



# Fundamentos de programación en PYTHON

---

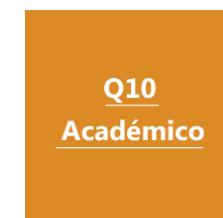
Carrera de Administración de Redes y Comunicaciones

# ■ Alianzas Estratégicas ■

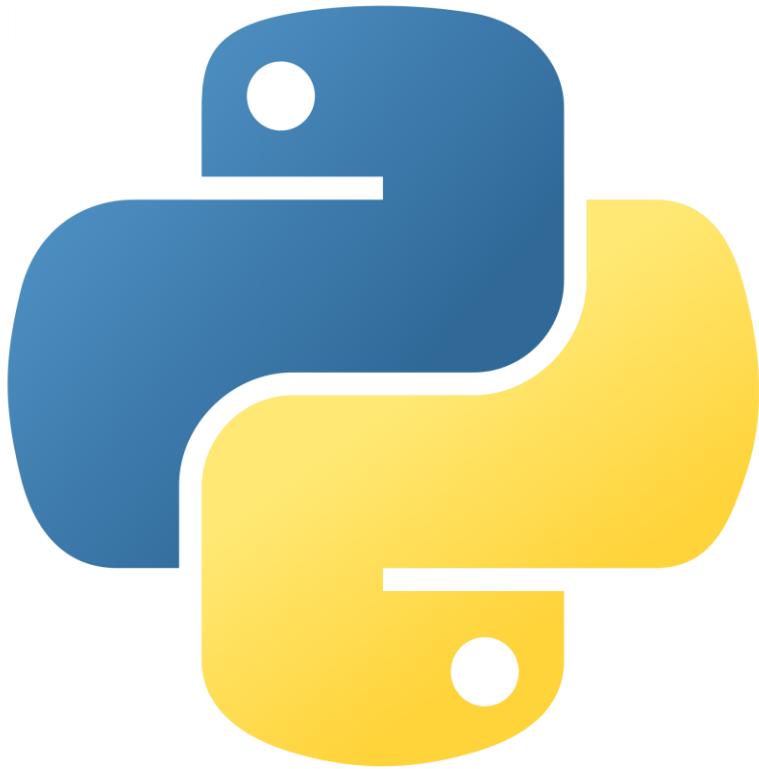




Google Classroom

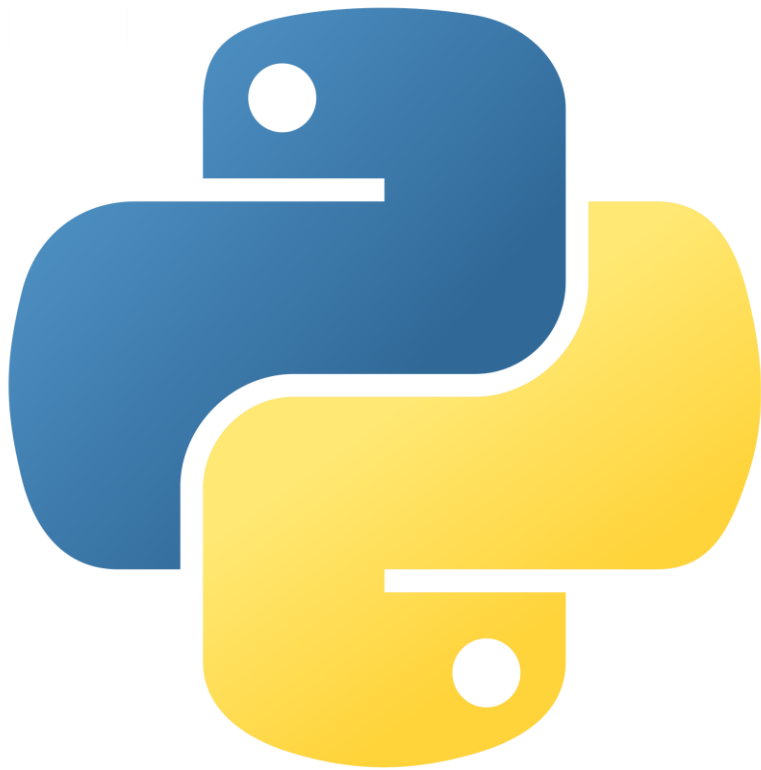


# Sobre el curso



El curso está orientado a la enseñanza de fundamentos de Programación en el lenguaje Python v 3.0., bajo la modalidad definida por la Academia de Programming de Cisco. Esta modalidad consiste en el desarrollo de la currícula PCAP dividida en 8 módulos.

Durante el curso se desarrollarán actividades de revisión del contenido académico y laboratorios prácticos utilizando el emulador de Python que nos provee el mismo Cisco (SandBox) o de algún otro IDE para escritorio, así como el desarrollo de exámenes y actividades orientadas al entendimiento de estos conceptos de acuerdo con los temas que considera Cisco en este curso.



