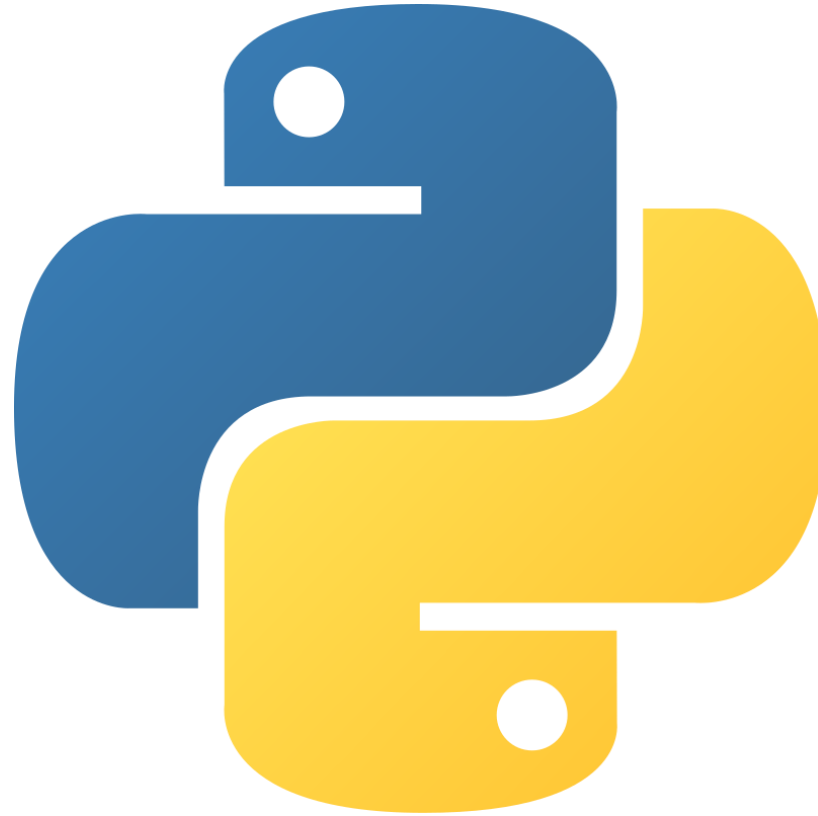


CURSO DE PYTHON

Temario:

- Listas
- Diccionarios
- Tuplas
- Ejercicios



LISTAS

1.Lista

Es una estructura de datos que pueden contener elementos de diferentes tipos. Los elementos están separados por comas.

```
>>> squares = [1, 4, 9, 16, 25]
>>> squares
[1, 4, 9, 16, 25]
```

```
>>> squares[0] # indexing returns the item
1
>>> squares[-1]
25
>>> squares[-3:] # slicing returns a new list
[9, 16, 25]
```

Métodos de una lista

lista=["apple" , "banana" , "cherry"]	
Descripción	Ejemplo
Indica la posición de lo que se busca	>>> lista.index("banana") 1
Agrega un elemento al final	>>> lista.append("orange") >>> lista >>> ["orange", "apple" , "banana" , "cherry"]
Borrar un elemento de lista	>>> lista.remove("cherry") >>> lista >>> ["orange", "apple" , "banana"]
Inserta un elemento en la posición indicada	>>> lista.insert(1, "pera") >>> lista >>> ["apple", "pera" , "banana", "cherry"]
Elimina un elemento de la lista	>>> lista.pop(2) >>> lista >>> ["apple","banana"]
Copia una lista en otra	>>> mylista = lista.copy()
Borra todos los elementos	>>> lista.clear()

Métodos de una lista

lista=["apple" , "banana" , "cherry"]	
Descripción	Ejemplo
Devuelve el numero de veces que el elemento aparece	>>> lista.count("banana") 1
Ordena los elementos de una lista	>>> lista.sort() >>> ["apple" , "banana" , "cherry"]
Invierte los elementos de la lista	>>> lista.reverse() >>> ["cherry" , "banana", "apple"]
Tamaño de una lista	>>> len(lista)
Obtener un elemento de la lista	>>> lista[1] >>> banana

Tuplas

2.Tuplas

Es una estructura de datos ordenada e **inmutable** , se escriben entre paréntesis. Las tuplas pueden repetir elementos

```
>>> mytupla = ("manzana", "plátano", "fresa")
>>> mytupla
("manzana", "plátano", "fresa")
```

Métodos de una tupla

tupla=(“apple” , “banana” , “cherry”)	
Descripción	Ejemplo
Tamaño de una tupla	>>> len(tupla) 3
Elemento de una tupla	>>> tupla[1] >>> banana
Convertir tupla a lista para agregar elemento	>>>y = list(tupla) >>> y[1] = “Orange” >>> tupla = tuple(y)
Convertir tupla a lista para remover elemento	>>>y = list(tupla) >>> y.remove(“banana”)

Diccionario

3.Diccionario

Es una estructura de datos donde los valores se almacenan en pared de clave y valor.

Es una colección ordenada, modificable y no admite elementos duplicados.

```
>>> tel = {'jack': 4098, 'luis': 4139}
>>> tel['guido'] = 4127
>>> tel
{'jack': 4098, 'luis': 4139, 'guido': 4127}
>>> tel['jack']
4098
```


Métodos de un diccionario

dic={"casa": 12, "cocina" : "grande", "oficina" : "nueva"}	
Descripción	Ejemplo
Obtener el valor de la clave	>>> dic["casa"] 12
Otro método para obtener el valor de una clave	>>> x = dic.get("casa") >>> print(x) 12
Obtiene las llaves	>>> x = dic.keys() >>> print(x) dict_keys(['casa', 'cocina', 'nueva'])
Agregar una nueva llave	>>> x = dic.keys() >>> dic["sala"] = "verde" >>> print(x) dict_keys(['casa', 'cocina', 'nueva', 'sala'])
Obtiene los valores de las llaves	>>> x = dic.values() >>> print(x) dict_values([12, 'grande', 'nueva'])
Cambiar un valor	>>> x = dic.values() >>> dic["oficina"] = "vieja" >>> print(x) dict_values([12, 'grande', 'vieja'])

Métodos de un diccionario

dic={"casa": 12, "cocina" : "grande", "oficina" : "nueva"}	
Descripción	Ejemplo
Obtiene los elementos del diccionario	<pre>>>> x = thisdict.items() >>> print(x) dict_items([('casa', 12), ('cocina', 'grande'), ('oficina', 'nueva')])</pre>
Elimina elemento con el nombre de la clave	<pre>>>> dic.pop("casa")</pre>
Elimina el ultimo elemento	<pre>>>> dic.popitem()</pre>
Vacía el valor del diccionario	<pre>>>> dic.clear()</pre>

GRACIAS

Twitter: @ccencho

Facebook: /ccencho

Mail: nccencho@smelpro.com

Visítanos: www.smelpro.com