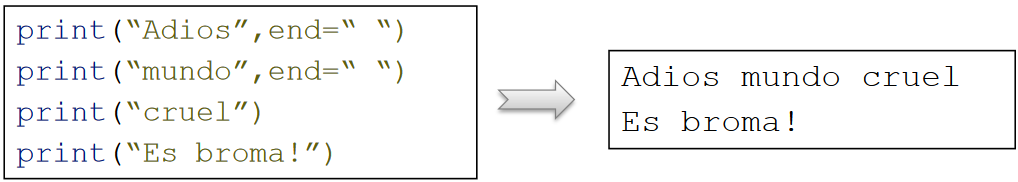
Introducción de datos mediante -> **input()**

Para devolver el valor absoluto -> **abs()**

Mostrar datos por la pantalla -> **print()**

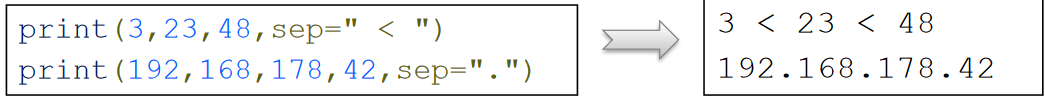
Introducción de salto de línea al final -> **end=”\n”**

Para imprimir en la misma línea usando espacios -> **end=” “**





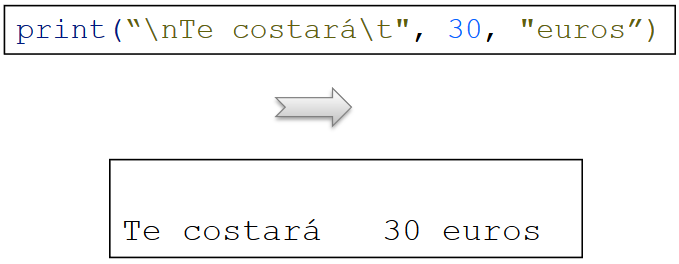
Para cambiar el separador en el print -> **sep=” “**



