Operaciones con Lista (Python)

**len**

Este sirve para saber cuantos elementos hay en una lista.

**Ejemplo:**

>>>t=[1,2,3]  
>>>len(t)  
>>>3

**Cambiar los elementos**

En las listas se puede cambiar los elementos de una lista por otros

**Ejemplo:**

>>>u=['5',8,'to']  
>>>u[1]=9+2  
>>>u  
>>>y=['dos',11,'A']

**append**

Este sirve para anadir un elemento al final de una lista.

**Ejemplo**

>>>y=['dos',8,'5A']  
>>>y.append('tres')  
>>>y  
['dos',8,'5A','tres']

**extend**

Se usa para anadir elementos de otra lista, es decir concadenar

**Ejemplo:**

>>>Lista.extend(['4','6'])  
>>>Lista  
>>>['a','b','alfa  
>>>['cinco',71,'A','cosa','4','6']

**insert**

sirve para anadir un elemento insertandola en una posicion concreta

**Ejemplo:**

>>>lis=[1,8,'tres','cinco']  
>>>lis.insert(2,'emacs')  
>>>lis  
[1,8,'emacs','tres','cinco']

**remove**

sirve para borrar un elemento de un lista

**Ejemplo:**

>>>mar=[9,14,'dos','seis']  
>>>mar  
>>>lis  
[1,8,'emacs','tres','cinco']

**In**

Se utiliza para verificar si un determinado dato esta dentro de una lista, y responde false o true

**Ejemplos**

>>>ti=(2,'feo',3,'bonito')  
>>>'ka' in ti  
false

**Count**

Dice el numero de veces que un dato o elemento esta en una lista

**Ejemplo**

>>>lu=('feo','sa','feo','sa','sa')   
>>>lu.count('sa')   
3

**Reverse**

Muestra al reves los elementos de una lista

**Ejemplos**

>>>casa=('Pedro',2,5,4,3)  
>>>casa.reverse()  
>>>casa   
(3,4,2,5,'Pedro') 

**Multiplicar**

Con listas tambien se puede multiplicar, de modo que el resultado sea la repeticion de los elementos de la cadena, segun el numero que pongamos al multiplicar

**Ejemplos**

>>>j=(5,3,'g')  
>>>j\*5  
(5,3,'g',5,3,'g',5,3,'g',5,3,'g',5,3,'g')  
>>>j  
(5,3,'g')

**del**

Sirve para borrar un elemento de un lista no importando el valor sino la posicion

**Ejemplo**

>>>go=[72,23,'om','ar']  
>>>go  
[72,23,'om','ar']  
>>>del go[1]  
[72,'om','ar']

**Sumatoria**

El comportamiento de la suma es muy parecido al de extend, la diferencia esta en que la suma devuelve una nueva lista.

**Ejemplo**

>>>tr = [5,6,8,7]  
>>>lu = ['mo', 'ho', 'la', 'co']  
>>>lu + tr  
[5,6,8,7,'mo','ho','la','co']  
>>>tr  
[5,6,8,7]

**list**

Permite convertir un string en una lista.

**Ejemplo**

>>>[5,7]+list("am")  
[5,7,'a','m']

**comparar**

Permite comparar entre listas.

**Ejemplo**

>>>a = [7,4,5]  
>>>b = [4,5,7]  
>>>c = b  
>>>a == b  
False  
>>>c == b  
True

**O:**

>>>[12,75,89]<[2,1,3]  
False  
>>>[8,7,9]<=[8,7,9]  
True  
>>>[12,75,89]>[2,1,3]  
True

**Slicing**

Sirve para generar subconjuntos, es decir, listas que provienen de otras listas y se indican por dos indices (inicio y fin)

**Ejemplo:**

>>>a=[1,'ba',3,4]  
>>>a[1:3]  
['ba',3]

**index**

Sirve para mostrar el indice de un elemento de una lista

**Ejemplos**

>>>ka=[2,'perro',8,'gato']  
>>>ka.index('perro')

**For**

ES un ciclo que recorre los elementos de una lista

**Ejemplo**

>>>tu=['Omar','Emmanuel','Antonio','Omar','Jose','jose','Emmanuel','Jose']  
>>>Numero=0  
>>>for tus in tu:  
if tus='jose'  
Numero=numero+1  
>>>Numero  
3

**anidamiento**

Este sirve para anadir un elemento a la lista.

