

Python_2_solucion

February 25, 2020

1 Estructuras de datos

En programación es habitual querer agrupar variables. Para ello existen diferentes estructuras ya definidas por python que nos ayudaran a agruparlas.

1.1 Listas

Una lista en python es simplemente una lista de cosas, ya sean números, letras, textos, booleanos... En python definir, crear, una lista usaremos los simbolos [], dentro separaremos con comas , los elementos que la conforman. Por ejemplo podemos definir una lista de numeros primos del 1 al 100 de esta forma.

```
[1]: primos = [2, 3, 5, 7, 11, 13, 17, 19, 23, 29, 31, 37, 41, 43, 47, 53, 59, 61, 67, 71, 73, 79, 83, 89, 97]
      print(primos)
```

```
[2, 3, 5, 7, 11, 13, 17, 19, 23, 29, 31, 37, 41, 43, 47, 53, 59, 61, 67, 71, 73, 79, 83, 89, 97]
```

Por ejemplo también podríamos hacerlo con texto

```
[2]: frutosSecos = ["avellanas", "nueces", "anacardos", "almendras"]
      print(frutosSecos)
```

```
['avellanas', 'nueces', 'anacardos', 'almendras']
```

Para acceder a un elemento de una lista introduciremos su posición. Por ejemplo `primos[5]` nos devolvera el sexto primo. De la misma manera que `frutosSecos[3]` nos devolvera el cuarto primo. **Atencion** en python, como en muchos lenguajes de programación, **el primer elemento de una lista es el 0 no el 1**. Por lo que si queremos acceder a *avellanas* deberemos escribir `frutosSecos[0]`

```
[3]: print(primos[5])
      print(frutosSecos[3])
      print(frutosSecos[0])
```

```
13
almendras
avellanas
```

Podemos iterar, recorrer, pasar por todos los elementos, de una lista haciendo uso de los bucles. Una forma fácil en python es usar la instrucción for.

```
[4]: #La siguiente línea la leeremos como, por cada fruto seco dentro de la lista
      ↪frutos secos haz lo siguiente.
for frutoSeco in frutosSecos:
    print(frutoSeco)
```

```
avellanas
nueces
anacardos
almendras
```

1.1.1 Tu turno

Haz una función que sume todos los elementos de la lista primos.

```
[10]: def sumarPrimos(primos):
        acumulador = 0
        for elemento in primos:
            acumulador = acumulador + elemento
        return acumulador

suma = sumarPrimos(primos)
print(suma)
```

```
1060
```

Haz una función que recorra los elementos de la lista frutos secos y que imprima todos los frutos secos que cotengan la letra e.

```
[20]: def imprimirfrutosConLetra(frutosSecos, letraABuscar):
        for elemento in frutosSecos:
            elementoYaImpreso = False
            for letra in elemento:
                if letra==letraABuscar and not elementoYaImpreso:
                    print(elemento)
                    elementoYaImpreso = True

imprimirfrutosConLetra(frutosSecos, "e")
```

```
avellanas
nueces
almendras
```

1.2 Modificar una lista

Si queremos modificar un valor de una lista lo que haremos es `lista[indice] = nuevoValor`. Por ejemplo para cambien el primer fruto seco por *pistachos* haremos lo siguiente.

```
[21]: print(frutosSecos[0])
```

avellanas

```
[22]: frutosSecos[0]="pistachos"  
print(frutosSecos)
```

['pistachos', 'nueces', 'anacardos', 'almendras']

También podemos añadir y borrar elementos de una lista. Para añadir usaremos la función `myLista.append(nuevoElemento)`, esto añadirá al final de la lista el elemento agregado. Para borrar un elemento usaremos la función `pop` y la función `remove`. La función `pop` borra un elemento dado el índice, la posición, donde se encuentra `myLista.pop(indice)`. Por otro lado `remove`, borra el elemento que le digamos `myLista.remove(elemento)`.

```
[23]: primos = [2, 3, 5, 7, 11, 13, 17, 19, 23, 29, 31, 37, 41, 43, 47, 53, 59, 61, 67, 71, 73, 79, 83, 89, 97]  
print(primos)  
primos.pop(1)  
print(primos)
```

[2, 3, 5, 7, 11, 13, 17, 19, 23, 29, 31, 37, 41, 43, 47, 53, 59, 61, 67, 71, 73, 79, 83, 89, 97]

[2, 5, 7, 11, 13, 17, 19, 23, 29, 31, 37, 41, 43, 47, 53, 59, 61, 67, 71, 73, 79, 83, 89, 97]

Fijaos que se ha borrado el número que 3, que estaba en la posición 1. *Recordad* que la primera posición es la 0.

```
[26]: primos = [2, 3, 5, 7, 11, 13, 17, 19, 23, 29, 31, 37, 41, 43, 47, 53, 59, 61, 67, 71, 73, 79, 83, 89, 97]  
print(primos)  
primos.remove(5)  
print(primos)
```

[2, 3, 5, 7, 11, 13, 17, 19, 23, 29, 31, 37, 41, 43, 47, 53, 59, 61, 67, 71, 73, 79, 83, 89, 97]

[2, 3, 7, 11, 13, 17, 19, 23, 29, 31, 37, 41, 43, 47, 53, 59, 61, 67, 71, 73, 79, 83, 89, 97]

