

LISTAS,

- Tienen orden
 - Son mutables (en relación a sus datos)
 - Valores entre corchetes [a,b,1,2]
1. L=[] L=list() Abrir nueva lista en blanco
 2. L.append(x) Agregar el valor x al final de la lista ("x" si es string, x si es número)
 3. L.index(x) Entrega la posición del elemento x de la lista L, partiendo de 0
 4. len(L) Entrega el n° de elementos de la lista L
 5. L[x] Entrega el valor que está en la posición x de la lista L, partiendo de 0
 6. del L[x] Elimina el valor de la lista L que se encuentra en la posición x
 7. L.remove(x) Elimina el valor x de la lista L
 8. L.insert(x,y) Inserta el valor y en la posición x, corriendo todos los valores que le siguen en la lista L hacia la derecha
 9. L.count(x) Cuenta cuántas veces se repite el elemento x en la lista L
 10. sum(L) suma los términos, en una lista de NUMEROS, NO STRINGS
 11. min(L) Entrega el valor mínimo de la lista L, comenzando por números y luego letras
 12. max(L) Entrega el valor máximo de la lista L, comenzando por la z y terminando por el 0
 13. L.sort() Sólo modifica la lista L y la ordena por números y luego por abecedario si hubiesen string
 14. L.reverse() Modifica la lista L y la ordena inversamente, partiendo de la z al 0
 15. list(range(3)) Crea una lista que contiene los números del 0 al 2
 16. x in L Entrega True o False, de si el valor x se encuentra en la lista L
 17. list("hola") Entrega una lista, en la que cada elemento es un carácter del string
 18. L1+L2 Entrega la concatenación de los caracteres de ambas listas, pero NO modifica ninguna de ellas, sólo los muestra juntos
 19. b=list(L) convertir la tupla L en una lista b
 20. sorted(L)[x] ordena la lista L y entrega el valor de la posición x
 21. L.sort(key=def) ordena la lista L de acuerdo a la función def
 22. L[x][y] Entrega el valor de posición y del valor de la posición x. Se puede obtener el valor de una lista o tupla dentro de otra lista o tupla

TUPLAS

- Valores entre parentesis (a,b,1,2)
 - Tienen orden
 - Son inmutables (en relación a sus datos)
1. T=() T=tuple() Tupla vacía, AUNQUE SE PUEDE DEFINIR UNA TUPLA SIN COMAS , T=2,4,5
 2. x,y,z=T ó x,_z=T si no queremos usar y "Desempaquetar" tupla, o sea, obtener los elementos, en ese orden, y asignarles nuevas variables a cada valor, se debe saber el número de elementos que posee la tupla
 3. (4,5,7)<(7,0,0) True, porque compara el primero con el primero
(4,5,7)<(4,2,1) False, porque compara el segundo con el segundo, ya que los primeros son iguales
 4. Tupla[x] Entrega el valor x-ésimo de la tupla
 5. b=tuple(L) Convertir la lista L en una tupla b
 6. sum(T) Suma los valores numéricos de la tupla
 7. T[x][y] Entrega el valor de posición y del valor de la posición x. Se puede obtener el valor de una lista o tupla dentro de otra lista o tupla

DICCIONARIO

- No tienen orden
 - Pares llave-valor
 - Llaves no repetidas
 - Son mutables (en relación a sus datos)
 - Con la llave puedo obtener el valor, pero con el valor no puedo obtener la llave
1. d={} d=dict() Abrir diccionario
 2. d={"x":2,"y":4} x=Llave (no pueden estar repetidas, pueden ser números, string o tuplas), 2=valor
 3. x=list(d) Crea una lista de SÓLO LAS LLAVES del diccionario

D.B.Q.B.

- | | |
|---|--|
| 4. <code>x=list(d.values())</code> | Crea una lista de SÓLO LOS VALORES del diccionario |
| 5. <code>x in, not in d</code> | Busca la LLAVE en el diccionario |
| 6. <code>x in, not in d.values()</code> | Busca el valor en los valores del diccionario |
| 7. <code>len(d)</code> | Cantidad de LLAVES |
| 8. <code>d[llave]=valor</code> | Agregar o modifica la llave "llave" con su respectivo valor "valor" al diccionario d |
| 9. <code>d.items()</code> | Es una lista de las tuplas,[(llave, valor)] |
| 10. <code>d.values()</code> | Es una lista de los valores |
| 11. <code>del d[llave]</code> | Elimina llave y SU VALOR ASOCIADO |
| 12. <code>sum(d)</code> | Suma llaves SÓLO CUANDO SON ENTEROS del diccionario d |
| 13. <code>sum(d.values)</code> | Sumas los valores del diccionario d |

CONJUNTOS

- No tienen orden
 - Elementos no repetidos
 - Son mutables (en relación a sus datos)
 - El elemento puede ser TUPLA, NÚMERO, STRING O CARÁCTER
- | | |
|--|---|
| 1. <code>c=set()</code> | Conjunto vacío, para abrirlo, DIFERENTE de <code>c={0}</code> este es un conjunto que tiene en número 0 |
| 2. <code>c.add(x)</code> | Agrega el valor x |
| 3. <code>c.remove(x)</code> | Remueve el valor x en el conjunto |
| 4. <code>x in, not in c</code> | Booleano si el valor x está, no está en el conjunto c |
| 5. <code>c & v</code> (et o ampersand) | Intersección de los conjuntos c & v |
| 6. <code>c v</code> | Unión del conjunto c y v |
| 7. <code>c-v</code> | Diferencia del conjunto c menos v, elementos de c que no están en v |
| 8. <code>a^v</code> (intercalación o acento circunflejo) | Diferencia simétrica. Elementos que están en v o c pero no en ambos |
| 9. <code>a<b</code> | Subconjunto (booleano) |
| 10. <code>len(c)</code> | Entrega el largo de c, o sea, la cantidad de elementos que tiene el conjunto c |
| 11. <code>set(x)</code> | Lleva la variable x a conjunto, si x es un string, hace un conjunto con cada uno de los caracteres que con tiene dicho string |

FUNCIONES ESPECIALES

- | | |
|------------------------------------|---|
| 1. <code>x.lower()</code> | Llevar un string todo a minúsculas |
| 2. <code>x.upper()</code> | Llevar un string todo a mayúsculas |
| 3. <code>x.split()</code> | Hace una lista de todas los caracteres separados por un espacio que tiene la variable x |
| 4. <code>x.replace('a','b')</code> | En el string x reemplaza con 'b' cada 'a' que encuentre |
| 5. <code>pprint</code> | Imprimir hacia el lado, en vez de hacia abajo |
| 6. <code>map(función,x)</code> | Retorna una lista de la x aplicada la fucion "funcion" |

