

Estructuras de datos en Python

Introducción a Python
José Miguel Gimeno

Estructuras de datos en Python



- Listas
- Tuplas
- Diccionarios

Listas

- Son conjuntos **ordenados** de elementos
- Se delimitan por corchetes `[]`
- Los elementos se separan por comas
- Son heterogéneas
- Son mutables

Listas

Las posiciones de una lista se numeran desde 0

Elemento 1	Elemento 2	Elemento 3	...	Elemento N
0	1	2	...	N - 1

Slicing en listas

Acceso a un elemento concreto:

- Primer elemento: **lista[0]**
- Tercer elemento: **lista[2]**
- Último elemento: **lista[-1]**
- Penúltimo elemento: **lista[-2]**

Slicing en listas

Acceso a varios elementos mediante **rangos**:

- El rango no incluye el índice final
- El resultado puede ser un elemento o una lista
- Se pueden usar los índices negativos
- El tercer índice indica el intervalo (salto)

Funciones básicas con listas

- Cantidad de elementos: **len(lista)**
- Solo para listas numéricas:
 - Valor máximo: **max(lista)**
 - Valor mínimo: **min(lista)**
 - Suma de elementos: **sum(lista)**

Operaciones con listas

- Concatenación de listas: $+$
- Reproducción de listas: $*$
- Operador de inclusión:
 - **in**
 - **not in**

Funciones especiales con listas

- Agregar elemento e al final:
lista.append(e)
- Insertar elemento e en la posición i:
lista.insert(i,e)
- Contar apariciones de e: **lista.count(e)**
- Eliminar la primer aparición de e:
lista.remove(e)

Funciones especiales con listas

- Devuelve elemento en la posición i y lo elimina: **lista.pop(i)**
- Posición de la primera aparición de e: **lista.index(e)**
- Ordenar de forma creciente: **lista.sort()**
- Ordenar de forma decreciente: **lista.reverse()**

Funciones especiales con listas

- Vaciar la lista: **lista.clear()**

Comprensión de listas

Sintaxis general:

[expresión **for** elemento **in** iterable **if**
condición]

if es **opcional**

Tuplas

- Son conjuntos **ordenados** de elementos
- Se delimitan por paréntesis ()
- Los elementos se separan por comas
- Son heterogéneas
- Son **inmutables**
- Es más rápida que una lista

Tuplas

Las posiciones de una tupla se numeran desde 0

Elemento 1	Elemento 2	Elemento 3	...	Elemento N
0	1	2	...	N - 1

Slicing en tuplas

Acceso a un elemento concreto:

- Primer elemento: **tupla[0]**
- Tercer elemento: **tupla[2]**
- Último elemento: **tupla[-1]**
- Penúltimo elemento: **tupla[-2]**

Slicing en tuplas

Acceso a varios elementos mediante **rangos**:

- El rango no incluye el índice final
- El resultado puede ser un elemento o una tupla
- Se pueden usar los índices negativos
- El tercer índice indica el intervalo (salto)

Cómo agregar un elemento a una tupla

- Concatenación incorrecta:
- Concatenación correcta:

Diccionarios

- Cada elemento es un par clave-valor
- Se delimitan por llaves {}
- No están ordenados
- Los elementos se separan por comas
- Pueden ser heterogéneos
- Sus claves son inmutables y sus valores mutables

Diccionarios: sintaxis

Diccionario = {clave:valor, clave:valor,...}

- Las claves pueden ser valores simples de distintos tipos
- Los valores asociados a las claves pueden ser datos simples o estructurados

Diccionarios: definición

- Diccionario nulo
 - `dic={}`
- Con clave-valor
 - `dic={'bolis':3,'lápices':8,'sacapuntas':0}`
 - ¡No están ordenados!

bolis	lapices	sacapuntas
3	8	0

Diccionarios: métodos

- dict() devuelve un diccionario si se le pasa como parámetro su representación
- zip() recibe como parámetros dos elementos de tipo lista, cadena o tupla y devuelve la relación clave-valor
 - Hay que usarlo con dict para obtener el diccionario

Diccionarios: métodos

- `items()` devuelve una lista de tuplas clave-valor
- `keys()` devuelve lista de claves
- `values()` devuelve lista de valores
- `clear()` elimina todos los elementos del diccionario

Diccionarios: métodos

- `copy()` devuelve una copia del diccionario
- `fromkeys()` recibe un iterable y un valor devolviendo un diccionario con el mismo valor en todas las claves
- `get()` devuelve el valor dada una clave
- `pop()` igual que `get`, pero eliminando el par

Diccionarios: métodos

- `setdefault()` dos formas
 - Devuelve el valor dada una clave
 - Establece un nuevo para clave-valor
- `update()` recibe otro diccionario y si tienen claves iguales actualiza el valor correspondiente, si tienen claves distintas, añade los nuevos pares