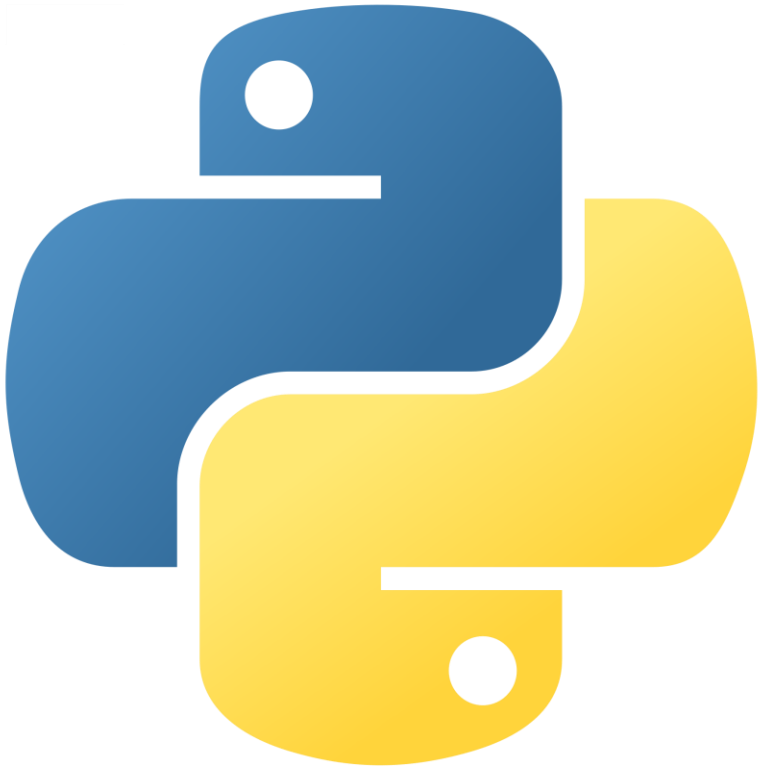
## SEMANA 4

### LISTAS

#### LOGRO DE LA SESION

Al concluir la sesión, los participantes desarrollan una comprensión completa de las listas como estructuras de datos fundamentales. Esto incluye la creación de listas, comprender su sintaxis y cómo acceder a elementos individuales mediante índices. Además, como agregar, eliminar y modificar elementos, y la aplicación de técnicas de segmentación para acceder a subconjuntos de una lista.

# Contenido

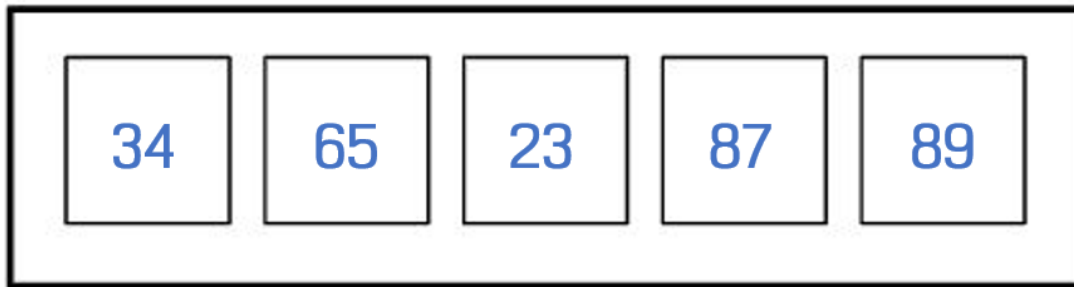


- Introducción
- Creación de listas
- Recorrido de una lista
- Funciones para la gestión de listas
- Ejercicios
- Conclusiones

# Introducción



```
Lista = [ 34, 65, 23, 87, 89 ]
```



Una lista es una colección ordenada de datos a la que se puede acceder utilizando índices numéricos.

Este tipo de dato es muy versátil y puede contener cualquier combinación de datos, como cadenas, flotantes, enteros, incluso otras listas.



# Creación de listas (1/2)



```
nombre_lista = [ dato1, dato2, dato3, ... ]
```

```
lista1 = [45,87,23,45]
print(lista1)
lista2 = ["Chiclayo","Cuzco","Arequipa","Huancayo"]
print(lista2)
lista3 = [65,"Gustavo Coronel",76.34]
print(lista3)
lista4 = [43,67,23,[54,67,23,54]]
print(lista4)
```





## Creación de listas (2/2)



```
lista = [43,67,23,[54,67,23,54]]  
print("Elemento [0]: ",lista[0])  
print("Elemento [3]: ",lista[3])  
print("Elemento [3,1]: ",lista[3][1])
```





