mcpp_taller3_camila_valencia

August 25, 2016

1 Taller 3

Métodos Computacionales para Políticas Públicas - URosario
Entrega: viernes 26-ago-2016 11:59 PM
[Camila Valencia] [camilavalenciar@gmail.com]

1.1 Instrucciones:

- Guarde una copia de este *Jupyter Notebook* en su computador, idealmente en una carpeta destinada al material del curso.
- Modifique el nombre del archivo del *notebook*, agregando al final un guión inferior y su nombre y apellido, separados estos últimos por otro guión inferior. Por ejemplo, mi *notebook* se llamaría: mcpp_taller3_santiago_matallana
- Marque el *notebook* con su nombre y e-mail en el bloque verde arriba. Reemplace el texto "[Su nombre acá]" con su nombre y apellido. Similar para su e-mail.
- Desarrolle la totalidad del taller sobre este *notebook*, insertando las celdas que sea necesario debajo de cada pregunta. Haga buen uso de las celdas para código y de las celdas tipo *markdown* según el caso.
- Recuerde salvar periódicamente sus avances.
- Cuando termine el taller:

    1. Descárguelo en PDF.
    2. Suba los dos archivos (.pdf y .ipynb) a su repositorio en GitHub antes de la fecha y hora límites.

(El valor de cada ejercicio está en corchetes [ ] después del número de ejercicio.)

Antes de iniciar, por favor descarge el archivo mcpp_taller3_listas_ejemplos.py del repositorio, guárdelo en la misma carpeta en la que está trabajando este taller y ejecútelo con el siguiente comando:

```
In [5]: run mcpp_taller3_listas_ejemplos.py
```

Este archivo contiene tres listas (l0, l1 y l2) que usará para las tareas de esta sección. Puede ver los valores de las listas simplemente escribiendo sus nombres y ejecutándolos en el Notebook. Inténtelo para verificar que mcpp_taller3_listas_ejemplos.py quedó bien cargado. Debería ver:

In [1]: l0 Out[1]: []
In [2]: l1 Out[2]: [1, 'abc', 5.7, [1, 3, 5]]
In [3]: l2 Out[3]: [10, 11, 12, 13, 14, 15, 16]

```
In [6]: l0

Out[6]: []

In [7]: l1

Out[7]: [1, 'abc', 5.7, [1, 3, 5]]

In [8]: l2

Out[8]: [10, 11, 12, 13, 14, 15, 16]
```

1.2 1. [1]

Cree una lista que contenga los elementos 7, "xyz" y 2.7.

```
In [9]: l4=[7,"xyz",2.7]

In [10]: l4

Out[10]: [7, 'xyz', 2.7]
```

1.3 2. [1]

Halle la longitud de la lista l1.

```
In [11]: len(l1)

Out[11]: 4
```

1.4 3. [1]

Escriba expresiones para obtener el valor 5.7 de la lista l1 y para obtener el valor 5 a partir del tercer elemento de l1.

```
In [12]: l1[2]

Out[12]: 5.7

In [13]: l1[3][2]

Out[13]: 5
```

1.5 4. [1]

Prediga qué ocurrirá si se evalúa la expresión l1[4] y luego pruébelo.

Prediciendo esto me va a generar un error ya que python empieza a contar en 0. Luego el primer elemento esta en la posicion 0, si le pido que me muestre el 4 posicion elemento me mostrara el 5to elemento. Como la lista solo tiene 4, me dira que hay un error porque no hay elemtos que mostrar

```
In [14]: l1[4]
```

```
---------------------------------------------------------------------------

IndexError                                Traceback (most recent call last)

<ipython-input-14-2c2a81dbcaa5> in <module>()
----> 1 l1[4]


IndexError: list index out of range
```

1.6 5. [1]

Prediga qué ocurrirá si se evalúa la expresión l2[-1] y luego pruébelo.

Mostrara el ultimo elemento de l2, es decir 16

```
In [15]: l2[-1]

Out[15]: 16
```

1.7 6. [1]

Escriba una expresión para cambiar el valor 3 en el tercer elemento de l1 a 15.0.

```
In [17]: l1[3][1]=15.0
         l1

Out[17]: [1, 'abc', 5.7, [1, 15.0, 5]]
```

1.8 7. [1]

Escriba una expresión para crear un "slice" que contenga del segundo al quinto elemento (inclusive) de la lista l2.

```
In [20]: l2[1:5]

Out[20]: [11, 12, 13, 14]
```

1.9 8. [1]

Escriba una expresión para crear un "slice" que contenga los primeros tres elementos de la lista l2.

```
In [21]: l2[:3]

Out[21]: [10, 11, 12]
```