**Elementos de ordenación :**

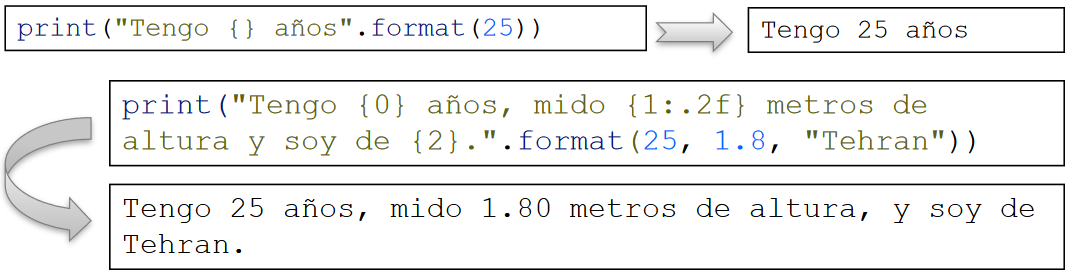
**\n** -> Introduce un salto de línea

**\t** -> pone un espacio tabulado

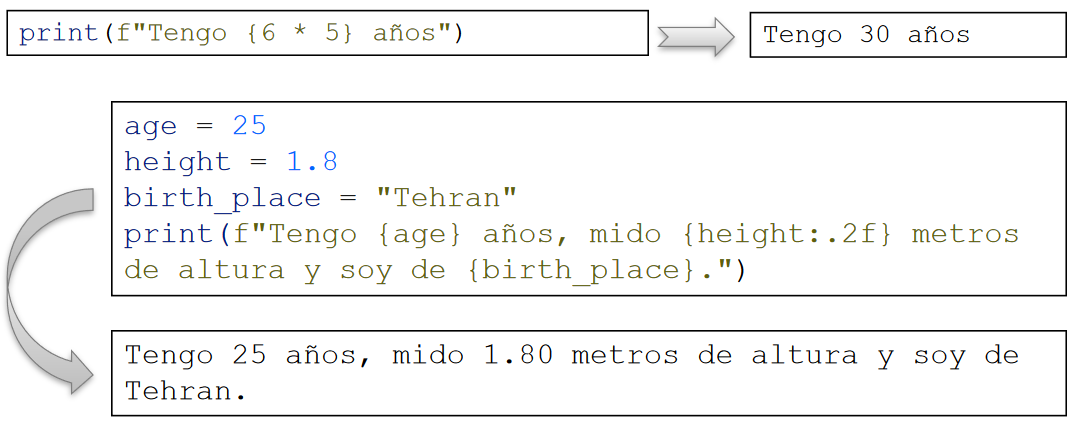


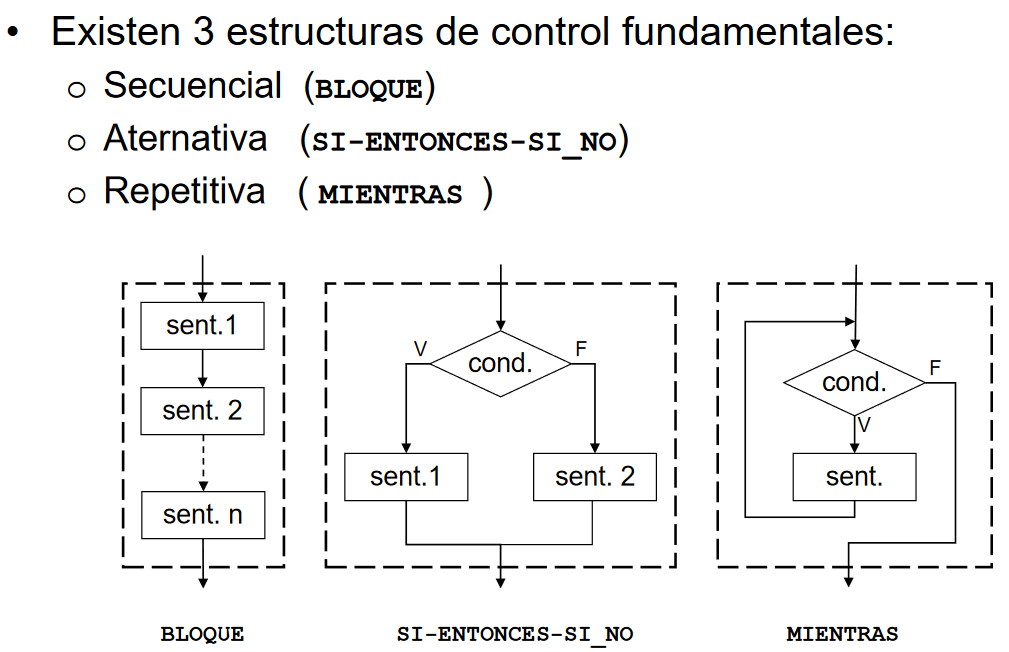
La función **.format()** permite:

* {} reemplazar el campo por el parámetro x de .format(x)
* {x1} reemplazar el campo por el parámetro x1 de .format(x1,x2,x3)
* {x1:.mf} reemplazar el campo por el parámetro x1 de .format(x1,x2,x3) con los decimales “m”

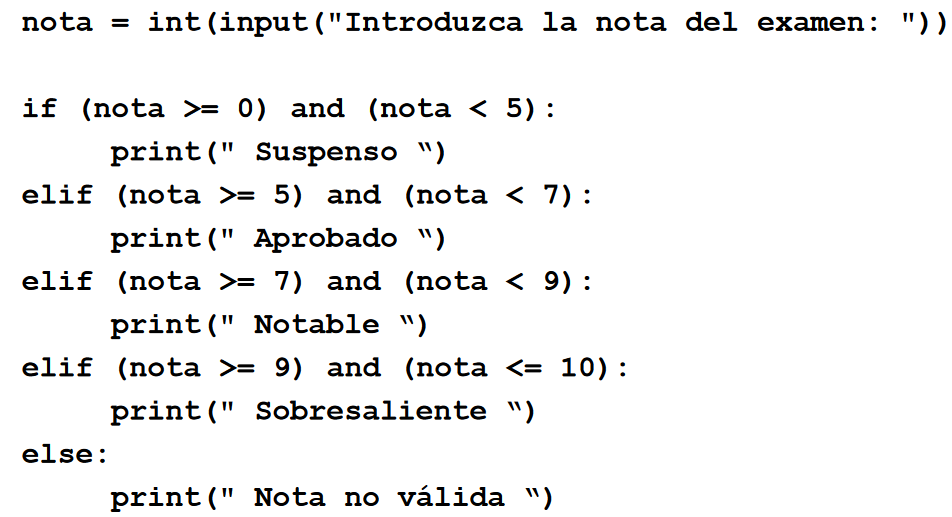
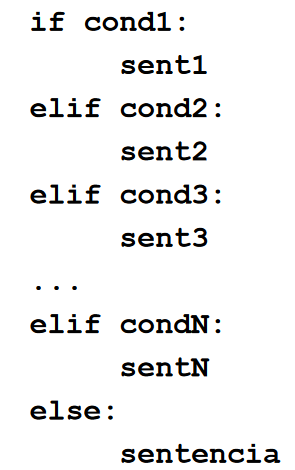


