

Diccionarios

Fundamentos de Programación

FIEC04341

Sesión 01

Diccionarios (Hash)

- Los diccionarios permiten una asociación entre una etiqueta (clave) y un valor. En ocasiones entre dos piezas de información.
- Cada elemento de un diccionario es un par clave:valor

```
dict = { 'Name' : 'Sara' , 'Age' : 7 , 'Peso' : 120 }
```

Sara	7	120
Name	Age	Peso

Acceder a un par (clave: valor)

- No se pueden modificar las claves pero si los valores asociados a las mismas.
- Las claves y los valores pueden ser de cualquier tipo.

```
>>>dict = { 'Name' : 'Sara' , 'Age' : 7 , 'Peso' : 120.0 }
>>>print(dict['Name'])
Sara
>>>print(dict['Age'])
7
>>>print(type(dict['Peso']))
<class 'float'>
```

Indexar un Diccionario

- Si la clave no existe se generará un error.

```
>>> print(type(dict['Peso']))
<class 'float'>
>>> dict['LastName']
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
KeyError: 'LastName'
>>> dict[0]
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
KeyError: 0
```

Actualizar un diccionario

- Se puede actualizar añadiendo un nuevo elemento (par clave, valor), modificando una entrada o borrando una entrada

```
>>>dict = {'Name': 'Sara', 'Age': 7, 'Peso': 120.0}
>>>dict['Age']=8 #actualiza una entrada
>>>dict['LastName']='Alzola' #añade una entrada
>>>print(dict)
{'Peso': 120.0, 'Age': 8, 'Name': 'Sara', 'LastName': 'Alzola'}
>>>del dict['LastName'] #borra una entrada
>>>print(dict)
{'Peso': 120.0, 'Age': 8, 'Name': 'Sara'}
```

Diccionarios como listas

```
>>>dict = { 'Name': 'Sara', 'Age': 7, 'Peso': 120.0 }
>>>'Age' in dict
True
>>>len(dict)
3
>>>print(dict.keys())
dict_keys(['Peso', 'Age', 'Name'])
>>>print(dict.values())
dict_values([120.0, 8, 'Sara'])
>>>a=list(dict.keys())
>>>b=list(dict.values())
>>>a
['Peso', 'Age', 'Name']
>>>b
[120.0, 8, 'Sara']
```

Iterar con diccionarios

```
dict = {'Name': 'Sara', 'Age': 7, 'Peso': 120.0}
#iteramos por las claves del dict
for clave in dict:
    print(clave)

#iterar por los valores del dict
for val in dict.values():
    print(val)

#iterar por clave y acceder al valor
for clave in dict:
    print(clave, 'corresponde a: ', dict[clave])

#iterar por el par (clave,valor) (tupla)
for clave, valor in dict.items():
    print(clave, 'corresponde a: ', valor)
```

Características de los diccionarios

- Un diccionario no puede contener múltiples elementos con la misma clave.
- Una clave es inmutable. Puede ser una cadena, un número o una tupla.
- Los valores no tienen que ser únicos.
- Un diccionario es optimizado para insertar y buscar elementos rápidamente
- Un diccionario no tiene un método interno `sort()` como la lista, pero Python provee de otro método llamado `sorted()`

Ordenar un diccionario

```
>>>dict = {8.3: 'Jorge', 6.5: 'Lucia', 7.8: 'Sophia'}
>>>sorted(dict)
[6.5, 7.8, 8.3]
>>>sorted(dict, reverse=True)
[8.3, 7.8, 6.5]
>>>sorted(dict.keys(), reverse=True)
```

Problema: ordenar los scores

- Se requiere los estudiantes con los 3 mejores scores de la lista

VALOR	CLAVE
Estudiante	Score
Joshua	8.65
Juan	9.12
Lisette	8.45
Johanna	7.81
María	8.05
Jorge	7.21
Anita	8.31

Estudiante	Score
Joshua	8.65
Juan	9.12
Lisette	8.45
Johanna	7.81
María	8.05
Jorge	7.21
Anita	8.31

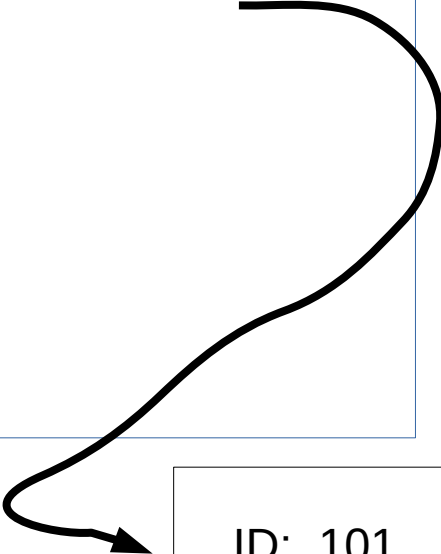
resultados.csv

Datos más complejos

```
101;Johnny Jones;USA;8.32;Fish;21
102;Juan Martino;Spain;9.01;Gun;36
103;Joseph Smyth;USA;8.85;Cruiser;18
104;Stacey O'Neill;Ireland;8.91;Malibu;22
105;Aideen Wu;Japan;8.65;Fish;24
106;Zack MacFadden;Scotland;7.82;Thruster;26
107;Aaron Valentino;Italy;8.98;Gun;19
```

Conjunto de líneas de datos

Salida en pantalla



```
ID: 101
Name: Johnny 'wave-boy' Jones
Country: USA
Average: 8.32
Board type: Fish
Age: 21
```