

22 de mayo de 2024

INGENIERIA EN DESARROLLO DE SOFTWARE

Materia: FUNDAMENTOS DE PROGRAMACION II

1.4 Actividad. Colecciones

Elaborado por: Jesús Angel Hernández
Martínez

LISTAS

[1]: *#Característica Ordenada se refiere a que cada elemento tiene una posición de memoria que puede ser accedido a través de un índice*
#Al poseer esta característica es posible utilizar el método sorted() para ordenar sus elementos

```
#Ejemplo ordenación de Lista
lista_desordenada = [8,6,3,15,1,2,20,4,5]
lista_ordenada = sorted(lista_desordenada)
print(lista_desordenada)
print(lista_ordenada)
#Al usar el método sorted de las listas podemos ordenar la colección
alfa_desordenado = ["L","F","A","X","R","B"]
alfa_ordenado = sorted(alfa_desordenado)
print(alfa_desordenado)
print(alfa_ordenado)
```

```
[8, 6, 3, 15, 1, 2, 20, 4, 5]
[1, 2, 3, 4, 5, 6, 8, 15, 20]
['L', 'F', 'A', 'X', 'R', 'B']
['A', 'B', 'F', 'L', 'R', 'X']
```

[2]: *#Las listas son colecciones que pueden contener cualquier objeto*

```
coleccion_multiple = [1,"A",True,2/5,(5,15),{"usuario": "Jesus", "clave":"swordfish"},complex(5,15)]
print(coleccion_multiple)
for x in coleccion_multiple:
    print(f'El elemento {x} es un objeto de tipo: {type(x)}')
```

```
[1, 'A', True, 0.4, (5, 15), {'usuario': 'Jesus', 'clave': 'swordfish'}, (5+15j)]
El elemento 1 es un objeto de tipo: <class 'int'>
El elemento A es un objeto de tipo: <class 'str'>
El elemento True es un objeto de tipo: <class 'bool'>
El elemento 0.4 es un objeto de tipo: <class 'float'>
El elemento (5, 15) es un objeto de tipo: <class 'tuple'>
El elemento {'usuario': 'Jesus', 'clave': 'swordfish'} es un objeto de tipo: <class 'dict'>
El elemento (5+15j) es un objeto de tipo: <class 'complex'>
```

[3]: *#Las listas pueden ser anidadas sin límite*

```
#En este ejemplo tenemos una lista con 2 tetramestres
#cada tetramestre contiene dos elementos, el nombre del tetramestre y una lista con 3 materias
#cada materia es una lista con dos elementos, el nombre de la materia y su calificación
calificaciones = [
    [
        "Tetramestre 1",[
            ["Administracion",100],
            ["Programacion",100],
            ["Idioma Chino Mandarin",0]
        ]
    ],
    [
        "Tetramestre 2",[
            ["Economia",100],
            ["Bases de datos",100],
            ["Matematicas",100]
        ]
    ]
]

for tetra in calificaciones:
    print(tetra)
```

```
['Tetramestre 1', [['Administracion', 100], ['Programacion', 100], ['Idioma Chino Mandarin', 0]]]
['Tetramestre 2', [['Economia', 100], ['Bases de datos', 100], ['Matematicas', 100]]]
```

[4]: *#Las listas pueden ser accedidas con un índice, de ahí si característica de ordenadas.*

```
#Si deseo acceder a la calificación de chino mandarín debo identificar su índice
#Paso 1 - Como el elemento que busco se encuentre en el primer tetramestre, podemos acceder a él a través de calificaciones[0]
#Paso 2.- Como calificaciones[0] contiene dos elementos y la materia chino mandarín se encuentra dentro del segundo elemento,
#accedemos a través de calificaciones[0][1]
#Paso 3.- Como calificaciones[0][1] contiene 3 elementos, observamos que chino mandarín se encuentra en el tercer elemento,
#entonces accedemos a través de calificaciones[0][1][2]
#Paso 4.- Por último el elemento calificaciones[0][1][2] contiene dos elementos, el primero es el nombre de la materia y el segundo la calificación,
#entonces accedemos a la calificación a través de calificaciones[0][1][2][1]
print(calificaciones[0])# Paso 1
print(calificaciones[0][1]) #Paso 2
print(calificaciones[0][1][2]) #Paso 3
print(calificaciones[0][1][2][1]) #Paso 4 Calificación = 0
```

```
['Tetramestre 1', [['Administracion', 100], ['Programacion', 100], ['Idioma Chino Mandarin', 0]]]
[['Administracion', 100], ['Programacion', 100], ['Idioma Chino Mandarin', 0]]
['Idioma Chino Mandarin', 0]
0
```

[5]: *#La Mutabilidad de las listas nos permiten agregar, modificar o eliminar valores de esta colección.*
#Metodos append(), pop(), insert(), remove()

```
#Ejemplo 1 Agregar un nuevo tetramestre a nuestra lista de calificaciones con el metodo append()
for tetra in calificaciones:
    print(tetra)
```

```
['Tetramestre 1', [['Administracion', 100], ['Programacion', 100], ['Idioma Chino Mandarin', 0]]]
['Tetramestre 2', [['Economia', 100], ['Bases de datos', 100], ['Matematicas', 100]]]
```