# Recorrido de una lista con for (1/2)

```
# Datos
lista = [43,67,23,54,89,34]
# Proceso
suma = 0;
for i in range(0,len(lista)):
    suma += lista[i]
# Reporte
print("Lista:",lista)
print("Suma:",suma)
```



## Recorrido de una lista con for (2/2)

```
# Datos
lista = [43,67,23,54,89,34]
# Proceso
suma = 0;
cont = 0;
for e in lista:
    suma += e
    cont += 1
# Reporte
print("Lista:",lista)
print("Tamaño:",cont)
print("Suma:",suma)
```



# Funciones para la gestión de listas (1/4)

FUNCIÓN	DESCRIPCIÓN	EJEMPLO	RESULTADO
clear	Elimina todos los elementos de una lista.	<pre>l = [76,20,67] l.clear() print(l)</pre>	[ ]
len	Retorna el tamaño de una lista.	<pre>l = [12,54,67] t = len(l) print(t)</pre>	3
sort	Ordena los elementos de una lista.	<pre>l = [76,20,67] l.sort() print(l)</pre>	[20, 67, 76]
reverse	Invierte los elementos de una lista.	<pre>l = [10,20,30] l.reverse() print(l)</pre>	[30, 20, 10]





# Funciones para la gestión de listas (2/4)

FUNCIÓN	DESCRIPCIÓN	EJEMPLO	RESULTADO
append	Añade un nuevo elemento a la lista	<pre>l = [76,20,67] l.append(54) print(l)</pre>	[76, 20, 67, 54]
max	Retorna el elemento de mayor valor	<pre>l = [76,20,67] m = max(l) print(m)</pre>	76
min	Retorna el elemento de menor valor	<pre>l = [76,20,67] l.sort() print(l)</pre>	[20, 67, 76]
insert	Inserta un elemento en una posición específica.	<pre>l = [10,20,30] l.insert(1,40) print(l)</pre>	[10, 40, 20, 30]



# Funciones para la gestión de listas (3/4)



FUNCIÓN	DESCRIPCIÓN	EJEMPLO	RESULTADO
<code>extend</code>	Añade los elementos de otra lista.	<pre>l = [76,20] l.extend([90,74]) print(l)</pre>	[76, 20, 90, 74]
<code>index</code>	Busca un elemento y retorna la primera posición.	<pre>l = [76,20,67,20] i = l.index(20) print(i)</pre>	1
<code>count</code>	Se utiliza para contar el número de veces que un elemento aparece en una lista.	<pre>l = [76,20,67,20,20] n = l.count(20) print(n)</pre>	3

# Funciones para la gestión de listas (4/4)



FUNCIÓN	DESCRIPCIÓN	EJEMPLO	RESULTADO
pop	Elimina y devuelve el elemento en una posición específica, por defecto, el último.	<pre>l = [76,20,67,20,20] x = l.pop(2) print(l) print(x)</pre>	<pre>[76, 20, 20, 20] 67</pre>
Copy	Crea una copia de una lista.	<pre>l = [76,20,67,20] c = l.copy() print(c)</pre>	<pre>[76, 20, 67, 20]</pre>
remove	Elimina la primera ocurrencia de un elemento.	<pre>l = [76,20,67,20,20] l.remove(20) print(l)</pre>	<pre>[76, 67, 20, 20]</pre>





# Ejercicios

Debes aplicar los conceptos que se bien utilizando: Entrada de datos ⇨ Proceso ⇨ Reporte



# Ejercicio 1



Elaborar un programa que le permita ingresar al usuario 5 notas.

Luego debe calcular el promedio de las 3 mejores notas.

## Ejercicio 2



Desarrollar un programa que genere N números enteros positivos entre 20 y 99 inclusive.

El valor para N lo debe ingresar el usuario.

Luego debe encontrar la media y la moda.

# Ejercicio 3



Desarrollar un programa que solicite al usuario una palabra y luego encuentre las veces que se repite cada vocal.

# Ejercicio 4



Desarrollar un programa que genere dos listas de números enteros de tamaño  $N$ .

El valor de  $N$  lo debe solicitar al usuario.

Luego debe encontrar la lista de números comunes.

# Ejercicio 5



Desarrollar un programa que genere una lista de 5 números enteros.

Luego debe eliminar de la lista el numero mayor y el numero menor.

Finalmente debe encontrar el promedio de los 3 números que quedan en la lista.

# Ejercicio 6



Desarrollar un programa que lea el nombre y precio de 5 productos.

Luego debe mostrar el nombre de los productos con mayor y menor precio.



# Conclusiones

- Las listas en Python pueden contener elementos de diferentes tipos de datos y pueden modificarse fácilmente después de su creación.
- Los elementos de una lista se almacenan en orden y se accede a ellos mediante índices.
- Las listas en Python se pueden recorrer fácilmente mediante bucles for, lo que facilita el procesamiento de cada elemento de la lista de manera individual.

GRACIAS  
TOTALES