**Listas:**

lista = [1, “string”, True, False, 121, “Python”]

**len(lista)** Devuelve la cantidad de elementos de la lista.

**sum(lista)** Devuelve la suma de los elementos de la lista.

**max(lista)** Devuelve el valor más alto de la lista.

**min(lista)** Devuelve el valor más bajo de la lista.

**print(lista[X])** Hace un print del elemento en la posición X de la lista

**lista.append(X)** agrega el elemento X al final de la lista.

**lista.insert(Y, X)** inserta el elemento X en la posición Y.

**lista.pop()** Elimina el último elemento de la lista.

**lista.pop(X)** Elimina el elemento en la posición X de la lista

**lista.remove(X)** Elimina el primer X que se encuentre en la lista (de izquierda a derecha)

**lista.clear()** borra todos los elementos de la lista

**lista.index(X)** devuelve la primera posición de la primera X de la lista (de izquierda a derecha)

**lista.count(X)** Devuelve cuantas veces aparece X en la lista.

**lista.sort()** Ordena la lista por mayúsculas y alfabéticamente

**lista.reverse()** invierte los elementos de la lista

**for i in lista:**

**print(i)**

Devuelve los elementos de la lista

**Diccionarios:**

diccionario = {

“nombre” : “Gabriel”

“edad” : 20

}

**len(diccionario)** Devuelve la cantidad de elementos en el diccionario (clave : valor = 1 elemento)

**diccionario.get(clave)** Devuelve el valor asignado a esa clave, si no existe devuelve None

**diccionario[clave]** devuelve el valor de una clave

**diccionario[clave]** **= valor** Agrega o modifica un valor

**diccionario.pop(clave)** elimina una clave del diccionario

**diccionario.clear()** elimina todos los elementos del diccionario

**diccionario.items()** devuelve una lista de tuplas (clave, valor)

**for clave, valor in diccionario.items():**

**print(clave, “ = ”, valor)**

Devuelve la clave y valor de todos los elementos en el diccionario

**Función isupper()**

La función any(c.isupper() for c in txtIn) en Python verifica si **algún carácter en la cadena texto es una letra mayúscula**.

**Desglose paso a paso:**

1. **for c in txtIn**: Esto recorre cada carácter c de la cadena txtIn.
2. **c.isupper()**: Es un método de Python que devuelve True si el carácter c es una letra en mayúscula (como 'A', 'Z', etc.), y False en caso contrario.
3. **(c.isupper() for c in txtIn)**: Esto es un **generador**, que produce una secuencia de valores True o False dependiendo de si cada carácter es mayúscula.
4. **any(...)**: La función any() devuelve True si **al menos uno** de los elementos del generador es True. Si **ninguno** es True (es decir, no hay letras mayúsculas), devuelve False.

**Ejemplo:**

txtIn = "abc123"

resultado = any(c.isupper() for c in txtIn)

print(resultado) # False

txtIn = "abcD123"

resultado = any(c.isupper() for c in txtIn)

print(resultado) # True

**Función isdigit()**

La expresión any(c.isdigit() for c in txtIn) en Python verifica si **algún carácter en la cadena txtIn es un dígito numérico (del 0 al 9)**.

**Desglose paso a paso:**

1. **for c in txtIn**: Recorre cada carácter c en la cadena txtIn.
2. **c.isdigit()**: Este método devuelve True si el carácter c es un **dígito decimal** (como '0', '1', ..., '9'). Si no lo es, devuelve False.
3. **(c.isdigit() for c in txtIn)**: Esto crea un **generador** que produce una secuencia de True o False dependiendo de si cada carácter es un dígito.
4. **any(...)**: Devuelve True si **al menos uno** de los elementos del generador es True, es decir, si **hay al menos un número** en la cadena. Si no hay ninguno, devuelve False.

**Ejemplo:**

txtIn = "HolaMundo"

resultado = any(c.isdigit() for c in txtIn)

print(resultado) # False

txtIn = "Hola123"

resultado = any(c.isdigit() for c in txtIn)

print(resultado) # True

**Función isalpha()**

La expresión cadena.isalpha() en Python verifica si **todos los caracteres de la cadena cadena son letras del alfabeto** y **no está vacía**.

**Detalles clave:**

* + **Devuelve True** si: Todos los caracteres son **letras** (A–Z o a–z, también incluye letras de otros alfabetos como á, ñ, etc.).
  + La cadena **no está vacía**.
  + **Devuelve False** si: Hay **espacios**, **números**, **símbolos** o **caracteres vacíos** en la cadena.

**Ejemplos:**

"Hola".isalpha() # True

"HolaMundo".isalpha() # True

"Hola123".isalpha() # False (tiene números)

"Hola Mundo".isalpha() # False (tiene espacio)

"".isalpha() # False (cadena vacía)

**Función strip()**

La expresión txtIn.strip() en Python se utiliza para **eliminar los espacios en blanco al principio y al final** de una cadena txtIn.

**¿Qué elimina exactamente?**

strip() elimina:

* Espacios (" ")
* Tabulaciones (\t)
* Saltos de línea (\n)
* Otros caracteres en blanco (como \r)

Por defecto, no elimina caracteres **dentro** de la cadena, solo en los **bordes** (inicio y fin).

**Ejemplos:**

txtIn = " Hola mundo "

print(txtIn.strip()) # "Hola mundo"

txtIn = "\n\t Texto con espacios y tabulaciones \t\n"

print(txtIn.strip()) # "Texto con espacios y tabulaciones"

txtIn = "SinEspacios"

print(txtIn.strip()) # "SinEspacios" (no cambia nada)

txtIn = " "

print(txtIn.strip()) # "" (cadena vacía)

**Función zfill ()**

La función zfill() **rellena una cadena (str) con ceros (0) a la izquierda** hasta alcanzar una longitud especificada.

Sintaxis:

cadena.zfill(ancho)

* **cadena**: la cadena de texto a la que se le quiere aplicar el relleno.
* **ancho**: longitud total deseada de la cadena resultante. Si la longitud de la cadena original es mayor o igual al ancho, **no se agrega nada**.

🔹 Ejemplos:

print("42".zfill(5)) # '00042' print("7".zfill(3)) # '007'

print("python".zfill(10)) # '0000python' print("-5".zfill(4)) # '-005'

Nota importante:

* Si la cadena comienza con un **signo positivo (+)** o **negativo (-)**, los ceros se agregan **después del signo**:

print("-42".zfill(5)) # '-0042' print("+3".zfill(4)) # '+003'

🔹 ¿Para qué se usa comúnmente?

* Para formatear números con ceros a la izquierda (como en códigos, IDs o fechas).
* Para generar salidas con longitud uniforme.