

listas

October 10, 2024

1 Ejercicio 1

Eres dueño de un restaurante con cinco platos de comida, organizados en el menú de la lista de Python a continuación. Un día decides: eliminar la sopa de frijoles (“sopa de frijoles”) del menú y agregar ensalada de remolacha asada (‘ensalada de remolacha asada’) al menú. Implementa este cambio en la lista a continuación.

Eliminar la ‘sopa de frijoles’ y agregar ‘ensalada de remolacha asada’ al final del menú.

```
[ ]: menu = [ 'carne guisada con cebolla', 'sopa de frijoles', 'risotto con trucha y gambas', 'sopa de pescado con nata y cebolla']
print(menu)
menu.remove("sopa de frijoles")
print(menu)
menu.append("ensalada de remolacha asada")
print(menu)
```

```
['carne guisada con cebolla', 'sopa de frijoles', 'risotto con trucha y gambas', 'sopa de pescado con nata y cebolla']
['carne guisada con cebolla', 'risotto con trucha y gambas', 'sopa de pescado con nata y cebolla']
['carne guisada con cebolla', 'risotto con trucha y gambas', 'sopa de pescado con nata y cebolla', 'ensalada de remolacha asada']
```

2 Ejercicio 2

La lista `num_clientes` contiene la cantidad de clientes que entraron a tu restaurante todos los días durante el último mes (que duró treinta días). Complete los valores para cada uno de los siguientes:

`avg_first_seven` - número promedio de clientes que visitaron en los primeros siete días

`avg_last_seven` - número promedio de clientes que visitaron en los últimos siete días

`max_month`: número de clientes en el día que obtuvo la mayor cantidad de clientes en el último mes

`min_month` - número de clientes en el día que obtuvo la menor cantidad de clientes en el último mes

```
[ ]: num_clientes = [137, 147, 135, 128, 170, 174, 165, 146, 126, 159, 141, 148, 132, 147, 168, 153, 170, 161, 148, 152, 141, 151, 131, 149, 164, 163, 143, 143, 166, 171]
```

```
[32]: avg_first_seven = sum(num_clientes[:7])/7
      avg_last_seven = sum(num_clientes[-7:])/7
      max_month = max(num_clientes)
      min_month = min(num_clientes)
```

```
[35]: print("Promedio de los primeros 7 dias: {:.3f}".format(avg_first_seven))
      print("Promedio de los ultimos 7 dias: {:.3f}".format(avg_last_seven))
      print("Maximo numero de clientes durante el mes: {}".format(max_month))
      print("Minimo numero de clientes durante el mes: {}".format(min_month))
```

```
Promedio de los primeros 7 dias: 150.857
Promedio de los ultimos 7 dias: 157.000
Maximo numero de clientes durante el mes: 174
Minimo numero de clientes durante el mes: 126
```

3 Ejercicio 3

Las letras deben ser una lista de Python donde cada entrada es una letra mayúscula del abecedario. Por ejemplo, las dos primeras entradas deben ser “A” y “B”, y las dos últimas entradas deben ser “Y” y “Z”. Utilice el alfabeto de cadenas para crear esta lista.

La dirección debe ser una lista de Python donde cada fila en la dirección es un elemento diferente en la lista. Actualmente, cada fila de la dirección está separada por una coma.

```
[36]: alfabeto = "A.B.C.D.E.F.G.H.I.J.K.L.M.N.Ñ.O.P.Q.R.S.T.U.V.W.X.Y.Z"
      direccion = "Mr. H. Potter,The cupboard under the Stairs,4 Privet Drive,Little_
      ↪Whinging,Surrey"
```

```
[39]: letters = alfabeto.split(".")
      formatted_address = direccion.split(",")
      print(letters)
      print(formatted_address)
```

```
['A', 'B', 'C', 'D', 'E', 'F', 'G', 'H', 'I', 'J', 'K', 'L', 'M', 'N', 'Ñ', 'O',
 'P', 'Q', 'R', 'S', 'T', 'U', 'V', 'W', 'X', 'Y', 'Z']
['Mr. H. Potter', 'The cupboard under the Stairs', '4 Privet Drive', 'Little
Whinging', 'Surrey']
```

