

- [Diferencia entre un contador y un acumulador](#)
- [Acumulador](#)
- [COMO RESTRINGIR A SOLO NUMEROS???](#) 🤔 🤖
  - [METODOS DE LA CLASE](#) La clase str
  - La clase str en Python representa cadenas de caracteres. Es uno de los tipos de datos fundamentales del lenguaje y se utiliza para almacenar texto.
  - [SOLO NUMEROS](#)
  - [SOLO TEXTO](#) 📄
- [Para Numeros INT](#) 🕒
- [EJERCICIOS](#)

# Diferencia entre un contador y un acumulador

---

## Contador:

- Se utiliza para **contar la cantidad de veces** que ocurre un evento específico.
- Se inicializa a un valor inicial (comúnmente 0).
- Se incrementa en 1 cada vez que ocurre el evento que se está contando.
- Se utiliza principalmente dentro de bucles. ejemplo :

```
contador += 1
#ó
contador = contador + 1
```

## Acumulador

---

Se utiliza para **acumular un valor** a lo largo de un proceso.

- Se inicializa a un valor inicial (comúnmente 0).
- Se actualiza sumándole un nuevo valor cada vez que se repite el proceso.

```
suma += variable
#ó
suma = suma + variable
```

# COMO RESTRINGIR A SOLO NUMEROS???



## METODOS DE LA CLASE La clase str

Estos métodos te permiten manipular el texto de diferentes maneras, como verificar si contiene letras mayúsculas, minúsculas o números, buscar subcadenas, unir cadenas, convertir mayúsculas a minúsculas y viceversa, etc.

**La clase str en Python representa cadenas de caracteres. Es uno de los tipos de datos fundamentales del lenguaje y se utiliza para almacenar texto.**

## SOLO NUMEROS

```
entrada=input("Ingrese un número: ")
print (type(entrada))
if entrada.isdigit():
    print(f"el numero ingresado es {entrada}.")
else:
    print("Error: Debe ingresar un número válido.")
    exit()
```

## SOLO TEXTO



```
entrada=input("Ingrese texto: ")

if entrada.isalpha() or entrada.isspace():

    print(f"el text ingresado es {entrada}.")

else:

    print("Error: Debe ingresar solo texto (letras o espacios).")
```

Algunos ejemplos:

Método	Descripción	Ejemplo
<code>isalpha()</code>	Verifica si todos los caracteres son letras.	<code>texto = "palabra"; es_letra = texto.isalpha(); print(es_letra)</code> (Salida: True)
<code>isdigit()</code>	Verifica si todos los caracteres son dígitos.	<code>texto = "12345"; es_digito = texto.isdigit(); print(es_digito)</code> (Salida: True)
<code>isalnum()</code>	Verifica si todos los caracteres son alfanuméricos (letras y números).	<code>texto = "texto123"; es_alfanumerico = texto.isalnum(); print(es_alfanumerico)</code> (Salida: True)
<code>islower()</code>	Verifica si todos los caracteres son minúsculas.	<code>texto = "palabra en minúsculas"; es_minuscula = texto.islower(); print(es_minuscula)</code> (Salida: True)
<code>isupper()</code>	Verifica si todos los caracteres son mayúsculas.	<code>texto = "PALABRA EN MAYÚSCULAS"; es_mayuscula = texto.isupper(); print(es_mayuscula)</code> (Salida: True)
<code>isspace()</code>	Verifica si todos los caracteres son espacios en blanco.	<code>texto = " "; es_espacio = texto.isspace(); print(es_espacio)</code> (Salida: True)
<code>startswith()</code>	Verifica si la cadena comienza con una subcadena específica.	<code>texto = "Hola mundo"; empieza_con_hola = texto.startswith("Hola"); print(empieza_con_hola)</code> (Salida: True)
<code>endswith()</code>	Verifica si la cadena termina con una	<code>texto = "Hola, mundo, mundo"; termina_con_mundo = texto.endswith("mundo"); print(termina_con_mundo)</code> (Salida: True)

