

```
In [2]: nombre_fichero = './dat/blacklist.dat'

fichero = open (nombre_fichero, 'w')
fichero.write("malparit")
fichero.close()
```

```
In [11]: nombre_fichero = './dat/blacklist.dat'

fichero = open (nombre_fichero, 'a')
fichero.write("guarro" + "\n")
fichero.close()
```

```
In [12]: nombre_fichero = './dat/blacklist.dat'

fichero = open (nombre_fichero, 'r')
texto = fichero.read()
fichero.close()

lineas = texto.split("\n")

print (lineas)

# print (texto)
# type(texto)

['malparitguarro', 'guarro', 'guarro', 'guarro', '']
```

```
In [21]: nombre_fichero = './dat/blacklist.dat'

fichero = open (nombre_fichero, 'r')
texto = fichero.read()
fichero.close()

lineas = texto.split("\n")
lineas.append("la última de la lista")
print (lineas)

fichero = open (nombre_fichero, 'w', encoding="UTF-8")
fichero.write("\n".join(lineas))
fichero.close()

['', 'la última de la lista', 'la última de la lista']
```

```
In [31]: texto = """1. Traspasando el Top-Secret Ufo
2. Contacto Alienígena
3. Identidad Extraterrestre
4. La Física y Ufo.
5. Lugares Magnetizados
6. Colisión Ufo
7. Sobre nuestro Planeta
8. Abducciones
9. La Invasión silenciosa
10.Curaciones Ufo"""

nombre_fichero = "./dat/indice.txt"

fichero = open (nombre_fichero, 'w', encoding="UTF-8")
texto = fichero.write(texto)

fichero.close()

fichero = open (nombre_fichero, 'r')
texto = fichero.read()
lineas = texto.split("\n")
fichero.close()

#-----a totes les llistes canviar numeros a 0

for i, linea in enumerate (lineas):
    lineas[i] = "0"+ lineas[i]

nombre_fichero = "./dat/indice_new.txt"
fichero = open (nombre_fichero, 'w', encoding="UTF-8")
fichero.write("\n".join(lineas))
fichero.close()
```

```
In [35]: texto = """Llistes
En Python, les llistes són probablement l'estructura de dades més usada. Però el
nom 'llista' és probablement enganyós, ja que des del punt de vista de les seves
operacions (i el seu temps d'execució) es tracta més aviat de vectors (o arrays).
Una llista permet emmagatzemar seqüencialment un nombre variable de dades
(de qualsevol tipus).
Llistes per compressió
Les llistes per compressió proporcionen una manera concisa de crear llistes tot
utilitzant una sintaxi semblant a la de la definició de conjunts en matemàtic"""

nombre_fichero = "./dat/ejercicio5.txt"
fichero = open (nombre_fichero, 'w', encoding="UTF-8")
fichero.write(texto)
fichero.close()
```

```
In [36]: nombre_fichero = "./dat/ejercicio5.txt"
fichero = open (nombre_fichero, 'r', encoding="UTF-8")
texto = fichero.read()
fichero.close()
print(texto)
```

Llistes

En Python, les llistes són probablement l'estructura de dades més usada. Però el nom 'llista' és probablement enganyós, ja que des del punt de vista de les seves operacions (i el seu temps d'execució) es tracta més aviat de vectors (o arrays). Una llista permet emmagatzemar seqüencialment un nombre variable de dades (de qualsevol tipus).

Llistes per compressió

Les llistes per compressió proporcionen una manera concisa de crear llistes tot utilitzant una sintaxi semblant a la de la definició de conjunts en matemàtic

```
In [37]: nombre_fichero = "./dat/ejercicio5.txt"
fichero = open (nombre_fichero, 'r', encoding="UTF-8")
texto = fichero.read()
fichero.close()
print(texto)

for palabra in ('dades', 'variable', 'lliure'):
    if palabra in texto:
        print (palabra, texto.count(palabra))
    else:
        print ("no esta", palabra)
```

Llistes

En Python, les llistes són probablement l'estructura de dades més usada. Però el nom 'llista' és probablement enganyós, ja que des del punt de vista de les seves operacions (i el seu temps d'execució) es tracta més aviat de vectors (o arrays). Una llista permet emmagatzemar seqüencialment un nombre variable de dades (de qualsevol tipus).

