

Junto con su grupo de trabajo investigue 5 funciones built-in de Python y explique sus usos y dos ejemplos donde se utilice dicha función.\

1- List()

Una función se utiliza para convertir un objeto en una lista

La sintaxis es:

list[conjunto de números/datos]

Aquí se coloca cualquier dato o secuencia en cadena tales como tupla, objeto iterable, colecciones, un conjunto o diccionario, etc

Se devuelve una secuencia mutable de una lista de elementos

Ejemplo:

Creamos lista para el ejercicio de frutas

```
'''Crear una lista con nombres de frutas, posteriormente hacer una función que pida al usuario el nombre de otra fruta y que esta sea agregada a la lista para luego imprimir en pantalla '''

frutas = ['papaya', 'manzana', 'guineo', 'fresa']
fruta = input("Digite el nombre de una fruta: ")

frutas.append(fruta)

print("Frutas: ", frutas)
```

Salida:

```
Digite el nombre de una fruta: tomate
Frutas: ['papaya', 'manzana', 'guineo', 'fresa', 'tomate']
```

Ejemplo n2:

Lista para sacar el promedio de cierta cantidad de numeros

```
'''Crear un código donde una función reciba una lista de números ya hecha y
devuelva el promedio '''

#este ejercicio esta echo para poder devolver un promedio de un alisat atraves de un array

lista=[54, 48, 47, 46, 56, 37, 64, 46, 8, 58, 67, 14, 85, 73, 67, 43, 89, 30, 67, 82]

def promedio_std(lista):
    x = 0

    x = sum(lista) /len(lista)
    total = 0.0
    return x

print(promedio_std(lista))
```

Salida

```
54.05
```

2- Sorted()

función para clasificar números de una manera fácil e eficiente

su sintaxis:

`sorted(iterables)`

clave = clave para ordenar los valores

inverso = si se establece true para generar una lista en orden descendente
el resultado siempre será una lista

Ejemplo:

Tenemos unas listas y queremos que se orden de manera al revés

```
sampleObj = (3,6,8,2,5,8,10)
print(sorted(sampleObj,reverse=True))
sampledict = {'a':'sss','g':'wq','t':2}
print(sorted(sampledict,key= len))
```

Salida:

```
[10, 8, 8, 6, 5, 3, 2]
['a', 'g', 't']
```

Ejemplo n2:

Tenemos una lista y queremos que la ordene en forma de abecedario

```
sampleObj = ('manzana','abeja', 'zorrillo')
print(sorted(sampleObj,reverse=False))
```

Salida:

```
['abeja', 'manzana', 'zorrillo']
```

3- Input()

Esto sirve para que el usuario pueda digitar su nombre, numero o cualquier dato que el desee

Sintaxis:

Input()

Ejemplo:

Pedir el nombre de una fruta al usuario y como salida el X nombre de la fruta

```
fruta = input("Digite el nombre de una fruta: ")
```

Salida:

```
tomate
```

Ejemplo n2:

Pedimos un numero al usuario para hacer X operación matematica

```
a = float(input("Digite el primer numero: "))
```

Salida:

```
Digite el primer numero: 1
```

4- round()

Sirve para redondear un numero en Python

Sintaxis:

```
print(round 6.9)
```

Ejemplo:

```
a = float(input("Digite el primer numero: "))
b = float(input("Digite el segundo numero: "))

sum = a + b
print(round(sum))
```

Salida:

```
Digite el primer numero: 1.4
Digite el segundo numero: 1.1
2
```

Ejemplo n2:

```
a = float(input("Digite el primer numero: "))
b = float(input("Digite el segundo numero: "))

mult = a * b
print(round(mult))
```

Salida:

```
Digite el primer numero: 3
Digite el segundo numero: 5.4
16
```

5- Max()

Devuelve el numero mayor de un objeto iterable

Sintaxis:

`max(iterable) or max(num1,num2,+...)`

Ejemplo:

```
num = [9,5,3,12,23,4,6,7,21,2]  
print('Max es:', max(num))
```

Salida:

```
Max es: 23
```

Ejemplo n2:

```
num = [12,18,32,12,14,6,56]  
print('Max de edad: ', max(num))
```

Salida:

```
Max de edad: 56
```

Haga un cuadro comparativo donde se ejemplifique las diferencias entre tuplas, diccionarios, listas y conjuntos y un ejemplo por cada uno. (30%)

	<i>Diccionarios</i>	<i>Tuplas</i>	<i>Listas</i>	<i>Conjuntos</i>
<i>Características</i>	Son dinámicos, ya que pueden aumentar o disminuir, además se pueden agregar o eliminar elementos en él, son indexados, ya que los elementos del diccionario son accesibles a través del campo key, y son anidados, por lo que un diccionario puede contener a otro diccionario en su campo value.	Se usan para representar colecciones fijas de elementos, por ejemplo: las fechas de un calendario, los meses de un año, etc.	Se usan para almacenar conjuntos de elementos relacionados del mismo tipo o de tipos distintos.	Es una colección no ordenada de objetos únicos. No puede incluir objetos cambiables como listas, diccionarios, e incluso otros conjuntos. Los elementos de un set son únicos , lo que significa que no puede haber elementos duplicados.
<i>Sintaxis</i>	diccionario = {"clave1": "valor1", "clave2": "Valor2", "clave3": "Valor3"}	tupla1 = ('elemento1', 'elemento2', 'elemento3', 'elemento4', 'elemento5')	lista = [valor1, valor2, valor3, valor4]	conjunto = {"valor1", "valor2", "valor3"}
<i>Ejemplos</i>	diccionario = { "Nombre": "Alfredo", "Edad": 19, "Código": "SMIS069221" } print(diccionario)	días = ('lunes', 'martes', 'miércoles', 'jueves', 'viernes', 'sábado', 'domingo') print(días)	lista = ['uno', 2, 3.14159, 'cinco', 6, 'siete'] print(lista)	países = {"Mexico", "Venezuela", "Cuba"} print(thisset)