En este caso hemos borrado el número 5

```
[24]: frutosSecos = ["avellanas", "nueces", "anacardos", "almendras"]  
print(frutosSecos)  
frutosSecos.append("macadamias")  
print(frutosSecos)
```

['avellanas', 'nueces', 'anacardos', 'almendras']

['avellanas', 'nueces', 'anacardos', 'almendras', 'macadamias']

Hemos añadido macadamias al final de la lista de frutos secos

1.2.1 Tu turno

```
[25]: numeros = [5, 85, 25, 44, 101, 54, 3, 16, 99]
```

Imprime la media de la suma de todos los elementos de la lista numero

nota: Para conseguir el numero de elementos, el tamaño, de una lista podemos usar la funcion `len(miLista)`

```
[29]: def media(numeros):  
    suma = 0  
    for num in numeros:  
        suma = suma + num  
    resultado = suma / len(numeros)  
    return resultado  
  
print(media(numeros))
```

48.0

Crea una función que *retorne* el numero maximo de una lista, pruebala con la lista numeros

```
[33]: numeros = [-5, -85, -25, -44, -101, -54, -3, -16, -99]  
  
def buscarMaximo(numeros):  
    maximo = numeros[0]  
    for numero in numeros:  
        if(numero > maximo):  
            maximo = numero  
    return maximo  
  
print(buscarMaximo(numeros))
```

-3

Dada la lista de numeros numeros, crea una nueva lista vacia y añade los números pares de la lista numeros.

```
[38]: numeros = [5, 85, 25, 44, 101, 54, 3, 16, 99]  
def cogerPares(numeros):  
    listaPares = []  
    for num in numeros:  
        if(num%2==0):  
            listaPares.append(num)  
    return listaPares  
print(cogerPares(numeros))
```

[44, 54, 16]

Crea una función que diga cuales de los elementos de una lista son primos. Para ello usa a la función `esPrimo`, que creaste en el anterior notebook (debereas copiarla, o rehacerla).

```
[46]: def esPrimo(n):
    for i in range(2,n):
        if(n%i==0):
            return False
    return True

def esPrimoLista(lista):
    primos = []
    for elemento in lista:
        if (esPrimo(elemento)):
            primos.append(elemento)
    return(primos)

primos = esPrimoLista(numeros)
print(primos)
```

[5, 101, 3]

```
[54]: def NOBOLEANOesMasgrandeQueCinco(numero):
    if (numero > 5):
        print("es mas grande")
    print ("no es mas grande")
    NOBOLEANOesMasgrandeQueCinco(int(input()))
```

4

no es mas grande

Crea una función que dada una lista de textos y un número, imprima las palabras con un número de letras mayor o igual al número. Usa la lista `hobbit` para testearla.

```
[56]: texto = "En un agujero en el suelo, vivía un hobbit. No un agujero húmedo, \
    ↪sucio, repugnante, \
con restos de gusanos y olor a fango, ni tampoco un agujero, seco, desnudo y \
    ↪arenoso, \
sin nada en que sentarse o que comer: era un agujero-hobbit, y eso significa \
    ↪comodidad"
hobbit = texto.replace(',',' ').split()
print(hobbit)
```

```
['En', 'un', 'agujero', 'en', 'el', 'suelo', 'vivía', 'un', 'hobbit.', 'No',
'un', 'agujero', 'húmedo', 'sucio', 'repugnante', 'con', 'restos', 'de',
'gusanos', 'y', 'olor', 'a', 'fango', 'ni', 'tampoco', 'un', 'agujero', 'seco',
'desnudo', 'y', 'arenoso', 'sin', 'nada', 'en', 'que', 'sentarse', 'o', 'que',
'comer:', 'era', 'un', 'agujero-hobbit', 'y', 'eso', 'significa', 'comodidad']
```

Crea una función que dada una lista de palabras devuelva una lista con las palabras que contengan

la letra *e*

```
[64]: def palabrasQueContienenLetra(lista, letra):
    palabrasConLetra = []
    for palabra in lista:
        palabraAnadida = False
        for letraActual in palabra:
            if (letra==letraActual) and (not palabraAnadida):
                palabrasConLetra.append(palabra)
                palabraAnadida = True
    return palabrasConLetra

listaPalabras = palabrasQueContienenLetra(hobbit, 'e')
print(listaPalabras)
```

```
['agujero', 'en', 'el', 'suelo', 'agujero', 'húmedo', 'repugnante', 'restos',
'de', 'agujero', 'seco', 'desnudo', 'arenoso', 'en', 'que', 'sentarse', 'que',
'comer:', 'era', 'agujero-hobbit', 'eso']
```

Crea una función que dada una lista de números, por ejemplo “numeros”, te devuelva la lista ordenada

```
[73]: def buscarMinimo(lista):
    minimo = lista[0]
    for numero in lista:
        if(numero < minimo):
            minimo = numero
    return minimo

def ordenarListaMinimo(lista):
    listaOrdenada = []
    for i in range(0,len(lista)):
        minimo = buscarMinimo(lista)
        listaOrdenada.append(minimo)
        lista.remove(minimo)
    return listaOrdenada

numeros = [5, 85, 25, 44, 101, 54, 3, 16, 99]
print(ordenarListaMinimo(numeros))
```

```
[3, 5, 16, 25, 44, 54, 85, 99, 101]
```

```
[ ]:
```