



# **Diccionarios**

CTIC - UNI

Universidad Nacional de Ingeniería

# ¿Qué es un diccionario?

- En términos de estructura de datos, un diccionario se denomina mejor una matriz asociativa, una lista asociativa o un mapeo.
- Podemos entenderlo como una lista de pares, donde el primer elemento del par, **la clave**, se utiliza para recuperar el segundo elemento, **el valor**.
- Por lo tanto, asignamos una clave a un valor.

# Pares de clave valor

- La clave actúa como un **índice** para encontrar el valor asociado.
- Al igual que un diccionario, busca una palabra por su ortografía para encontrar la definición asociada.
- Se puede buscar un diccionario para localizar el valor asociado con una clave.

# Diccionarios

- Almacena pares de entradas llamados **items**  
**{ 'CS' : '743-713-3350', 'CM' : '713-743-3333' }**
- Cada par de entradas contiene
  - Una **clave**
  - **Un valor**
- Cada clave y valor deben ser separados por dos puntos
- Los pares de entradas están separadas por comas
- Los items del diccionario están encerrados entre llaves

•  
•  
•

## Uso

- Las claves deben ser **únicas** dentro de un diccionario
  - Sin duplicados
- Si tenemos  
**edad = {'Alicia' : 25, 'Tom' :28}**  
entonces  
**edad['Alicia'] es 25**  
y  
**edad['Tom'] es 28**

# Los diccionarios son mutables

- Al igual que las listas, los diccionarios son una estructura de datos mutable.
  - puede cambiar el objeto a través de varias operaciones, como la asignación de índice.

```
mi_dic = {'juan':3, 'rita':10}  
print(mi_dic['juan']) # imprime 3  
mi_dic['juan'] = 100  
print(mi_dic['juan']) # imprime 100
```

# Los diccionarios son mutables

- `>>> edad = {'Alicia' : 25, 'Tom' : 28}`
- `>>> copia = edad`
- `>>> edad['Tom'] = 29`
- `>>> edad`  
`{'Roberto': 29, 'Alicia': 25}`
- `>>> copia`  
`{'Roberto': 29, 'Alicia': 25}`

# Las claves deben ser únicas

- `>>> edad = {'Alicia' : 25, 'Tom' : 28, 'Alicia' : 26}`
- `>>> edad`  
`{'Tom': 28, 'Alicia': 26}`



# Mostrar contenidos

- `>>> edad = {'Alicia' : 25, 'Carola': 'veintidos'}`
- `>>> edad.items()`  
`dict_items([ ('Alicia', 25), ('Carola', 'veintidos')])`
- `>>> edad.keys()`  
`dict_keys([ 'Alicia', 'Carola'])`
- `edad.values()`  
`dict_values([28, 25, 'veintidos'])`

# Actualizando directorios

- >>> edad = {'Alicia': 26 , 'Carola' : 22}
- >>> edad.update({'Tom' : 29})
- >>> edad  
{'Tom': 29, 'Carola': 22, 'Alicia': 26}
- >>> edad.update({'Carola' : 23})
- >>> edad  
{'Tom': 29, 'Carola': 23, 'Alicia': 26}

•  
•  
•

## Retornando un valor

- `>>> edad = {'Tom': 29, 'Carola': 23, 'Alicia': 26}`
- `>>> edad.get('Tom')`  
29
- `>>> edad['Tom']`  
29

## Eliminando un ítem específico (I)

- `>>> a = {'Alicia' : 26, 'Carola' : 'veintidos'}`
- `>>> a`  
`{'Carola': 'veintidos', 'Alicia': 26}`
- `>>> a.pop('Carola')`  
`'veintidos'`
- `>>> a`  
`{'Alicia': 26}`

## Eliminando un ítem específico (II)

- `>>> a.pop('Alicia')`  
26
- `>>> a`  
{}
- `>>>`

## Eliminar un item aleatorio

- `>>> edad = {'Tom': 29, 'Carola': 23, 'Alicia': 26}`
- `>>> edad.popitem()`  
`('Tom', 29)`
- `>>> edad`
- `{'Carola': 23, 'Alicia': 26}`
- `>>> edad.popitem()`  
`('Carola', 23)`
- `>>> edad`  
`{'Alicia': 26}`

# operadores comunes

Como con las otras estructuras de datos, los diccionarios responden a estos operadores:

- `len(mi_dic)`
  - número de pares clave:valor en el diccionario
- `elemento in mi_dic`
  - booleano, es elemento una clave en el diccionario.
- `for clave in mi_dic:`
  - itera a través de las claves de un diccionario.

# Resumen

- Las cadenas, listas, tuplas, conjuntos y diccionarios se ocupan de los agregados
- Existen dos grandes diferencias
  - ***Las listas y diccionarios son mutables***
    - A diferencia de cadenas, tuplas y conjuntos.
  - ***Los Strings, listas y tuplas son ordenadas***
    - A diferencia de los conjuntos y diccionarios.



# Agregados mutables

- Puede modificar elementos individuales
  - $x = [11, 22, 33]$   
 $x[0] = 44$   
trabaja
- No se puede guardar el valor actual
  - $x = [11, 22, 33]$   
 $y = x$   
no trabaja

# Agregados inmutables

- No se pueden modificar items individuales
  - `s = 'hola!'`  
`s[0] = 'H'`  
es un **ERROR**
- Puede guardar el valor actual
  - `s = 'hola!'`  
`t = s`  
si funcionará.

# Agregados ordenados

- Se puede acceder a los items de la colección a través de un **índice numérico**.

– **s = 'Hola!'**

**s[0]**

– **x = ['Alicia', 'Tom', 'Carola']**

**x[-1]**

– **t = (11, 22)**

**t[1]**

...

## Otros agregados

- No se pueden indexar conjuntos
  - `mi_set = {'Pera', 'Manzana', 'Naranjas'}`  
`mi_set[ 0 ]` está **ERRADO**
- Solo se pueden indexar diccionarios a través de sus claves.
  - `edad = {'Bob': 29, 'Carola': 23, 'Alicia': 26}`  
`edad[ 'Alicia' ]` funciona  
`edad[0]` produce un **ERROR**