Método	Descripción	Ejemplo
	subcadena específica.	
<code>find()</code>	Busca la primera aparición de una subcadena dentro de la cadena.	<pre> texto = "Hola, mundo"; posicion_hola = texto.find("Hola"); print(posicion_hola)(Salida: 0) </pre>
<code>rfind()</code>	Busca la última aparición de una subcadena dentro de la cadena.	<pre> texto = "Hola, mundo, mundo"; posicion_ultimo_mundo = texto.rfind("mundo"); print(posicion_ultimo_mundo)(Salida: 15) </pre>
<code>count()</code>	Cuenta el número de apariciones de una subcadena dentro de la cadena.	<pre> texto = "Hola, Hola, mundo"; numero_de_holas = texto.count("Hola"); print(numero_de_holas)(Salida: 2) </pre>
<code>replace()</code>	Reemplaza todas las apariciones de una subcadena por otra subcadena.	<pre> texto = "Hola, mundo. Hola, Tierra"; texto_reemplazado = texto.replace("Hola", "Adiós"); print(texto_reemplazado)(Salida: Adiós, mundo. Adiós, Tierra) </pre>
<code>strip()</code>	Elimina los espacios en blanco al principio y al final de la cadena.	<pre> texto = " Hola mundo "; texto_sin_espacios = texto.strip(); print(texto_sin_espacios)(Salida: Hola mundo) </pre>
<code>lstrip()</code>	Elimina los espacios en blanco al	<pre> texto = " Hola mundo"; texto_sin_espacios_izquierda = texto.lstrip(); </pre>

Método	Descripción	Ejemplo
	principio de la cadena.	<code>print(texto_sin_espacios_izquierda)</code> (Salida: Hola mundo)
<code>rstrip()</code>	Elimina los espacios en blanco al final de la cadena.	<code>texto = "Hola mundo ";</code> <code>texto_sin_espacios_derecha =</code> <code>texto.rstrip();</code> <code>print(texto_sin_espacios_derecha)</code> (Salida: Hola mundo)
<code>upper()</code>	Convierte todos los caracteres de la cadena a mayúsculas.	<code>texto = "hola mundo"; texto_mayusculas</code> <code>= texto.upper();</code> <code>print(texto_mayusculas)</code> (Salida: HOLA MUNDO)
<code>lower()</code>	Convierte todos los caracteres de la cadena a minúsculas.	<code>texto = "HOLA MUNDO"; texto_minusculas</code> <code>= texto.lower();</code> <code>print(texto_minusculas)</code> (Salida: hola mundo)
<code>capitalize()</code>	Convierte el primer caracter de la cadena a mayúscula y deja el resto en minúsculas.	<code>texto = "hola mundo";</code> <code>texto_capitalizado =</code> <code>texto.capitalize();</code> <code>print(texto_capitalizado)</code> (Salida:

## Para Numeros INT

Método	Descripción	Ejemplo
<code>isinstance(valor, int)</code>	Verifica si el valor es un número entero (int).	<code>valor = 10; es_entero =</code> <code>isinstance(valor, int);</code> <code>print(es_entero)</code> (Salida: True)
<code>type(valor) is int</code>	Verifica si el tipo de dato del valor es int.	<code>valor = 10; es_entero =</code> <code>type(valor) is int;</code> <code>print(es_entero)</code> (Salida: True)
<code>.isdigit()</code>	Verifica si el valor es una cadena	<code>valor_string = "12345";</code> <code>es_digito =</code>

Método	Descripción	Ejemplo
	que contiene solo dígitos (números).	<code>valor_string.isdigit();</code> <code>print(es_digito)</code> (Salida: True)
<code>int(valor, base)</code>	Convierte un valor a un número entero (int), especificando la base numérica si es necesario.	<code>valor_string = "10";</code> <code>numero_entero =</code> <code>int(valor_string);</code> <code>print(numero_entero)</code> (Salida: 10)
<code>try...except</code> <code>ValueError:</code>	Intenta convertir un valor a un número entero (int) y maneja el error si no es posible.	<code>valor_string = "hola"; try:</code> <code>numero_entero =</code> <code>int(valor_string);</code> <code>print(numero_entero) except</code> <code>ValueError: print("Error: El</code> <code>valor no es un número entero")</code> (Salida: Error: El valor no es un número entero)

