Diccionarios

Algoritmo de construcción de diccionarios a partir de un contenedor (Esquema 1):

Los pasos son los siguientes:

1. *Se va recorriendo el contenedor (con un for)*
2. *Se consulta si el elemento que será la clave ya está en el diccionario (con un if … not in...)*
   a) *Si no está se inserta una nueva pareja clave-valor : diccionario[clave]=valor*
   b) *Si está se actualiza el valor: diccionario[clave]=nuevo_valor*

Ejemplo : Para la lista=['a','b','c','a','b','b','z','a','b','c'], contar cuantas veces aparece cada valor

```
dic=dict()
for letra in  lista :
    if letra not in dic:
        dic[letra]=1
    else:
        dic[letra]+=1          //esto es lo mismo que dic[letra]= dic[letra]+1
```

Resultado: dic → { 'a': 3, 'b': 4, 'c': 2, 'z': 1}

Diccionarios

Algoritmo de construcción de diccionarios a partir de un contenedor (Esquema 2):

Los pasos son los siguientes:

1. *Se va recorriendo el contenedor (con un for)*
2. *Se consulta si el elemento que será la clave ya está en el diccionario (con un if)*
   Si no está se inserta una nueva pareja clave-valor con el valor neutro de la operación
   diccionario[clave]=valor_neutro
3. *En todo caso, se actualiza el valor: diccionario[clave]=actualización del valor*

Ejemplo : Para la lista lista=['a','b','c','a','b','b','z','a','b','c'], contar cuantas veces aparece cada valor

```
dic=dict()
for letra in  lista :
    if letra not in dic:
        dic[letra]=0          //Elemento neutro de la suma
    dic[letra]+=1
```

Resultado: dic →{ 'a': 3, 'b': 4, 'c': 2, 'z': 1}

Diccionarios

Algoritmo de construcción de diccionarios a partir de un contenedor (Esquema 1):

Ejemplo:

Par=namedtuple("pareja", "letra, número")
lista=[("z",21),("a",62),("J",7),("b",56),("c",90),("z",21),("a",1),("b",10),("b",2),("a",21)], obtener un diccionario que a cada letra le haga corresponder una lista con los valores asociados

```
dic=dict()
for p in lista:
    if p.letra not in dic:
        dic[p.letra]=[p.número]
    else:
        dic[p.letra].append(p.número)          // esto es lo mismo que dic[p.letra]+=[p.número]
```

Resultado: dic →{'z': [21, 21], 'a': [62, 1, 21], 'J': [7], 'b': [56, 10, 2], 'c': [90]}

Diccionarios

Algoritmo de construcción de diccionarios a partir de un contenedor (Esquema 2):

Ejemplo:

Par=namedtuple("pareja", "letra, número")

lista=[("z",21),("a",62),("J",7),("b",56),("c",90),("z",21),("a",1),("b",10),("b",2),("a",21)], obtener un diccionario que a cada letra le haga corresponder una lista con los valores asociados

```
dic=dict()
for p in lista:
    if p.letra not in dic:
        dic[p.letra]=list()                  // o bien =[ ].Valor neutro de listas (una lista vacía)
    dic[p.letra].append(p.número)    // esto es lo mismo que dic[p.letra]+=[p.número]
```

Resultado: dic →{'z': [21, 21], 'a': [62, 1, 21], 'J': [7], 'b': [56, 10, 2], 'c': [90]}

Diccionarios

Esquema para la construcción de diccionarios "complejos", cuyos valores son: un máximo, un mínimo, una suma, un promedio, una lista ordenada, con un filtro sobre un contenedor y en general que necesiten de un "cálculo o similar"

Hemos aprendido a construir diccionario en el que los *valores* son contadores o una lista o conjunto, pero ahora se plantea una *operación adicional* sobre los valores.

Estos ejercicios se resuelven generalmente, de una forma eficiente, en dos pasos:

1. Construir *un primer diccionario* en el que los *valores* sean listas o conjuntos. *(ver las diapositivas anteriores)*
2. Construir *un segundo diccionario* a partir del diccionario del punto anterior que será recorrido con *"items()"*.
   - Las *claves* del *segundo diccionario* serán las del primero (no hay que comprobar si ya está o no en el segundo diccionario)
   - Los *valores* se obtienen realizando la *operación adicional* de que se trate.

Diccionarios

Ejemplo (diccionario "complejo"):

Supongamos una lista con tuplas con datos de estudiantes con su edad y un equipo al que pertenecen.

estudiantes=

[(Ismael,19,E4), (Ruben,18,E2), (Lorena,20,E2), (Rocío,18,E1), (M.Mar,19,E1), (David,18,E3), (Mario,20,E3), (Daniel,20,E2), (Javier,17,E1), (Daniel,19,E4), (Javier,18,E1), (Adrián,18,E4), (Javier,21,E2), (Celia,22,E1), (David,23,E3), (Mario,19,E4), (Rocío,18,E4), (Javier,19,E1), (Carlos,20,E2), (Guillermo,20,E2), (José,20,E3), (Luis,19,E1), (Javier,21,E4), (Fernando,20,E1), (Pedro,18,E3), (Ana,20,E1), (Manuel,18,E4), (Gonzalo,17,E3)]

Cada tupla se ha creado con: Estudiante=namedtuple ("estudiante","nombre, edad, equipo")

Enunciado: se pide un diccionario que, a cada equipo le haga corresponder los tres estudiantes de mayor edad. De cada estudiante se quiere conocer el nombre y la edad.

