Ayudantía Programación

Listas y Ciclo For

Gonzalo Fernández

IWI-131 17/10/2017

Temas de la Ayudantía de Hoy

- Listas
- Ciclo For
- Ejercicios

Listas

- Arreglo ordenado de elementos, estos pueden ser:
  - Números: [1, 2, 34, 41, 1, 43]
  - Reales: [1.3, 2.444, 34.2, .4, 0.0]
  - Strings: [“Ohayou”, “Gozaimasu”]
  - Listas: [[1,2], [], [[[]]], [“ho”, [“la]]]
  - Y todo tipo de dato... [“asd”, 12, 2.31, [], { 5:3 }]

Funciones de Listas

- lista.append (elemento):
  - Agrega **elemento** al final de la lista
- lista.index (elemento):
  - Devuelve la posición de **elemento** en la lista
- lista.insert (posicion, elemento):
  - Inserta **elemento** en la posición **posición** de la lista

Funciones de Listas

- lista.sort ():
  - Ordena la lista de menor a mayor
- lista.reverse ():
  - Invierte el orden de los elementos de la lista
- range (inicio, fin, salto):
  - Devuelve una lista de números desde **inicio** hasta **fin – 1**, dejando una diferencia de **salto** entre cada número.

Ejemplos de range

```
1  print range(10)
2  print range(1, 10)
3  print range(1, 10, 2)
4  print "-----------------------------"
5  print range(10, 1)
6  print range(10, 1, 2)
7  print range(10, 1, -2)
8  print "-----------------------------"
9  print range(10, -1, -1)
```

```
[0, 1, 2, 3, 4, 5, 6, 7, 8, 9]
[1, 2, 3, 4, 5, 6, 7, 8, 9]
[1, 3, 5, 7, 9]
-----------------------------
[]
[]
[10, 8, 6, 4, 2]
-----------------------------
[10, 9, 8, 7, 6, 5, 4, 3, 2, 1, 0]
```

Iteración Sobre Listas

```
i = 0
while i < len(lista):
    print lista[i]
    i += 1
```

```
for i in lista:
    print i
```

Ejercicios

Ordenar de Mayor a Menor

Desarrolle la función **mayor_a_menor(lista)** que recibe una lista de números y retorna la misma lista ordenada de menor a mayor, según el siguiente ejemplo:

```
> print mayor_a_menor([1, 2, 65, 39, 213, 5, 20, 34])
 [213, 65, 39, 34, 20, 5, 2, 1]
```

Producto Interno

Desarrolle la función **producto_interno(a, b)** que recibe como parámetros 2 listas de números del mismo largo y retorna el valor del producto interno entre ellas, según el siguiente ejemplo:

```
> a = [1, 2, 3]
> b = [3, 2, 1]
> print producto_interno(a, b)
10
```

Coordenadas Satelitales

Jacinto está perdido en el bosque, su única esperanza es ser encontrado gracias al envío constante de su ubicación a la central de rescate Sansanitos S.A.

Esta es la lista de coordenadas que ha recibido la central hasta ahora, en formato [x, y]:

[[1, 2], [3, 2], [3, 1], [2, 2], [2, 3], [4, 2], [1, 2]]

Coordenadas Satelitales

Dada la inconsistencia de las coordenadas, la central determina que no es Jacinto quién se está moviendo si no que la interferencia de la zona no permite un envío correcto de los datos. Es por esto que la central te contrató a ti para desarrollar un programa capaz de calcular las **coordenadas promedio** en base a todas las coordenadas recibidas con las cuales poder salvar a Jacinto.

Coordenadas Satelitales

Desarrolle la función **coordenadas_promedio(lista)** que recibe una lista de listas de coordenadas [x, y] y retorna una única lista [x, y] con las coordenadas x e y promediadas en base a todas las coordenadas que la función recibió, según el siguiente ejemplo:

```
> coordenadas = [ [1, 2], [3, 2], [3, 1], [2, 2], [2, 3], [4, 2], [1, 2] ]
> coordenadas_promedio(coordenadas)
[2.2857142857142856, 2.0]
```

1. [20 %] Realice el ruteo de los siguientes programas e indique qué es lo que imprimen. Cada vez que el valor de una variable cambie, escríbalo en una nueva fila de la tabla. Recuerde que si una variable es de tipo string, debe colocar su valor entre comillas simples ' '.

***Importante**: La tabla tiene suficientes filas.*

```
def f1(li):
  for i in range(1, len(li)):
    v = li[i]
    p = i
    while p > 0 and li[p-1] > v:
      li[p] = li[p-1]
      p = p - 1
      li[p] = v
  return li

print f1([3, 2, 1, 7, 4])
```

f1			
li	i	v	p

1. [20 %] Realice el ruteo del siguiente programa e indique qué es lo que imprime. Cada vez que el valor de una variable cambie, escríbalo en una nueva fila de la tabla. Recuerde que si una variable es de tipo string, debe colocar su valor entre comillas simples ' '.

***Importante**: La tabla tiene suficientes filas.*

```
def f1(lista):
  d = {}
  i = 3
  while i > 0:
    for x in lista:
      if x[1] == i:
        if x[1] not in d:
          d[x[1]] = []
        d[x[1]].append(x)
    i -= 1
  return d
```

```
lista=[(1,3),(1,2),(2,1),(2,3)]
print f1(lista)
```