4 Ejercicio 4

Definir una función `percent_booleana()` que toma un argumento como entrada: calificaciones: lista de calificaciones que la gente le dio a una película, donde cada calificación es un número entre 1 y 5, inclusive. Decimos que a alguien le gustó la película, si le dieron una calificación de 4 o 5. Su función debe devolver el porcentaje de personas a las que les gustó la película.

Por ejemplo, si proporcionamos un valor de `[1, 2, 3, 4, 5, 4, 5, 1]`, entonces al 50 % (4/8) de las personas les gustó la película y la función debería devolver 0,5.

```
[49]: def percent_booleana(calificaciones):
        list_booleana = [i>=4 for i in calificaciones]
        percent_booleana = sum(list_booleana) / len(list_booleana)
        return percent_booleana
```

```
[53]: print(percent_booleana([1, 2, 3, 4, 5, 4, 5, 1]))
        print(percent_booleana([1, 2, 3, 4, 5, 4, 5, 1, 4, 1, 3, 4, 2, 3]))
        print(percent_booleana([4, 2, 5, 4, 4, 1, 3, 4, 2, 3, 3, 4, 4]))
```

```
0.5
0.42857142857142855
0.5384615384615384
```

5 Ejercicio 5

Estás haciendo análisis para un sitio web. Debes escribir una función que devuelva el porcentaje de crecimiento en el número total de usuarios en relación con un número específico de años atrás.

Tu función `percent_crecimiento()` debe tomar dos argumentos como entrada:

`num_usuarios` = Lista de Python con el número total de usuarios cada año. Entonces `num_usuarios[0]` es el número total de usuarios en el primer año, `num_usuarios[1]` es el número total de usuarios en el segundo año y así sucesivamente. La entrada final en la lista da el número total de usuarios en el último año completado.

`años_atrás` = número de años para retroceder en el tiempo al calcular el porcentaje de crecimiento.

Por ejemplo, digamos que `num_usuarios` = [920344, 1043553, 1204334, 1458996, 1503323, 1593432, 1623463, 1843064, 1930992, 2001078]. si `años_atrás` = 1, queremos que la función devuelva un valor de aproximadamente 0,036. Esto corresponde a un crecimiento porcentual de aproximadamente 3,6%, calculado como $(2001078 - 1930992) / 1930992$. si `años_atrás` = 7, nos gustaría devolver aproximadamente 0,66. Esto corresponde a un crecimiento porcentual de aproximadamente 66%, calculado como $(2001078 - 1204334) / 1204334$.

```
[65]: def percent_booleana(num_usuarios, anios_atras):
        current_year = num_usuarios[len(num_usuarios) - 1]
        last_year = num_usuarios[len(num_usuarios) - 1 - anios_atras]
        crecimiento = (current_year - last_year) / last_year
        return crecimiento
```

```
[66]: num_usuarios_prueba= [920344, 1043553, 1204334, 1458996, 1503323, 1593432,
↪1623463, 1843064, 1930992, 2001078]
```

```
[72]: print("{:.3f}".format(percent_booleana(num_usuarios_prueba, 1)))
        print("{:.3f}".format(percent_booleana(num_usuarios_prueba, 7)))
```

```
0.036
0.662
```

```
[73]: print("{:.3f}".format(porcent_booleana(num_usuarios_prueba, 5)))  
      print("{:.3f}".format(porcent_booleana(num_usuarios_prueba, 9)))  
      # Pruebas con valores fuera de rango  
      print("{:.3f}".format(porcent_booleana(num_usuarios_prueba, 10)))  
      print("{:.3f}".format(porcent_booleana(num_usuarios_prueba, 11)))
```

0.331

1.174

0.000

0.036