

# 5 - Secuencias (string, tupla, lista)

Horas de Libre Configuración

Curso 2023 - 2024



# Recuerda



- ▶ Los tipos de datos secuencia permiten iterar por todos los elementos del objeto.
- ▶ Python ofrece métodos comunes a las secuencias.
- ▶ En esta sesión se ven los tipos de datos cadena (string), tupla y lista.

# Cadena de caracteres (string)

- ▶ Secuencia de caracteres.
- ▶ Los literales de tipo cadena se forman con un texto entre comillas simples o dobles.
- ▶ Una cadena de caracteres es un objeto **inmutable**.

# Operaciones con cadenas



- ▶ Los operadores + y \* permiten concatenar y repetir cadenas.
- ▶ La primera posición de una cadena es la cero. Se puede acceder a la posición por su índice.
- ▶ `s[i]` devuelve el carácter en la posición `i` (salta excepción si la posición no existe)

# Operaciones con cadenas

- ▶ La posición se puede indicar desde el comienzo (0.. $n-1$ ) o desde el final ( $-1$ .. $-n$ ).

0	1	2	3
H	O	L	A
-4	-3	-2	-1

# Operaciones con cadenas



- ▶ `s[i:j]`: subcadena desde `i` hasta `j-1`.
- ▶ `s[i:j:k]`: subcadena desde `i` hasta `j-1` con salto de `k` posiciones.
- ▶ `s[i:]`: subcadena desde `i` hasta el final.
- ▶ `s[:j]`: subcadena desde el inicio hasta `j-1`.
- ▶ `s[::-1]`: devuelve la cadena al revés.

# Operaciones con cadenas

- ▶ `upper()` convierte a mayúsculas.
- ▶ `lower()` convierte a minúsculas.
- ▶ `capitalize()` primera letra a mayúscula.
- ▶ `title()` la primera de cada palabra a mayúscula.
- ▶ `swapcase()` invierte mayúsculas con minúsculas (y viceversa)

# Operaciones con cadenas



- ▶ `count(s)` cuenta el número de aparaciones de `s` en la cadena.
- ▶ `count(s,ini)` a partir de la posición.
- ▶ `count(s, ini, fin)` entre dos posiciones.
- ▶ `find(s)` encuentra la primera posición de `s` (en caso de no encontrarse, -1)



# Operaciones con cadenas



- ▶ `startswith(s)` indica si la cadena comienza por `s`.
- ▶ `endswith(s)` indica si termina por `s`.
- ▶ `replace(s1, s2)` cambia toda ocurrencia de `s1` por `s2`.
- ▶ `strip()` elimina caracteres a comienzo y final.

# Operaciones con cadenas



- ▶ `split(s)` devuelve una lista de elementos a partir de trocear la cadena con `s`.

# Tuplas



- ▶ Secuencia inmutable de valores del mismo o distinto tipo.
- ▶  $t = (v1, v2, v3...)$
- ▶ El acceso es con índice [ ]. El primer valor es 0 y admite las mismas operaciones que con cadenas.

# Tuplas



- ▶ Secuencia inmutable de valores del mismo o distinto tipo.
- ▶  $t = (v1, v2, v3...)$
- ▶ El acceso es con índice [ ]. Mismo comportamiento que en cadenas.
- ▶ `index(v)`: posición del valor.
- ▶ `count(v)`: número de ocurrencias del valor.

# Listas



- ▶ Secuencia mutable de valores del mismo o distinto tipo. Una vez creada, se pueden añadir/modificar/eliminar elementos.
- ▶  $l = [v1, v2, v3...]$
- ▶  $v \text{ in } l$ : indica si el valor  $v$  está en la lista  $l$ .
- ▶  $l[i]$  devuelve el valor en la posición  $i$  (error si la posición no existe)

# Listas



- ▶ Admite las mismas operaciones de troceado que las cadenas.
- ▶ `append(v)`: añade un valor al final de la lista.
- ▶ `extend(l)`: añade una lista al final.
- ▶ `insert(p, v)`: añade valor en la posición indicada.

# Listas



- ▶ `pop()`: devuelve el último elemento de la lista y lo elimina.
- ▶ `pop(p)`: devuelve el elemento de la posición y lo elimina.
- ▶ `remove(v)`: elimina todas las ocurrencias del valor.
- ▶ `reverse()`: invierte la lista.

# Listas



- ▶ `sort()`: ordena la lista.
- ▶ `sort(reverse=True)`: ordena en orden decreciente.
- ▶ `count(v)`: número de ocurrencias del valor.
- ▶ `index(v)`: posición de la primera ocurrencia del valor.
- ▶ `Index(v, p)`: a partir de una posición `p`.