También se puede usar para strings, definiéndose con una f delante:

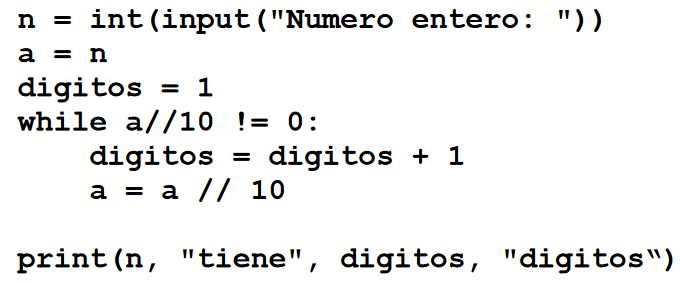
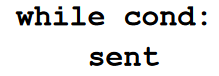




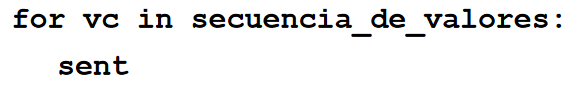
**Condicionales: Ejemplo**



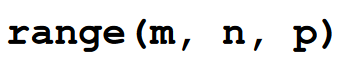
**Condicional (while): Ejemplo**



**Bucle controlado por contador(for):**



Los valores de secuencia se generarán con la función **range()**

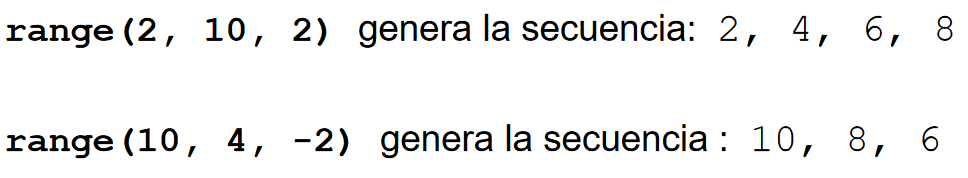


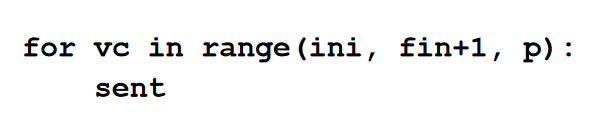
**m** es el inicio

**n** es el final

**p** es el paso

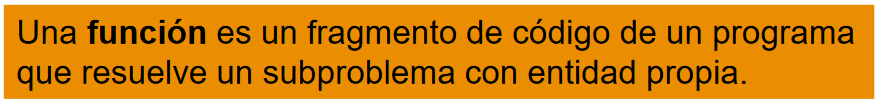
Siempre que el paso (p) sea positivo, se empieza en m y se termina en n-1. Si el paso (p) es negativo, se empieza en m y se termina en n-1.