**Ejemplo**

>>>st=['coma','tren']  
>>>ut=['ram',em',8]  
>>>ut[3].append(st)  
>>>ut  
['ram','emm',8,['coma','tren']]

**sort**

Es la funcion que nos permite ordenar los elementos de una lista

**Ejemplos**

54=(8,12,6,16,4)  
>>>54.sort()  
>>>54 (4,6,8,12,16)

**split**

Es la funcion que nos devuelve una lista con todas las palabras de una cadena

**Ejemplo**

>>>nem="jose fernando rodriguez"  
>>>nem.split()  
('jose','fernanado','rodriguez')

**Join**

Es la función que convierte una lista en una cadena de texto

**Ejemplos**

>>>5=('Nelson','Ramirez','Garcia')  
>>>5  
>>>('Nelson','Ramirez','Garcia')  
>>>join(5)  
>>>'Nelson Ramirez Garcia'

**pop**

Nos sirve para mostra el ultimo elemento de la lista y el cual es borrado

**Ejemplos**

>>>cristal  
(5's')  
>>>cristal.pop()  
's'  
>>>cristal  
(5)

**Tambien el pop nos permite borrar un elemento sabiendo su ubicacion dentro de la lsita**

**Ejemplos**

>>>ate=range(6)  
>>>ate  
(0,1,2,3,4,5)  
>>>ate.pop(4)  
4  
>>>ate  
(0,1,2,3,5)

Operaciones con cadenas (Python)

S=”hola mundo”

>>> print s[2]  
l  
>>> print s[5:10]  
Mundo  
>>> print s[:4]  
Hola  
>>> print s[8:]  
do  
>>> print s[-5:-2]  
Mun

**int len(string):**retorna la longitud de una cadena.

**int count(sub [,start[,end]]):** retorna el número de ocurrencias de una cadena dentro de otra, por ejemplo:

>>> print s.count(“Hola”)  
1  
>>> print s.count(“o”)  
2

**boolean endswidth(sub [,start[,end]]):** retorna verdadero o falso si una cadena termina con la cadena especificada.

>>> print s.endswith(“do”)  
True  
>>> print s.endswith(“d”)  
False

**boolean startswidth(sub [,start[,end]]):** similar a endswidth.

>>> print s.startswith(“Hola”)  
True  
>>> print s.startswith(“Mundo”)  
False

**int find(sub [,start[,end]]):** retorna la posición numérica de la primera ocurrencia de una cadena dentro de otra.

>>> print s.find(“o”)  
1

**int rfind(sub [,start[,end]]):** similar a find, pero en vez de retornar la posición de la primera ocurrencia, lo hace de la última.

>>> print s.rfind(“o”)  
9

**string lower():** retorna la cadena original en letras minúsculas.

>> print s.lower()  
hola mundo

**string upper():** retorna la cadena original en letras mayúsculas.

>>> print s.upper()  
HOLA MUNDO

**string replace(old, new [,count]):** reemplaza todas o un número dado de ocurrencias de una cadena dentro de otra.

>>> print s.replace(“o”,”O”)  
HOla MundO  
>>> print s.replace(“o”,”O”,1)  
HOla Mundo

**string strip([chars]):** retorna una cadena eliminando al inicio y el fin de la misma un caracter dado. Si no se indica caracter utiliza como predeterminado el espacio en blanco.

>>> print s.strip()  
Hola Mundo  
>>> print s.strip(“H”)  
ola Mundo  
>>> print s.strip(“o”)  
Hola Mund

**array split([sep [,maxsplit]]):** divide una cadena en un array de cadenas dado un separador. Si no se indica separador utiliza como predeterminado el espacio en blanco. Opcionalmente, se puede especificar el número máximo de divisiones.

>>> a = ['abc','def','ghi']  
>>> t = “|”  
>>> print t.join(a)  
abc|def|ghi

**string join(array):** combina los elementos de un array de cadenas con otra cadena.

>>> s.split()  
['Hola', 'Mundo']  
>>> s.split(“o”)  
['H', 'la Mund', '']