- Las funciones `isinstance()` y `type()` son más confiables para verificar si un valor es un número entero, ya que no dependen de la representación como cadena.
- La función `.isdigit()` es útil para validar cadenas de texto que contienen solo dígitos.
- La función `int()` se puede usar para convertir valores a números enteros, pero es importante manejar posibles errores de conversión.

mas ejemplos:

```
##* SOLO NUMEROS
entrada = input("Ingrese un número: ")
print (type(entrada))
print (entrada.isdigit())

if entrada.isdigit() :
    print(f"el numero ingresado es {entrada}.")
else:
    print(f"Error: Debe ingresar un número válido. fijate lo que pusiste!{
        entrada}")
```

```
exit
```

```
entrada = input("Ingrese un texto: ")
print (type(entrada))
print (entrada.isdigit())

if not entrada.isdigit() :
    print(f"el text ingresado es {entrada}.")
else:
    print(f"Error: Debe ingresar un texto válido. fijate lo que pusiste!{
        entrada}")
    exit
```

```
## SOLO TEXTO
```

```
entrada = input("Ingrese texto: ")
if entrada.isalpha() or entrada.isspace():
    print(f"el text ingresado es {entrada}.")
else:
    print("Error: Debe ingresar solo texto (letras o espacios).")
```

```
# METODOS DE LA CLASE La clase str
```

```
# Estos métodos te permiten manipular el texto de diferentes maneras, como
verificar si contiene letras mayúsculas, minúsculas o números, buscar subcadenas,
unir cadenas, convertir mayúsculas a minúsculas y viceversa, etc.
```

```
# La clase str en Python representa cadenas de caracteres. Es uno de los tipos de
datos fundamentales del lenguaje y se utiliza para almacenar texto.
```

```
# algunos ejemplos de cómo se usan estos métodos:
```

```
# capitalize(): Convierte el primer caracter de la cadena a mayúscula y deja el
resto en minúsculas.
```

```
texto = "hola mundo"
texto_capitalizado = texto.capitalize()
print(texto_capitalizado) # Salida: Hola mundo
```

```
# upper(): Convierte todos los caracteres de la cadena a mayúsculas.
```

```
texto = input("ingrese un texto:")
print(texto.upper()) # Salida: HOLA MUNDO
```

```
#lower(): Convierte todos los caracteres de la cadena a minúsculas.
```

```
texto = "HOLA MUNDO"
texto_minusculas = texto.lower()
print(texto_minusculas) # Salida: hola mundo
```

```
#isalnum(): Verifica si todos los caracteres de la cadena son alfanuméricos (letras
y números).
```

```
texto = "texto123"
es_alfanumerico = texto.isalnum()
print(es_alfanumerico) # Salida: True
```

```
texto = "texto con espacios"
es_alfanumerico = texto.isalnum()
print(es_alfanumerico) # Salida: False
```

```
#isalpha(): Verifica si todos los caracteres de la cadena son letras.
```

```
texto = "palabra"
```

```
es_letra = texto.isalpha()
print(es_letra) # Salida: True

texto = "palabra123"
es_letra = texto.isalpha()
print(es_letra) # Salida: False

#isdigit(): Verifica si todos los caracteres de la cadena son dígitos (números).
texto = "12345"
es_digito = texto.isdigit()
print(es_digito) # Salida: True

texto = "texto con seis"
es_digito = texto.isdigit()
print(es_digito) # Salida: False

# mas data : https://docs.python.org/es/3/library/string.html

valor = 10

if isinstance(valor, int):
    print(f"El valor {valor} es un número entero.")
else:
    print(f"El valor {valor} no es un número entero.")
```

## EJERCICIOS

---

- Realizar un programa que permita ingresar solo texto
- Realizar un programa que permita ingresar solo texto y los muestre todo en mayuscula
- Realizar un programa que permita ingresar solo numeros, si no se cumple este requisito, que me de la hasta 2 oportunidades mas y me muestre la cantidad de veces que lo hice mal.