

ENTRADA Y SALIDA DE DATOS

```
alert(title="TEST", message="Hola")
prompt(title="TEST", prompt="Ingresa tu nombre")
question(title="TEST", message="¿Desea continuar?")
combobox => cuadro con varias opciones, para elegir una.
values=["Día", "Noche"]

get() => self.txt_nombre.get()
delete() => self.txt_nombre.delete(0, 100000)
insert() => self.txt_nombre.insert(0, nombre)

format() => formatea el mensaje que se quiere mostrar, permitiendo insertar variables.
numero = 9.213124
mensaje = "El numero es: {0}".format(numero)
alert(title="TEST", message=mensaje) -> "El número es "9.213124"Se puede limitar la cantidad de decimales mostrados con {0:.2f}. Esto mostraría entonces 9.21
```

TIEMPO

```
self.id_tiempo = self.after(tiempo, nombre_función)
para indicar después de cuánto tiempo llamar a la función (NO SE EJECUTA, solo se nombra). El tiempo se expresa en milisegundos: 3 s = 3000 ms
self.after_cancel(self.id_tiempo) => para cancelar el llamado

inicio_temporizador = time.time()
final_temporizador = time.time()
tiempo_transcurrido = final_temporizador - inicio_temporizador
```

GENERAR NUMEROS ALEATORIOS

```
numero_aleatorio = random.randint(1, n) => genera un número aleatorio del 1 al n
numero_aleatorio = random.randrange(1, n) => Genera un número aleatorio dentro de un rango (1, n-1)
```

LISTAS

```
Estructura: lista = [4, "Camila", "rojo", 2000, True]

random.choice(lista) => selecciona un elemento de la lista al azar
lista[2] => elemento en posición 2 ("rojo")
lista.append(elemento) => inserta un elemento en la última posición de la lista.
self.lista.insert(0, elemento) => agrega un elemento al principio de la lista
self.lista_ventana.clear() => limpia la lista
len(lista) => muestra el largo de la lista. len = length
```

CÓDIGO ASCII

Para validar que el texto ingresado incluya solo letras:

letra >= "A" and letra <= "Z" or letra >= "a" and letra <= "z"

"A" = 65 | "Z" = 90 | "a" = 97 | "z" = 122

Recorrer dos listas y obtener los valores de ambas según el indice

```
for indice in range(cantidad_articulos):
    if(minimo == None or self.lista_precio[indice] < minimo):
        minimo = self.lista_precio [indice]
        indice_minimo=indice
```

OTRA FORMA DE HACERLO:

```
for indice in range(cantidad_articulos):
    if(indice==0 or self.lista_precio [indice] < self.lista_precio [indice_minimo]):
        indice_minimo = indice
```

Uso de los flags si no se ingresa ningún valor en las cajas de texto

```
flag_nombre_articulo = True
flag_precio_articulo = True
if (len(nombre_articulo) == 0 or len(precio_articulo) == 0):
    flag_nombre_articulo = False
    flag_precio_articulo = False
mensaje = "Todos los campos deben ser completados"
```

SÍMBOLOS

== => Igual que	!= => distinto de
< => menor que	<= => menor o igual que
> => mayor que	>= => mayor o igual que
or / => o	and / & => y

INSTRUCCIÓN IF

```
Estructura:
if (5 > 10):
    alert(title="TEST", message="Es mayor")
elif (5 == 10):
    alert(title="TEST", message="Es igual")
else:
    alert(title="TEST", message="Es menor")
```

INSTRUCCIÓN MATCH

```
Estructura:
match(mes):
    case "Enero":
        mensaje = "Estamos en Enero"
    case "Febrero":
        mensaje = "Estamos en Enero"
    case _:
        mensaje = "Ni Enero ni Febrero"
alert(title="TEST", message=mensaje)
```

INSTRUCCIÓN FOR

```
Estructura:
for numero in range(1, numero_ingresado + 1):
    if (numero_ingresado % numero == 0):
        contador_divisores +=1
```

INSTRUCCIÓN WHILE

```
Estructura:
while(True):
    if (numero > 5):
        break
    else:
        self.contador += 1
```

BOTONES

Habilitar/deshabilitar un botón:

```
self.btn.configure(state="normal") o
self.btn.configure(state="enabled")
self.btn.configure(state="disabled")
```

Mostrar/ocultar un botón:

```
self.btn.grid(row=2, pady=10, columnspan=2, sticky="nsew")
self.btn.grid_forget()
```

```
Validar que el texto incluya solo letras
flag_apellido_ok = False
apellido_str = ""

while(flag_apellido_ok == False):
    while(apellido_str == ""):
        apellido_str = prompt(title="", prompt="Apellido")

flag_apellido_ok = True
for letra in apellido_str:
    if(letra < "A" or letra > "z" or (letra > "Z" and letra < "a")):
        flag_apellido_ok = False
        apellido_str = ""
        break
```

Para sumar valores en un bucle => suma = suma + valor

Para sacar el promedio => promedio = suma / cantidad_elementos

Fórmula para saber si un número es primo

```
for numero in range(numero_ingresado):
    divisores = 0
    if (numero_ingresado % numero == 0)
        divisor += 1
    if (divisores == 2):
        mensaje = "Es primo"
```