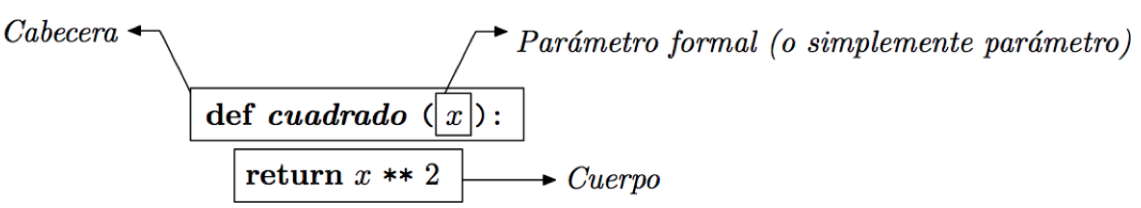


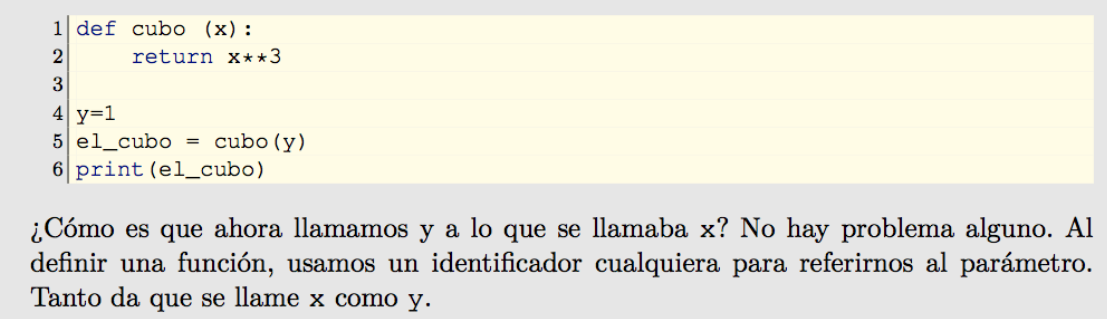
**ini** es la variable inicial

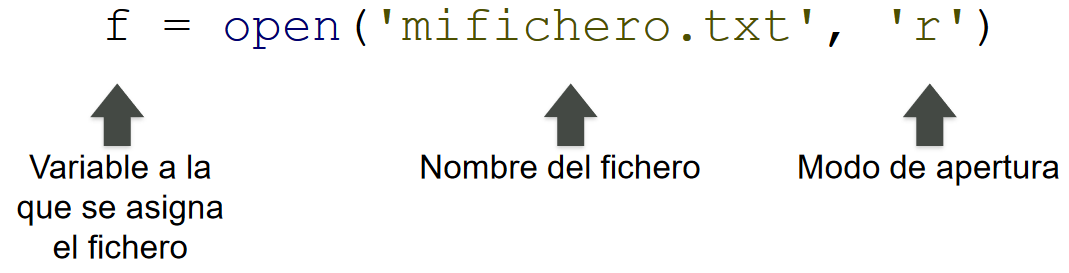
**fin** es la variable final



->Se importa una función ya existente de otro módulo





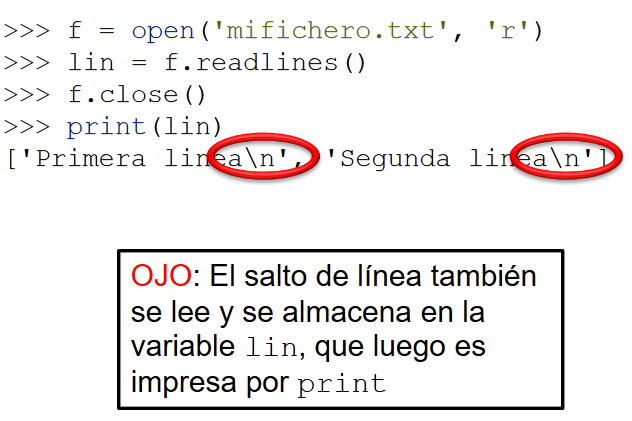
**Ficheros:** 

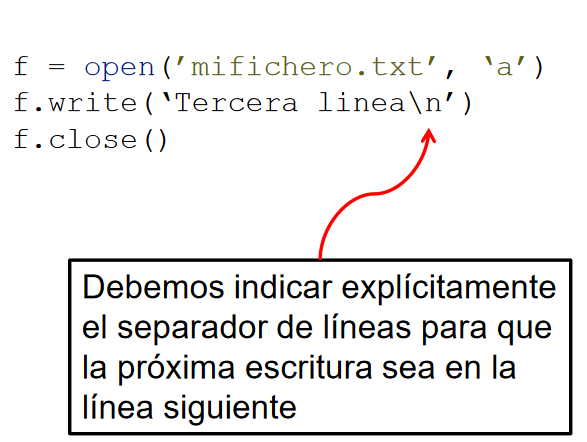
**Modos de apertura:**

**r** -> abre sólo para lectura

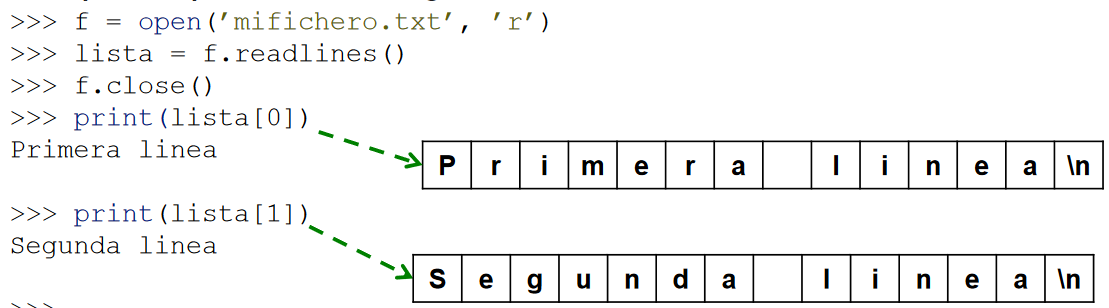
**w** -> abre para lectura y escritura

**a** -> abre para añadir. Si el fichero no existe se crea uno nuevo. Si existe se crea con permiso de escritura y se añaden los datos al final.

