* , dentro de print inserta por defecto un espacio como separador. Cambiar con sep="...." en el último parámetro.
* No hace conversiones implícitas. P. ej: int("123.456") da error.
* Puedo obtener 123 del ej. anterior sin que falle si hago int(float("123.456")) — Lo parsea a float y luego int() de un float devuelve su parte entera.
* "in" se llama "operador de membresía".
* Python transforma datos que no sean True o False al evaluar una condición. Los números ≠ de 0 se toman como True y el 0 como False. Todos los Strings no vacíos son verdaderos, y el String vacío se evalúa como falso.

---

* Los strings son inmutables
* Slices:

<string>[start:end:step]

- start: default = 0
- end: default = len(<string>) (hasta el final).
- step: default = 1 — Si es negativo toma el valor absoluto como step y recorre al revés.

---

* List(seq): genera una lista a partir de una secuencia (string, range, tupla, etc.)
* Métodos de listas:
  - append(elem): Agrega un elemento al final.
  - insert(index, elem): Agrega el elemento elem tal que quede ocupando el lugar "index"
  - pop(Index): Devuelve y quita de la lista el elemento en la posición index.
  - remove(elem): Elimina la primera aparición del elemento elem de la lista, si hay al menos 1.
* Las listas usadas como parámetros de funciones se pasan por referencia.
* La asignación genera una copia nueva de la lista siendo asignada si se le hace un cambio. Sino genera una referencia.
  - → lista = lista + [elem] genera una nueva lista con los elementos de la original, y el nuevo elemento al final.
  - → lista.append(elem) modifica la lista (las listas son mutables) sin crear una nueva.

---

* Las tuplas no son mutables y se puede crear una tupla a partir de una secuencia con tuple().
  Ejemplo: a = ("Hola") es de tipo String, por más que tenga (). b = tuple("Hola") genera ('H', 'o', 'l', 'a').
* Las tuplas soportan slices igual que los strings.
* Los op. relacionales comparan elemento por elemento y terminan al encontrar la primera diferencia
* Cuando el contenido es el mismo en 2 tuplas pero una tiene más elementos, esa es mayor.
  Ej: (0, 1, 2) < (0, 1, 2, 0)
* Está permitido hacer swaps con asignación de tuplas. Ejemplo: a, b = b, a equivale a swap(a, b)
* Solo se puede crear una tupla de 1 elemento colocando una , luego del elemento.
  Ejemplo: a = 3.5 es float ; b = 3.5, es una tupla que contiene un float.

---

* dict() crea un diccionario vacío.
* <nombre_diccionario>[<Key>] = <Value> permite crear una entrada (en listas esto no se podía).
* Crear diccionario: <nombre> = {<Key1>: <value1>, <Key2>: <value2>, .. }
* Métodos de diccionarios:
  - Keys(): devuelve las claves
    values(): devuelve los valores
    → En una "lista" (dict_keys ó dict_values) que no es indexable. Pasarla a lista con list().
  - get(Key): devuelve el value sin sacarlo. Si el Key no está NO da un error.
  - get(Key, defaultValue): devuelve el valor para la Key. Si la Key pedida no está devuelve el defaultValue (es un getOrDefault)
  - pop(Key): devuelve el value y lo quita.