



UNIVERSIDAD PRIVADA DE TACNA

Listas en Python

ING.LILIANA VEGA BERNAL

Python tiene muchos tipos de estructura de datos, una de ellas es la lista.

En algunos lenguajes de programación se las conocen como arreglos o matrices; y se caracterizan porque los **elementos** están entre **corchetes** y separados por una **coma**.

```
lista = [1, 2, 3, 4]
```

Lo especial de las listas en Python es que nos permiten almacenar cualquier tipo de valor como enteros, cadenas y hasta otras funciones; por ejemplo:

```
lista = [1, 2.5, 'DevCode', [5,6] ,4]
```

A large, faint watermark of the University of Tacna logo is centered in the background of the slide.

Métodos de las Listas

Append() agregar elementos a la lista

```
lista.append(10) # [2, 5, 'DevCode', 1.2, 5, 10]
```

```
lista.append([2,5]) # [2, 5, 'DevCode', 1.2, 5, [2, 5]]
```

Extend() agrega elementos independientes a una lista

```
lista.append([2,5]) # [2, 5, 'DevCode', 1.2, 5, 2, 5]
```

Remove() remueve un elemento de la lista

```
lista.remove(2)
```

Index() devuelve el número de índice del elemento pasado

```
lista.index('DevCode')
```

Count() cuantas veces un elemento se repite en la lista.

```
lista.count(5)
```

Reverse() invertir los elementos de la lista

```
lista.reverse() # [5, 1.2, 'DevCode', 5, 2]
```


Sorted() ordena los elementos de una lista

```
>>> bs=[5,2,4,2]
>>> cs=sorted(bs)
>>> bs
[5, 2, 4, 2]
>>> cs
[2, 2, 4, 5]
```

Sort() ordena los elementos de una lista

```
>>> ds=[5,3,4,5]
>>> ds.sort()
>>> ds
[2, 4, 5, 5]
```


Listas y cadenas

A partir de una cadena de caracteres, podemos obtener una lista con sus componentes usando la **función split**.

Si queremos obtener las palabras (separadas entre sí por espacios) que componen la cadena `c` escribiremos simplemente `c.split()`:

```
>>> c = " Una cadena con espacios "  
>>> c.split()  
['Una', 'cadena', 'con', 'espacios']
```

La *casi-inversa* de **split** es una función **join** que tiene la siguiente sintaxis:

`<separador>.join(<lista de componentes a unir>)`

y que devuelve la cadena que resulta de unir todas las componentes separadas entre sí por medio del *separador*.

```
>>> xs = ['aaa', 'bbb', 'cccc']  
>>> " ".join(xs) 'aaa bbb cccc'  
>>> ", ".join(xs) 'aaa, bbb, cccc'  
>>> "@ @".join(xs) 'aaa@ @bbb@ @cccc'
```



Programación I – Ing. Liliana Vega Bernal