

Fundamentos de Programación

TEMA: ARREGLOS DE DATOS 2

Semana 12

OBJETIVO DEL LABORATORIO

Construye arreglos bidimensionales para la manipulación de datos para casos de negocio.
Realiza ordenamiento de los elementos en un arreglo.

MARCO TEÓRICO

Así como podemos usar las listas unidimensionales, podemos usarlas como matriz; ósea bidimensionales. Nos permite con ello trabajar con tablas.

A su vez, los elementos de una lista; los podemos ordenar en base a algún criterio que tengamos.

Como fuente de información, se tiene la misma plataforma del Netacad. Tanto para realizar los ejemplos propuestos en la interfaz integrada, así como para repasar lo visto en clase.

RECURSOS

a. Hardware

- Pc Pentium IV a superior
- Conexión de red

b. Software

- Sistema Operativo Windows XP a superior
- Navegador Chrome o Firefox
- Edube Sandbox de Python desde Netacad

PROCEDIMIENTO

1. Listas bidimensionales:

Para hacer arrays bidimensionales en Python, se pueden hacer de dos formas, con tan solo lo que hemos visto hasta ahora.

La primera tal vez sea la más intuitiva ya que se asemeja a una de las formas de realizarse en otros lenguajes como cualquier programador verá, consistiendo simplemente en declarar una lista como se dice en Python dentro de otra lista.

?

```
1 cadena = [[1,2,3],[4,5,6]]
2 print (cadena[1])
```

Lo cual nos da como resultado:

[4, 5, 6]

Pero se puede hacer una lista introduciendo otras listas ya definidas, lo único que hay que hacer es definir las listas de las que se compone la lista final antes.

Por ejemplo supongamos que queremos hacer dos listas, una con nombres femeninos y otra con nombres masculinos y luego integrarla en otra lista, sería algo tan sencillo como esto.

?

```
1 nombres = ['Ana', 'Sonia', 'Helena']
2 nombres2 = ['Miguel', 'Carlos', 'Juan']
3 nombresFinales = [nombres, nombres2]
4 print ('Imprimiendo valor [1][1]', nombresFinales [1][1])
```



4

En el ejemplo vemos como se crea la lista `nombresFinales` que está compuesta por las dos listas anteriores, y que nos da como resultado:

Imprimiendo valor `[1][1]` Carlos

Bien ya hemos empezado a hacer las listas bidimensionales, pero ¿cómo las recorremos?, pues bien se debe de hacer con dos `for`, tal y como vemos en el siguiente ejemplo:

?

```
1 nombres = ['Ana', 'Sonia', 'Helena']
2 nombres2 = ['Miguel', 'Carlos', 'Juan']
3 nombresFinales = [nombres, nombres2]
4
5 for i in range(len(nombresFinales)):
6     print('\n')
7     for j in range(len(nombresFinales[i])):
8         print(nombresFinales[i][j], end=', ')
```

De esta forma se imprimen todos los nombres separados en dos líneas.

Ana, Sonia, Helena,
Miguel, Carlos, Juan

Esto es gracias a que cada vez que imprime un nombre hemos colocado la sentencia `end=', '` que le dice que después de cada impresión no haga una nueva línea, sino que una coma y que siga en la misma.

Es más, ya que estamos haciendo cosas divertidas, podríamos hacer que dentro del primer `for` se distinguiera si estamos recorriendo los nombres masculinos o femeninos para que la salida por consola nos informase de ello introduciendo tan solo un `if` y un `elif`:

?

```
1 nombres = ['Ana', 'Sonia', 'Helena']
2 nombres2 = ['Miguel', 'Carlos', 'Juan']
3 nombresFinales = [nombres, nombres2]
4
5 for i in range(len(nombresFinales)):
6     print('\n')
7     if i == 0:
8         print('##### NOMBRES FEMENINOS #####')
9     elif i == 1:
10        print('##### NOMBRES MASCULINOS #####')
11    for j in range(len(nombresFinales[i])):
12        print(nombresFinales[i][j], end=', ')
```

Dándonos como resultado:

NOMBRES FEMENINOS #####
Ana, Sonia, Helena,

NOMBRES MASCULINOS #####
Miguel, Carlos, Juan,



Como se puede ver son estructuras muy sencillas de trabajar y con grandes posibilidades, además de que para el tratamiento de datos son muy importantes.

CONCLUSIONES Y RECOMENDACIONES DE LA EXPERIENCIA

Si se puede trabajar con datos agrupados como lo son las listas, se puede tener estos datos en 2 dimensiones donde también deben tener la misma naturaleza.

De manera particular, se puede realizar el ordenamiento de los datos en base a un criterio que tengamos para ordenar los elementos.

Se recomienda usar el Edube SandBox para realizar nuestros ejercicios sin necesidad de instalar algún software adicional.

Así como en algún momento se solicita cálculos con números enteros, sería factible buscar realizar lo mismo; pero con números reales.

ACTIVIDAD VIRTUAL

1. Observa y analiza el siguiente video: <https://www.youtube.com/watch?v=lv3-aLH6SWo>, y responde las siguientes preguntas:
 - ¿Cuál es el arreglo o matriz que aparece en el video?
 - Realiza ese ejemplo en Python

