

# Diccionarios: Resolución mediante tuplas como clave. Listas y conjuntos por comprensión

Fundamentos de Programación  
Departamento de Lenguajes y Sistemas Informáticos



## Diccionarios (Esquema para resolución de problemas mediante el uso de tupla como clave)

Hemos visto en la lección anterior como construir **diccionario (res)** como resultado de recorrer un **diccionario previo (aux)**, en el que los valores son listas y los valores de **res** son el resultado de alguna operación (*máximo, ordenación, mínimo, promedio, ...*) a los valores de **aux**

Ahora se trata de *que la operación adicional no es tan inmediata como las descritas, por ejemplo, una **nueva agrupación** por otro criterio.*

Supongamos que nuestro tipo Estudiantes tiene un campo más, **el grado** que cursan los estudiantes.

Estudiante=NamedTuple("estudiante", [("grado",str),("nombre",str),("edad",int),("curso",str)])

```
estudiantes=[('TI3', 'Ismael', 19, 'C4'), ('TI4', 'Ruben', 18, 'C2'), ('TI3', 'Lorena', 20, 'C2'), ('TI4', 'Rocío', 18, 'C1'), ('IS2', 'M.Mar', 19, 'C1'), ('IS3', 'David', 18, 'C3'), ('IS2', 'Mario', 20, 'C3'), ('IS2', 'Daniel', 20, 'C2'), ('IS3', 'Javier', 17, 'C1'), ('TI3', 'Daniel', 19, 'C4'), ('IS3', 'Javier', 18, 'C1'), ('TI3', 'Adrián', 18, 'C4'), ('IS2', 'Javier', 21, 'C2'), ('IS3', 'Celia', 22, 'C1'), ('TI3', 'David', 23, 'C3'), ('IS2', 'Mario', 19, 'C4'), ('IS3', 'Rocío', 18, 'C4'), ('TI3', 'Javier', 19, 'C1'), ('TI3', 'Carlos', 20, 'C2'), ('IS3', 'Guillermo', 20, 'C2'), ('IS3', 'José', 20, 'C3'), ('TI4', 'Luis', 19, 'C1'), ('TI3', 'Javier', 21, 'C4'), ('TI3', 'Fernando', 20, 'C1'), ('TI3', 'Pedro', 18, 'C3'), ('TI4', 'Ana', 20, 'C1'), ('IS2', 'Manuel', 18, 'C4')]
```

**Ejercicio:** Obtener un diccionario que a cada **grado** le asocie una lista de tuplas con cada curso y el promedio de edades de dicho curso



## Diccionarios (Esquema para resolución de problemas mediante el uso de tupla como clave)

*Ejercicio: Obtener un diccionario que a cada **grado** le asocie una lista de tuplas con cada curso y el promedio de edades de dicho curso*

Datos de entrada

```
estudiantes=[('TI3', 'Ismael', 19, 'C4'), ('TI4', 'Ruben', 18, 'C2'), ('TI3', 'Lorena', 20, 'C2'), ('TI4', 'Rocío', 18, 'C1'), ('IS2', 'M.Mar', 19, 'C1'), ('IS3', 'David', 18, 'C3'), ('IS2', 'Mario', 20, 'C3'), ('IS2', 'Daniel', 20, 'C2'), ('IS3', 'Javier', 17, 'C1'), ('TI3', 'Daniel', 19, 'C4'), ('IS3', 'Javier', 18, 'C1'), ('TI3', 'Adrián', 18, 'C4'), ('IS2', 'Javier', 21, 'C2'), ('IS3', 'Celia', 22, 'C1'), ('TI3', 'David', 23, 'C3'), ('IS2', 'Mario', 19, 'C4'), ('IS3', 'Rocío', 18, 'C4'), ('TI3', 'Javier', 19, 'C1'), ('TI3', 'Carlos', 20, 'C2'), ('IS3', 'Guillermo', 20, 'C2'), ('IS3', 'José', 20, 'C3'), ('TI4', 'Luis', 19, 'C1'), ('TI3', 'Javier', 21, 'C4'), ('TI3', 'Fernando', 20, 'C1'), ('TI3', 'Pedro', 18, 'C3'), ('TI4', 'Ana', 20, 'C1'), ('IS2', 'Manuel', 18, 'C4')]
```

- 
- 1) un diccionario que *agrupe por grado y curso*
  - 2) Obtener la lista a partir el diccionario

Datos de salida

```
{'TI3': [('C4', 19.25), ('C2', 20.0), ('C3', 20.5), ('C1', 19.5)],  
'TI4': [('C2', 18.0), ('C1', 19.0)],  
'IS2': [('C1', 19.0), ('C3', 20.0), ('C2', 20.5), ('C4', 18.5)],  
'IS3': [('C3', 19.0), ('C1', 19.0), ('C4', 18.0), ('C2', 20.0)]}
```