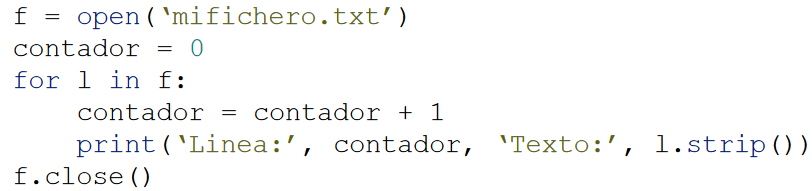




En lugar de leer una cadena de texto con el contenido completo, podemos cargar una lista con las líneas:



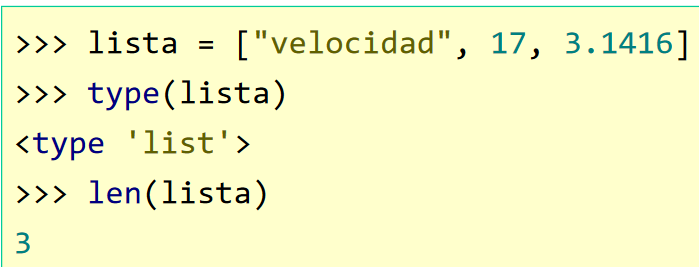
El bucle for se puede usar para leer línea a línea un fichero:



El **.strip()** se usa para que no salga un salto de línea adicional en la consola.

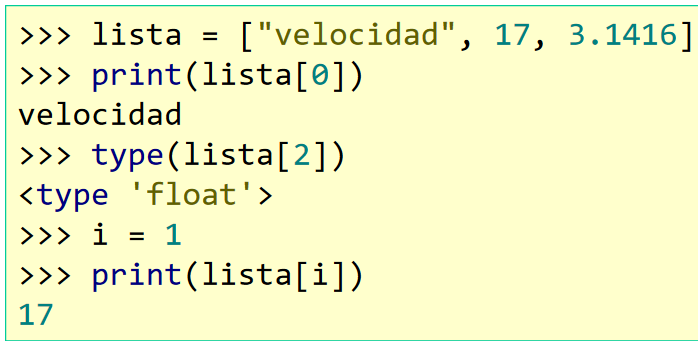
**Listas:**

Una lista es una secuencia de datos de cualquier tipo. Su número de elementos puede cambiar durante la ejecución del programa.



La función **len()** nos devuelve el tamaño de la lista (nº de elementos)

Para acceder a los elementos de una lista se usa un índice. Hay que tener en cuenta que las listas empiezan en 0 (0,1,2,3,4…).



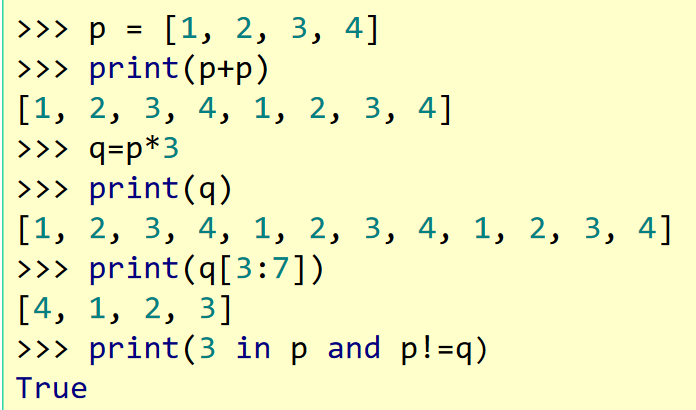
**Operadores:**

Operador de concatenación -> **+**

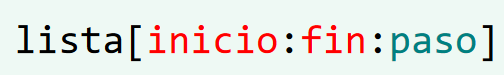
Operador de repetición -> **\***

Operador de corte (slicing) -> [**:**]

Comparaciones, pertenencia, identidad -> **== != in is**



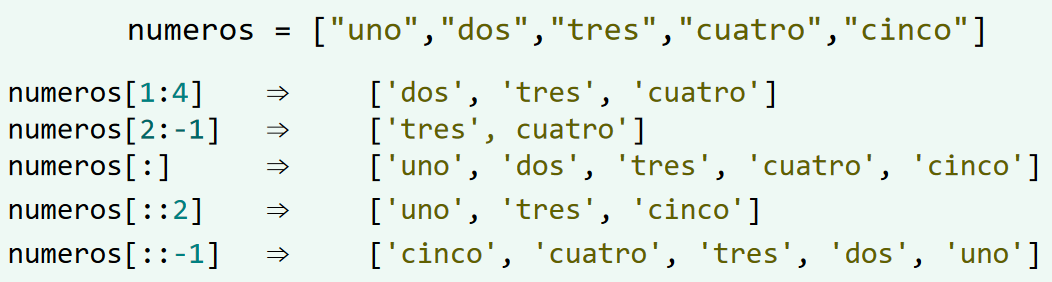
**Slicing:** selecciona los elementos de la lista desde el inicio hasta fin-1 de paso en paso.