[6]: `calificaciones.append(["Tetramestre 3"])`
`for tetra in calificaciones:`
 `print(tetra)`

```
['Tetramestre 1', [['Administracion', 100], ['Programacion', 100], ['Idioma Chino Mandarin', 0]]]
['Tetramestre 2', [['Economia', 100], ['Bases de datos', 100], ['Matematicas', 100]]]
['Tetramestre 3']
```

[7]: *#Ejemplo 2 con el mismo metodo agregare calificaciones al tetramestre 3(lista anidada)*
`calificaciones[2].append(["Contabilidad", 100],["FrontEnd",100])`
`for tetra in calificaciones:`
 `print(tetra)`

```
['Tetramestre 1', [['Administracion', 100], ['Programacion', 100], ['Idioma Chino Mandarin', 0]]]
['Tetramestre 2', [['Economia', 100], ['Bases de datos', 100], ['Matematicas', 100]]]
['Tetramestre 3', [['Contabilidad', 100], ['FrontEnd', 100]]]
```

[8]: *#Con el metodo pop() podemos eliminar el ultimo elemento de una lista*
#Ejemplo 3 supongamos que deseamos eliminar la calificación de 'FrontEnd' de la lista anidada
`calificaciones[2][1].pop()`
`for tetra in calificaciones:`
 `print(tetra)`

```
['Tetramestre 1', [['Administracion', 100], ['Programacion', 100], ['Idioma Chino Mandarin', 0]]]
['Tetramestre 2', [['Economia', 100], ['Bases de datos', 100], ['Matematicas', 100]]]
['Tetramestre 3', [['Contabilidad', 100]]]
```

- [9]: *#Ejemplo 4 deseamos cambiar la calificacion de 'Idioma Chino Mandarin' a 15*
`calificaciones[0][1][2][1] = 15`
`for tetra in calificaciones:`
 `print(tetra)`

`['Tetramestre 1', [['Administracion', 100], ['Programacion', 100], ['Idioma Chino Mandarin', 15]]]`
`['Tetramestre 2', [['Economia', 100], ['Bases de datos', 100], ['Matematicas', 100]]]`
`['Tetramestre 3', [['Contabilidad', 100]]]`
- [10]: *#Ejemplo 5 nos informaron que agregaramos el tetramestre 0 para aquellos alumnos que tomaran cursos #de induccion y preparacion, el metodo append() solo nos permite agregar al final de una lista #por lo que recurrimos al metodo insert() para agregar elementos en la posicion que deseamos.*
`calificaciones.insert(0, ['Tetramestre 0'])`
`for tetra in calificaciones:`
 `print(tetra)`

`['Tetramestre 0']`
`['Tetramestre 1', [['Administracion', 100], ['Programacion', 100], ['Idioma Chino Mandarin', 15]]]`
`['Tetramestre 2', [['Economia', 100], ['Bases de datos', 100], ['Matematicas', 100]]]`
`['Tetramestre 3', [['Contabilidad', 100]]]`
- [11]: *#Ejemplo 6 Observamos que escribimos incorrectamente la palabra tetramestre #vamos a modificarlo accediendo a su indice*
`calificaciones[0]=["Tetramestre 0"]`
`for tetra in calificaciones:`
 `print(tetra)`

`['Tetramestre 0']`
`['Tetramestre 1', [['Administracion', 100], ['Programacion', 100], ['Idioma Chino Mandarin', 15]]]`
`['Tetramestre 2', [['Economia', 100], ['Bases de datos', 100], ['Matematicas', 100]]]`
`['Tetramestre 3', [['Contabilidad', 100]]]`
- [12]: *#Ejemplo 7 Nos informan que eliminaran el tetramestre 0 debido a la poca afluencia de alumnos #el metodo pop() no es util ya que elimina el ultimo elemento de una lista #por lo que debemos recurrir al metodo remove()*
`calificaciones.remove(['Tetramestre 0'])`
`for tetra in calificaciones:`
 `print(tetra)`

`['Tetramestre 1', [['Administracion', 100], ['Programacion', 100], ['Idioma Chino Mandarin', 15]]]`
`['Tetramestre 2', [['Economia', 100], ['Bases de datos', 100], ['Matematicas', 100]]]`
`['Tetramestre 3', [['Contabilidad', 100]]]`
-

Bibliografia

Fernandez Montoro, A. (2013). *Python 3 al descubierto*. Mexico D.F: Alfaomega Grupo Editor S.A de C.V .

Lujan Castillo, J. D. (2020). *Aprende a programar con Python*. Ciudad de Mexico: Alfaomega Grupo Editor S.A de C.V.

Marzal Varo, A., Garcia Sevilla, P., & Gracia Luengo, I. (2016). *Introduccion a la programacion con Python 3*. Universitat Jaume I. Servei de Comunicacio.