**Listas en Python**

Las listas nos permiten tener valores asociados a una misma variable. Se dice entonces que las listas son *colecciones de datos.*

1. **Crear una lista**

Vamos a crear una lista, observa como los elementos de la lista van encerrados entre corchetes y separados por comas:

>>>calificaciones=[88, 90, 75, 60, 100, 98]

>>>calificaciones

**¿Qué te despliega Python? [88, 90, 75, 60, 100, 98]**

*En el Shell podemos ver la lista sin necesidad del comando print, sin embargo, en un programa es necesario print para desplegarla en pantalla*.

1. **INDEXING de Listas**

**¿Y si quisiéramos utilizar u obtener sólo uno de los números de la lista?**

|  |  |
| --- | --- |
| **Prueba lo siguiente:** | **¿Qué valor te despliega?** |
| >>>calificaciones[0] | **88** |
| >>>calificaciones[3] | **60** |
| >>>calificaciones[-1] | **98** |
| >>>calificaciones[-2] | **100** |

**Reflexiona: ¿Qué significa el número entre corchetes?**

**La indicacion de que forma llamar a la lista, mejor dicho de un elemento dentro de ellas**

**¿Qué puedes intuir acerca de colocar números negativos en los corchetes?**

**Que son los numeros que se van restando del ulitmo elemento y ese es el que se llama**

*El número entre corchetes después del nombre de la lista se conoce como* ***índice.*** *De manera que el* ***indexing*** *nos proporciona un elemento de la lista. El primer índice de la lista siempre es 0.*

1. **SLICING de listas**

En Python podemos obtener una sección de la lista a través de Slicing. Prueba lo siguiente:

Creamos una lista de números:

>>>nueva=[45, 48, 98, 2, 4, 6, 35, 12, 10, 7]

>>>nueva[0:3]

¿Qué sucedió?Me desplego los elementos designados del 0 al 3

|  |  |
| --- | --- |
| **Prueba lo siguiente:** | **¿Qué valor te despliega?** |
| >>>nueva[4:7] | **[4, 6, 35]** |
| >>>nueva[-3:] |  |
| >>>nueva[:5] |  |
| >>>nueva[:] |  |
| >>>nueva[1:8:2] |  |
| >>>nueva[6::-1] |  |
| >>>nueva[::-1] |  |

Reflexiona ¿Qué implica cada uno de los 3 datos que se ponen entre los corchetes, separados por dos puntos? ¿Qué significa no poner ningún número, es decir dejar vacío algún lugar entre los dos puntos?

*Slicing de listas producen una nueva lista con los elementos solicitados de la lista original.*

***La lista original NO se ve modificada.***

1. **Contenido de las listas**

En Python una lista puede contener datos de diferente tipo (sin embargo, lo más común es que sean datos del mismo tipo), creamos una lista con datos de diferentes tipos:

>>>sara=[“Rosas 44”, “Jalisco”, 24, 12345.78]

**Las listas son mutables.** Con la lista creada, realiza lo siguiente:

>>>sara[1]=”Guanajuato”

>>>sara

¿Qué sucedió con la lista? Acabas de comprobar que las listas son **mutables**, prueba ahora lo siguiente:

>>>sara.append(“Ingeniera”)

Despliega la lista, ¿Qué sucedió con la lista? ¿Comprendes qué hace “ append “?

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

1. **Métodos.**

Las listas tienen **métodos** que se pueden aplicar a ellas. Crea una lista de caracteres:

>>> lista1 = [’a’, ’b’, ’c’, ’d’, ‘a’]

Ejecuta la instrucción, despliega la lista y observa los resultados

|  |  |  |
| --- | --- | --- |
| **Método** | **Resultado** | **¿Qué hace el método?** |
| lista1.append(‘e’)  lista1 |  |  |
| lista1.insert(2, ’new’)  lista1 |  |  |
| lista1.count(‘a’) |  |  |
| lista1.remove(‘a’)  lista1 |  |  |
| lista1.sort()  lista1 |  |  |
| lista1.reverse()  lista1 |  |  |
| lista1.index(‘b’) |  |  |
| lista1.pop(2)  lista1 |  |  |
| lista1.pop( )  lista1 |  |  |

Crea la lista lista2=[3, 4, 7, 8, 12, 1, 5] y prueba las siguientes instrucciones:

|  |  |  |
| --- | --- | --- |
| **Método** | **Resultado** | **¿Qué hace el método?** |
| lista2.sort()  lista2 |  |  |
| lista2.sort(reverse=True)  lista2 |  |  |

¿Cuál es el formato para aplicar un método a una lista?

¿Cómo puedes saber qué parámetros recibe un método?

1. **Funciones**.

A las listas se les pueden aplicar funciones. Veamos las más utilizadas:

Crea la lista datos=[12,34,56,2,98,45] y prueba las siguientes funciones:

|  |  |  |
| --- | --- | --- |
| **Función** | **Resultado** | **¿Qué hace la función?** |
| len(datos) |  |  |
| max(datos) |  |  |
| min(datos) |  |  |
| sum(datos) |  |  |

Reflexiona ¿Cuál es la diferencia entre aplicar un método o una función a una lista?

1. **Lista vacía**

Declaración: >>>listaV=[]

Qué sucede si pides su primer elemento: >>>listaV[0]

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

y así: >>>listaV[-1] \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

1. **Usos.**

Imagina todo lo que puede servirte utilizar listas para la resolución de problemas.