Si se omite inicio o fin se toman el primer o último elemento.

Valores negativos significan posiciones a contar desde el final (-1) hacia atrás.

Si se omite el paso, se supone que es 1.

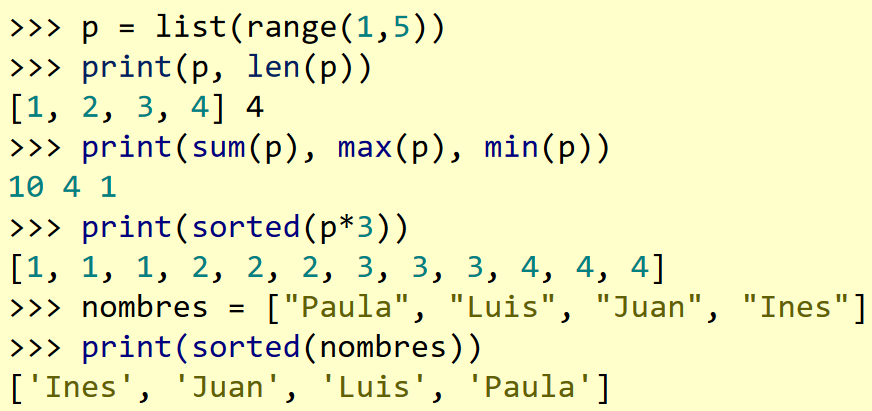


Longitud -> **len()**

Creación de listas numéricas -> **list(range())**

Funciones para listas numéricas -> **sum()** “suma” **max()** “máximo” **min()** “mínimo”

Ordenación de elementos -> **sorted()**

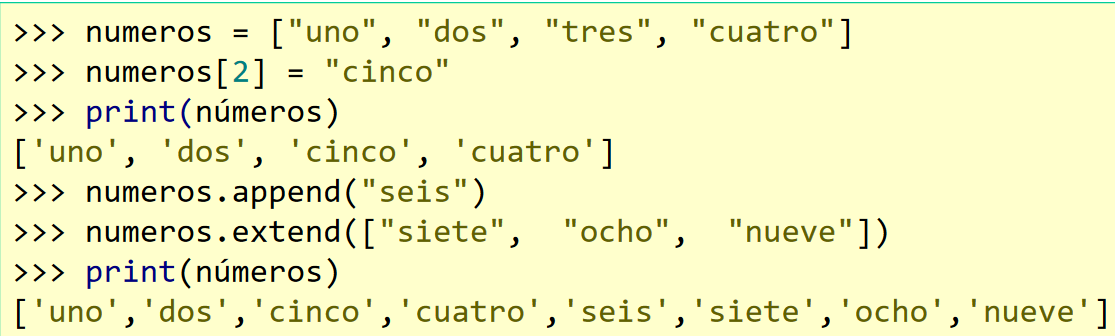


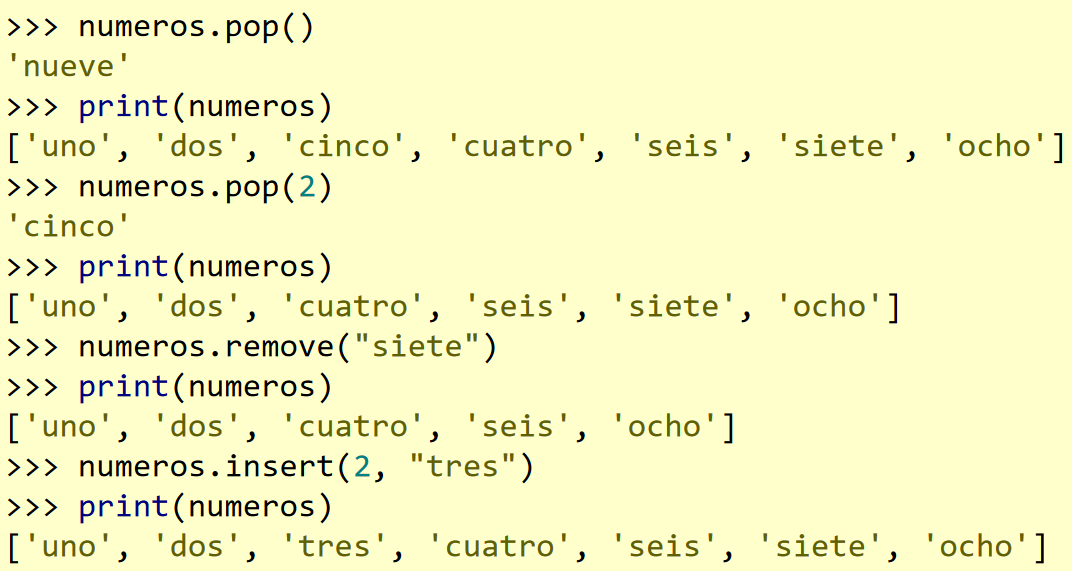
Las **listas** son variables mutables. Quiere decir que los métodos pueden cambiar la información de una variable sin tener que crear una nueva modificada.