1.10 9. [1]

Escriba una expresión para crear un "slice" que contenga del segundo al último elemento de la lista l2.

```
In [23]: l2[1:]

Out[23]: [11, 12, 13, 14, 15, 16]
```

1.11 10. [1]

Escriba un código para añadir cuatro elementos a la lista l0 usando la operación append y luego extraiga el tercer elemento (quítelo de la lista). ¿Cuántos "appends" debe hacer?

```
In [ ]: Ya que la lista es vacia tengo que hacer 4 appends

In [26]: import random
         for i in range(4):
             l0.append(random.randint(0,55))
             print (l0)

[9]
[9, 6]
[9, 6, 6]
[9, 6, 6, 18]


In [31]: del l0[2]
         l0

Out[31]: [9, 6, 18]
```

1.12 11. [1]

Cree una nueva lista nl concatenando la nueva versión de l0 con l1, y luego actualice un elemento cualquiera de nl. ¿Cambia alguna de las listas l0 o l1 al ejecutar los anteriores comandos?

```
In [34]: n1=l0+l1
         n1

Out[34]: [9, 6, 18, 1, 'abc', 5.7, [1, 15.0, 5]]
```

```
In [37]: n1[2]=5555
         n1

Out[37]: [9, 6, 5555, 1, 'abc', 5.7, [1, 15.0, 5]]

In [38]: l0

Out[38]: [9, 6, 18]

In [39]: l1

Out[39]: [1, 'abc', 5.7, [1, 15.0, 5]]
```

Al hacer este cambio no se alteran ninguna de las dos listas l0 o l1 ya que se actualizo el elemento el la lista n1

1.13 12. [2]

Escriba un loop que compute una variable all_pos cuyo valor sea True si todos los elementos de la lista l3 son positivos y False en otro caso.

```
In [46]: l3=[5,-8,33,-77,-9,55,3]

In [47]: all_pos = True
         for i in l3:
             if i<0:
                 all_pos=False
                 break
         print(all_pos)


False
```

1.14 13. [2]

Escriba un código para crear una nueva lista que contenga solo los valores positivos de la lista l3.

```
In [50]: pos=[]
         for i in l3:
             if i>0:
                 pos.append(i)


         print (pos)

[5, 33, 55, 3]
```

1.15 14. [2]

Escriba un código que use append para crear una nueva lista nl en la que el i-ésimo elemento de nl tiene el valor True si el i-ésimo elemento de l3 tiene un valor positivo y Falso en otro caso.

```
In [54]: n1=[]

         for i in l3:
             if i>0:
                 n1.append(True)
             if i<0:
                 n1.append(False)
         print(n1)
```

```
[True, False, True, False, False, True, True]
```

1.16 15. [3]

Escriba un código que use range, para crear una nueva lista nl en la que el i-ésimo elemento de nl es True si el i-ésimo elemento de l3 es positivo y Falso en otro caso.

Pista: Comience por crear una lista de longitud adecuada, con False en cada índice.

```
In [62]: n1=[False]*len(l3)
         n1
         for i in range(len(l3)):
             if l3[i]>0:
                 n1[i]=True

         n1
```

```
Out[62]: [True, False, True, False, False, True, True]
```

1.17 16. [4]

En clase construimos una lista con 10000 números aleatorios entre 0 y 9, a partir del siguiente código:

import random

N = 10000

random_numbers = [] for i in range(N): random_numbers.append(random.randint(0,9))

Y creamos un "contador" que calcula la frecuencia de ocurrencia de cada número del 0 al 9, así:

count = [] for x in range(0,10): count.append(random_numbers.count(x))

Cree un "contador" que haga lo mismo, pero sin hacer uso del método "count". (De hecho, sin usar método alguno.)

Pistas:

- Esto puede lograrse con un loop muy sencillo. Si su código es complejo, piense el problema de nuevo.

- Es muy útil iniciar con una lista "vacía" de 10 elementos. Es decir, una lista con 10 ceros.

```
In [9]: import random

        N = 10000
        random_numbers = []
        for i in range(N):
            random_numbers.append(random.randint(0,9))

        contar=[0]*10
        numeros=[0,1,2,3,4,5,6,7,8,9]
        for n in random_numbers:
            for i in numeros:
                if i==n:
                    contar[n]=contar[n]+1

        print(contar)
```

```
[1032, 991, 950, 1073, 954, 1030, 984, 947, 1008, 1031]
```

```
In [15]: count = []
         for x in range(0,10):
             count.append(random_numbers.count(x))
         print(count)
```

```
[1032, 991, 950, 1073, 954, 1030, 984, 947, 1008, 1031]
```