## Diccionarios (Esquema para resolución de problemas mediante el uso de dos diccionarios y función auxiliar)

*Obtener un diccionario que a cada **grado** le asocie una lista de tuplas con cada curso y el promedio de edades de dicho curso*

```
aux1=dict()
for e in estudiantes:
    if (e.grado,e.curso) not in aux1:
        aux1[(e.grado,e.curso)]=list()
        aux1[(e.grado,e.curso)].append(e.edad)
```

```
{('TI3', 'C4'): [19, 19, 18, 21], ('TI4', 'C2'): [18], ('TI3', 'C2'): [20, 20], ('TI4', 'C1'): [18, 19, 20], ('IS2', 'C1'): [19], ('IS3', 'C3'): [18, 20], ('IS2', 'C3'): [20], ('IS2', 'C2'): [20, 21], ('IS3', 'C1'): [17, 18, 22], ('TI3', 'C3'): [23, 18], ('IS2', 'C4'): [19, 18], ('IS3', 'C4'): [18], ('TI3', 'C1'): [19, 20], ('IS3', 'C2'): [20]}
```

```
aux2=dict()
for c,v in aux1.items():
    aux2[c]=sum(v)/len(v)
```

```
{('TI3', 'C4'): 19.5, ('TI4', 'C2'): 18.0, ('TI3', 'C2'): 20.0, ('TI4', 'C1'): 19.0, ('IS2', 'C1'): 19.0, ('IS3', 'C3'): 19.0, ('IS2', 'C3'): 20.0, ('IS2', 'C2'): 20.5, ('IS3', 'C1'): 19.0, ('TI3', 'C3'): 20.5, ('IS2', 'C4'): 19.5, ('IS3', 'C4'): 18.0, ('TI3', 'C1'): 19.5, ('IS3', 'C2'): 20.0}
```

```
res=dict()
for c,v in aux2.items():
    if c[0] not in res:
        res[c[0]]=list()
        res[c[0]].append((c[1],v))
return res
```

```
{'TI3': [('C4', 19.25), ('C2', 20.0), ('C3', 20.5), ('C1', 19.5)],
'TI4': [('C2', 18.0), ('C1', 19.0)],
'IS2': [('C1', 19.0), ('C3', 20.0), ('C2', 20.5), ('C4', 18.5)],
'IS3': [('C3', 19.0), ('C1', 19.0), ('C4', 18.0), ('C2', 20.0)]}
```



## Generación de listas por comprensión

```
Par=NamedTuple('pareja',[(número,int),(Letra,str)])
```

```
lista=[Par(2,'a'),Par(5,'z'),Par(4,'c'),Par(9,'b'),Par(4,'c'),Par(6,'w'),Par(1,'t')]
```

Para obtener una nueva lista con sólo las letras hemos trabajado el siguiente esquema:

```
lista_letras=list()
for elto in lista:
    lista_letras.append(elto.lettra)
```

*Resultado:* lista\_letras → ['a','z','c','b','c','w','t']

Esto mismo se puede *hacer por comprensión* con la siguiente sintaxis.

```
lista_letras=[elto.lettra for elto in lista]
```



## Generación de listas por comprensión

```
Par=NamedTuple('pareja',[(número,int),(Letra,str)])
```

```
lista=[Par(2,'a'),Par(5,'z'),Par(4,'c'),Par(9,'b'),Par(4,'c'),Par(6,'w'),Par(1,'t')]
```

Para obtener una nueva lista con las tuplas de las que el número es par, el esquema sería:

```
lista_filtrada=list()
for elto in lista:
    if elto.número%2==0:
        lista_filtrada.append(elto)
```

*Resultado:* lista\_filtrada → [(2,'a'),(4,'c'),(4,'c'),(6,'w')]

Esto mismo se puede *hacer por comprensión* con la siguiente sintaxis.

```
lista_filtrada =[elto for elto in lista if elto.número%2==0]
```



## Generación de conjuntos por comprensión

```
Par=NamedTuple('pareja',[(número,int),(Letra,str)])
```

```
conjunto={Par(2,'a'),Par(5,'z'),Par(4,'c'),Par(9,'b'),Par(6,'w'),Par(1,'t')}
```

Para obtener un nuevo conjunto con sólo los números hemos trabajado el siguiente esquema:

```
conjunto_números=set()  
for elto in conjunto:  
    conjunto_números.add(elto.número)
```

*Resultado:* conjunto\_números → {1, 2, 4, 5, 6, 9}

Esto mismo se puede *hacer por comprensión* con la siguiente sintaxis.

```
conjunto_números={elto.número for elto in conjunto}
```



## Generación de conjuntos por comprensión

```
Par=NamedTuple('pareja',[(número,int),(Letra,str)])
```

```
conjunto={Par(2,'a'),Par(5,'z'),Par(4,'c'),Par(9,'b'),Par(6,'w'),Par(1,'t')}
```

Para obtener un nuevo conjunto de tuplas con letras mayores que la 'm', el esquema habitual es:

```
conjunto_filtrado=set()
for elto in conjunto:
    if elto.lettra>'m':
        conjunto_filtrado.add(elto)
```

*Resultado:* conjunto\_filtrado  $\rightarrow$  {(6,'w'),(5,'z'),(1,'t')}

Esto mismo se puede *hacer por comprensión* con la siguiente sintaxis.

```
conjunto_filtrado={elto for elto in conjunto if elto.lettra>'m'}
```





## Ejercicio:

*Proyecto T11\_Olimpiadas:*

- Descargar e Implementar el proyecto