Llistes per compressió

Les llistes per compressió proporcionen una manera concisa de crear llistes tot utilitzant una sintaxi semblant a la de la definició de conjunts en matemàtic

dades 2

variable 1

no esta lliure

```
In [14]: # Ejercicio 7:
# Crea un programa que lea y muestre el fichero
# blacklist.dat.
# Seguidamente el programa debe pedir nuevas palabras y si no
# existen, las añade al fichero por orden alfabético.

nombre_fichero = "./dat/blacklist.dat"
fichero = open (nombre_fichero, 'r', encoding="UTF-8")
texto = fichero.read()
fichero.close()
lineas = texto.split("\n")

palabra = input ("Palabra que quieres añadir o salir: ")

while palabra != "salir":
    if palabra not in lineas:
        lineas.append(palabra)
    else:
        print ("la palabra esta repetida")
        palabra = input ("siguiente palabra o salir: ")

lineas.sort()

nombre_fichero = "./dat/blacklist.dat"
fichero = open (nombre_fichero, 'w', encoding="UTF-8")
fichero.write("\n".join(lineas))
fichero.close()

print (lineas)
```

```
Palabra que quieres añadir o salir: puto
siguiente palabra o salir: puto
la palabra esta repetida
siguiente palabra o salir: salir
['', 'chorlito', 'guarro', 'maldito', 'puto', 'roedor']
```

```
In [24]: # Copia y pega estos datos en un fichero con la extensión.csv
# - Haz un programa Python para :
# 1. Leer el fichero
# 2. Imprimir todas las líneas en columnas separadas por un tabulador: "Any
# Codi Municipi Edats Homes Dones Total"
# 3. Crea una copia del archivo y marca con un asterisco adicional al final *,
# todos los registros que tienen el valor de hombres mayor que el de
# mujeres. Si el valor de hombres es menor, escribe solo un separador
# adicional.

texto = """Any;Codi;Municipi;Edats;Homes;Dones;Total
2017;250019; Abella de la Conca; De 0 anys;0;0;0
2017;250019; Abella de la Conca; D'1 any;0;1;1
2017;250019; Abella de la Conca; De 2 anys;1;1;2
2017;250019; Abella de la Conca; De 3 anys;0;0;0
2017;250019; Abella de la Conca; De 4 anys;3;0;3
2017;250019; Abella de la Conca; De 5 anys;2;1;3
2017;250019; Abella de la Conca; De 6 anys;0;0;0
2017;250019; Abella de la Conca; De 7 anys;1;0;1"""

lineas = texto.split('\n')

print ("Homes Dones")

for linea in lineas[1:]:
    camps = linea.split(";")

    if len(camps) > 4:
        marca = ""
        if int(camps[4]) > int(camps[5]):
            marca = "*"

    print (camps[4], '\t', camps[5], marca)
```

```
Homes Dones
0      0
0      1
1      1
0      0
3      0 *
2      1 *
0      0
1      0 *
```

```
In [27]: # Queremos inventarnos Las pernотaciones (estancias) de varios hoteles, para
# distintos años y meses.
# Queremos generar los datos de forma aleatoria en función de las plazas de cada
# hotel. Sabemos que en los meses de julio y agosto la ocupación de todos los
# establecimientos está entre el 70 y 100%.
# Usando este fragmento de programa, pide el nombre del hotel, el año y las
# plazas máximas y genera un csv con los datos del hotel, año, plazas, dando un
# número aleatorio de pernотaciones que no supere las (plazas* días) del mes,
# para cada mes (01 a 12).

import easygui_qt as easy
labels = ["Establiment", 'Any', 'Capacitat']
resposta = easy.get_many_strings(title="Pernотaciones", labels=labels)

print (resposta)
```

```
OrderedDict([('Establiment', 'Hotel Ot'), ('Any', '2022'), ('Capacitat', '20')])
```

```
In [34]: from random import randint

nom = resposta ["Establiment"]
anyy = resposta ["Any"]
capacitat = resposta ["Capacitat"]

lineas = []

for i in range(1,13):
    linea = f"{nom};{anyy};{capacitat}; {i};{randint(0,int(capacitat)*28)}"
    print(linea)
```

```
Hotel Ot;2022;20; 1;260
Hotel Ot;2022;20; 2;293
Hotel Ot;2022;20; 3;476
Hotel Ot;2022;20; 4;325
Hotel Ot;2022;20; 5;187
Hotel Ot;2022;20; 6;311
Hotel Ot;2022;20; 7;366
Hotel Ot;2022;20; 8;109
Hotel Ot;2022;20; 9;245
Hotel Ot;2022;20; 10;91
Hotel Ot;2022;20; 11;416
Hotel Ot;2022;20; 12;416
```

```
In [ ]:
```

```
In [ ]:
```