

Estructuras de datos anidadas

Programación

<http://progra.usm.cl>

UTFSM



Departamento de Informática
Universidad Técnica Federico Santa María

Repaso

Listas:

- tienen orden
- mutables
- `list()`
- `[1, 2, 3]`

Diccionarios:

- no tienen orden
- pares llave-valor
- llaves no repetidas
- mutables
- `dict()`
- `{1: 2, 3: 4}`

Tuplas:

- tienen orden
- inmutables
- `tuple()`
- `(1, 2, 3)`

Conjuntos:

- no tienen orden
- elementos no repetidos
- mutables
- `set()`
- `{1, 2, 3}`

Problema 1

Los datos de los alumnos de una universidad están almacenados en un diccionario.

Las llaves son los roles, y los valores son tuplas con el nombre, el apellido y la fecha de nacimiento:

```
alumnos = {  
    '201599001-3': ('Esteban', 'Aguila', (1994, 10, 5)),  
    '201599003-k': ('Monica', 'Barrios', (1993, 11, 27)),  
    '201599004-8': ('Victor', 'Delgado', (1994, 3, 18)),  
    '201499077-k': ('Felipe', 'Galdames', (1995, 2, 12)),  
    '201588501-5': ('Romina', 'Munoz', (1996, 5, 25)),  
    '201588502-3': ('Jean', 'Pineda', (1994, 8, 23)),  
    '201588503-1': ('Cinthia', 'Sotelo', (1994, 10, 9)),  
    '201488510-0': ('Amparo', 'Zambrano', (1995, 11, 22)),  
}
```

Problema 1 (continuación)

Los inscritos de cada ramo están registrados en otro diccionario. Las claves son los nombres de los ramos, y cada valor es el conjunto de los roles de los alumnos que están tomando el ramo:

```
inscritos = {  
    'Progra': {'201599003-k', '201599004-8', '201588503-1',  
              '201488510-0'},  
    'Mate':  {'201599001-3', '201599003-k', '201599004-8',  
              '201499077-k'},  
    'Fisica': {'201599001-3', '201599003-k', '201499077-k',  
              '201588502-3', '201488510-0'},  
    'Quimica': {'201599001-3', '201599004-8', '201588502-3',  
               '201588503-1', '201488510-0'},  
}
```

Problema 1 (continuación)

Ejercicio 1

Escriba la función `ramos_alumno(rol)` que retorne la lista de los ramos que está tomando el alumno con el rol entregado como parámetro:

```
>>> ramos_alumno('201588503-1')
['Quimica', 'Progra']
>>> ramos_alumno('201599003-k')
['Mate', 'Progra', 'Fisica']
```

Problema 1 (continuación)

Ejercicio 2

Escriba la función `alumno_mas_joven()` que retorne el nombre completo del alumno más joven:

```
>>> alumno_mas_joven()  
'Romina Munoz'
```

Problema 2

Los resultados de un campeonato de fútbol están almacenados en un diccionario:

```
resultados = {  
    ('Honduras', 'Chile'): (0, 1),  
    ('Espana', 'Suiza'): (0, 1),  
    ('Suiza', 'Chile'): (0, 1),  
    ('Espana', 'Honduras'): (3, 0),  
    ('Suiza', 'Honduras'): (0, 0),  
    ('Espana', 'Chile'): (2, 1),  
}
```

Problema 2 (continuación)

Ejercicio 1

Escriba la función `obtener_lista Equipos` que retorne la lista de los equipos que participan en el campeonato:

```
>>> obtener_lista Equipos()  
['Honduras', 'Suiza', 'Espana', 'Chile']
```


Problema 2 (continuación)

Ejercicio 2

El equipo que gana un partido, recibe 3 puntos.

En caso de empate, ambos equipos reciben 1 punto.

Escriba la función `calcular_puntos(equipo)` que retorne la cantidad de puntos obtenidos por el equipo pasado como parámetro:

```
>>> calcular_puntos('Chile')
6
>>> calcular_puntos('Suiza')
4
```

Problema 2 (continuación)

Ejercicio 3

La *diferencia de goles* de un equipo es el total de los goles que anotó menos el total de goles que recibió.

Escriba la función `calcular_diferencia(equipo)` que retorne la diferencia de goles del equipo pasado como parámetro:

```
>>> calcular_diferencia('Chile')
1
>>> calcular_diferencia('Honduras')
-4
```

Problema 3

La asistencia de los alumnos a clases puede ser llevada en una tabla como ésta:

Clase	1	2	3	4	5	6	7
Pepito	✓	✓	✓				
Yayita	✓	✓	✓		✓		✓
Fulanita	✓	✓	✓	✓	✓	✓	✓
Panchito	✓	✓	✓		✓	✓	✓

La tabla puede ser representada como una lista de listas con valores booleanos:

```
alumnos = ['Pepito', 'Yayita', 'Fulanita', 'Panchito']
asistencia = [
    [True, True, True, False, False, False, False],
    [True, True, True, False, True, False, True ],
    [True, True, True, True, True, True, True ],
    [True, True, True, False, True, True, True ]
]
```

Problema 3 (continuación)

Ejercicio 1

Escriba la función `total_por_alumno(tabla)` que reciba como parámetro la tabla de asistencia y retorne una lista con el número de clases asistidas por cada alumno:

```
>>> total_por_alumno(asistencia)
[3, 5, 7, 6]
```

Problema 3 (continuación)

Ejercicio 2

Escriba la función `total_por_clase(tabla)` que reciba como parámetro la tabla de asistencia y retorne una lista con el número de alumnos que asistió a cada clase:

```
>>> total_por_clase(asistencia)
[4, 4, 4, 1, 3, 2, 3]
```

Problema 3 (continuación)

Ejercicio 3

Escriba la función `alumno_estrella(tabla)` que reciba como parámetro la tabla de asistencia y retorne el nombre del alumno que asistió a más clases:

```
>>> alumno_estrella(asistencia)
'Fulanita'
```