Asignación directa a un elemento -> **p[1] = 7**

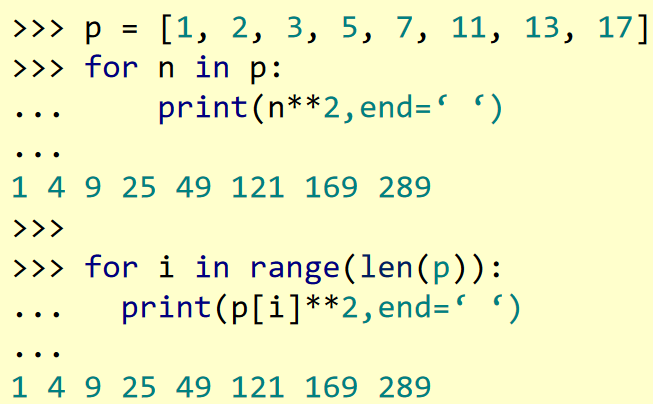
Métodos de inserción -> **.append() .extend()** **.insert()**

Métodos de borrado -> **.pop() .remove()**



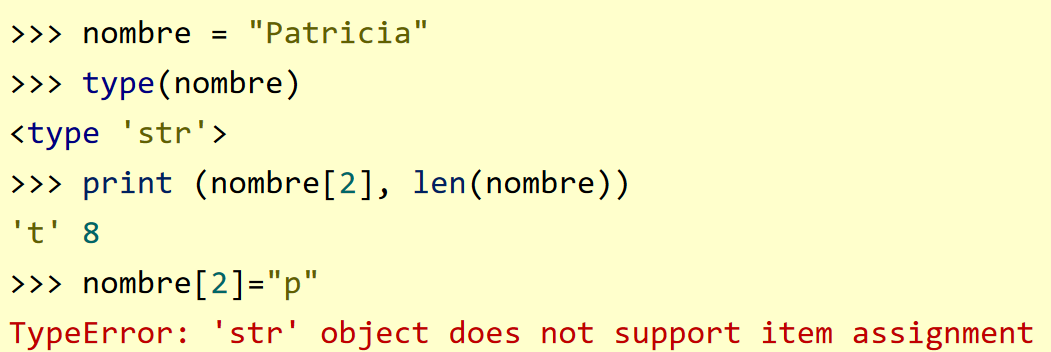


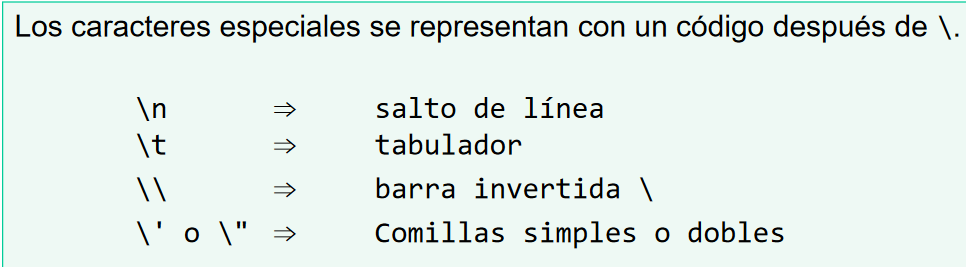
Se puede acceder a la lista usando el **for**:

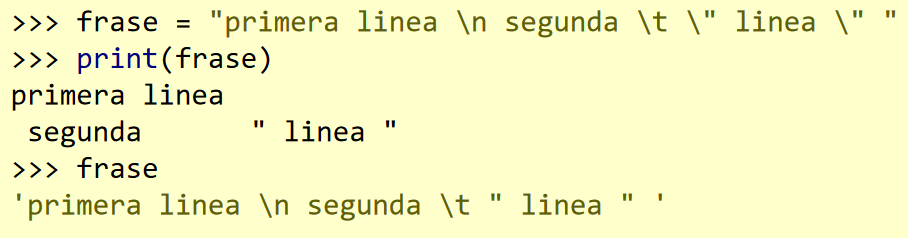


**Cadenas:**

Una cadena es un tipo especial de lista donde sus elementos son caracteres y se representan encerrados entre comillas. Son inmutables así que no se puede cambiar su contenido directamente.