Si hay menos de tres, los que haya y si empatan, cualquiera de ellos.

Diccionarios

Paso 1. *Construcción del primer diccionario*: "a cada equipo le hacemos corresponder una lista con los estudiantes de dicho equipo". Como se quiere el nombre y la edad, los valores estarán formados por tuplas con dichos elementos.

Esquema 1

```
dic_aux=dict()
for e in estudiantes:
  if e.equipo not in dic_aux:
    dic_aux[e.equipo ]=[(e.nombre, e.edad)]
  else:
    dic_ aux[e.equipo append((e.nombre, e.edad))
```

Esquema 2

```
dic_aux=dict()
for e in estudiantes:
    if e.equipo not in dic_aux:
        dic_aux[e.equipo ]=list()
    dic_ aux[e.equipo ].append((e.nombre, e.edad))
```

El resultado es de este primer diccionario es:

{'E4': [('Ismael', 19), ('Daniel', 19), ('Adrián', 18), ('Mario', 19), ('Rocío', 18), ('Javier', 21), ('Manuel', 18)],
'E2': [('Ruben', 18), ('Lorena', 20), ('Daniel', 20), ('Javier', 21), ('Carlos', 20), ('Guillermo', 20)],
'E1': [('Rocío', 18), ('M.Mar', 19), ('Javier', 17), ('Javier', 18), ('Celia', 22), ('Javier', 19), **('Luis', 19)**,
('Fernando', 20), ('Ana', 20)],
'E3': [('David', 18), ('Mario', 20), ('David', 23), ('José', 20), ('Pedro', 18), ('Gonzalo', 17)] }

Diccionarios

Primer diccionario (*se visualiza en esta dispositiva para mejor comprensión del paso 2*)

{*'E4'*: [*('Ismael', 19)*, *('Daniel', 19*), *('Adrián', 18)*, *('Mario', 19)*, *('Rocío', 18)*, *('Javier', 21)*, *('Manuel', 18)*],
'E2': [*('Ruben', 18)*, *('Lorena', 20)*, *('Daniel', 20)*, *('Javier', 21)*, *('Carlos', 20)*, *('Guillermo', 20)*],
'E1': [*('Rocío', 18)*, *('M.Mar', 19)*, *('Javier', 17)*, *('Javier', 18)*, *('Celia', 22)*, *('Javier', 19)*, ***('Luis', 19)***,
('Fernando', 20), *('Ana', 20)*],
'E3': [*('David', 18)*, *('Mario', 20)*, *('David', 23)*, *('José', 20)*, *('Pedro', 18)*, *('Gonzalo', 17)*] }

Paso 2. *Construcción del segundo diccionario*: Mantenido las mismas *claves* realizamos la *operación adicional* *(los 3 de mayor edad)* sobre los *valores* de dic_aux.

```
res=dict()
for clave, valor in dic_aux.items():
    res[clave]=sorted (valor, key=lambda e:e[1], reverse=True)[:3]
```

El resultado del ejercicio es :

res→ {*'E4'*: [*('Javier', 21)*, ('Ismael', 19), *('Daniel', 19)*], *'E2'*: [*('Javier', 21)*, *('Lorena', 20)*, *('Daniel', 20)*],
'E1': [*('Celia', 22)*, *('Fernando', 20)*, *('Ana', 20)*], **'E3'**: [*('David', 23)*, *('Mario', 20)*, *('José', 20)*]}

Diccionarios

Variante1 del ejercicio: Supongamos que en vez de nombre y edad sólo quisieran el nombre.

Es claro que, en el primer diccionario, los valores deben tener el nombre y la edad, para poder ordenar por esta última. En este caso, nos apoyamos en una *lista auxiliar*, de la que escogeremos el nombre, con el siguiente esquema:

```
res=dict()
for clave, valor in dic_aux.items():
    list_aux=sorted(valor, key=lambda e:e[1], reverse=True)[:3]
    res[clave]=[nombre for nombre, edad in list_aux]
```

El resultado del ejercicio es :

res→{'E4': ['Javier', 'Ismael', 'Daniel'], 'E2': ['Javier', 'Lorena', 'Daniel'], 'E1': ['Celia', 'Fernando','Ana'], 'E3': ['David', 'Mario', 'José']}

Diccionarios

Variante2 del ejercicio: Supongamos además que los nombres, lo quieren en *orden alfabético*

En este caso bastaría con ordenar la lista de *valores*.

```
res=dict()
for clave, valor in dic_aux.items():
    list_aux=sorted(valor, key=lambda e:e[1], reverse=True)[:3]
    res[clave]=sorted([nombre for nombre, edad in list_aux])
```

El resultado del ejercicio es :

res→{*'E4'*: [*'Daniel'*, *'Ismael'*, *'Javier'*], *'E2'*: [*'Daniel'*, *'Javier'*, *'Lorena'*], **'E1'**: [*'Ana'*, *'Celia'*, *'Fernando'*], **'E3'**: [*'David'*, *'José'* , *'Mario'*]}