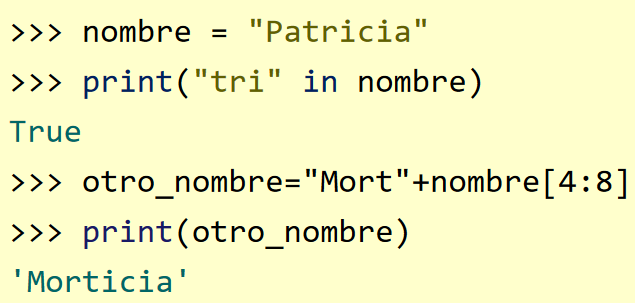






Se usan los mismos operadores que en las listas (**+ \*** [ **:** ] **== !=**)

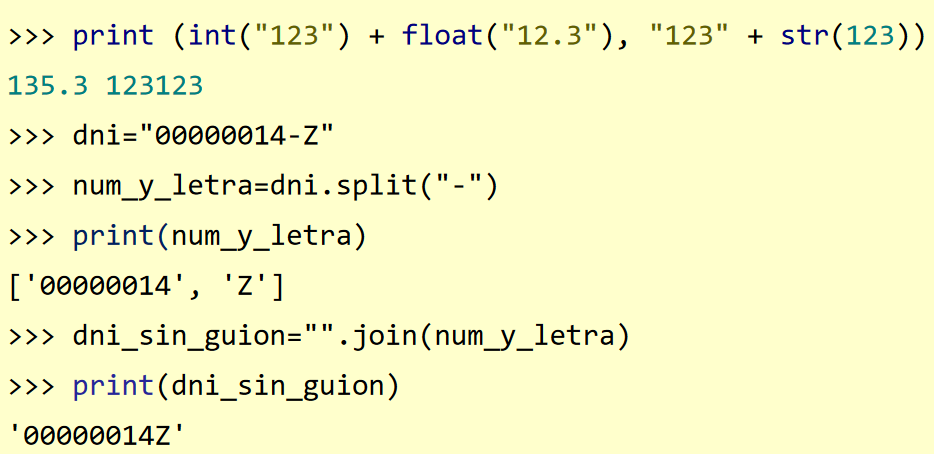
Pertenencia **in** (permite buscar caracteres o subcadenas)



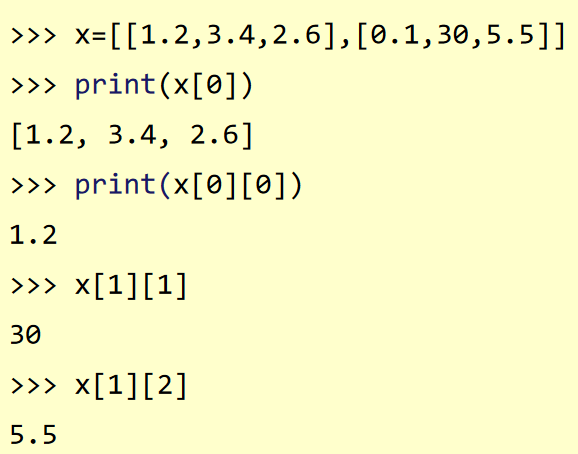
Longitud -> **len(cadena)**

Lectura -> **input()**

Métodos -> **.join(secuencia) .split(separador)**

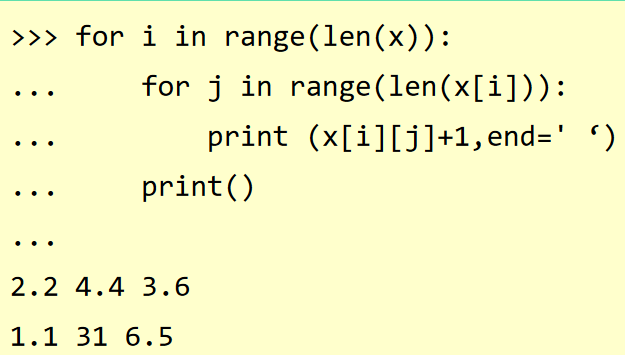


**Matrices:**



Número de filas -> **len(x)**

Número de columnas -> **len(x[i])**



El bucle más interno se ejecuta primero para i con el valor 0, como si hubiésemos escrito x[0][j]+1y después con el valor 1, como si hubiésemos escrito x[1][j]+1. Cada vez, con j valiendo primero 0, después 1 y finalmente 2.

De este modo se obtienen las parejas de índices [0][0], [0][1], [0][2], [1][0], [1][1], [1][2], dentro del bucle más interno.

Se puede hacer con matrices lo mismo que con listas.

