



*Ciencia de Datos*

## *Taller 05 Python*

### **Formatos**

La forma de visualizar la información es 1 de los puntos importa antes en la informática para poder mostrar los datos numéricos o de texto combinados en mensajes únicos y con forma de presentación diferentes requiere de unos métodos y funciones para formatear su presentación.

Cuando se manipula información estamos implícitamente manipulando datos de tipos diferentes, al manipular cada dato, dependiendo del tipo que sea se tienen unas funciones especiales para transformar consultar o manipular esos datos más explícitamente sobre las variables que los contienen. Hoy recordamos que una de las habilidades en programación eso de manipular las cadenas de texto.

#### Objetivos

hola identificar varias maneras de presentación de la información combinando variables de diferente tipo, utilizando los métodos de formateo pues sentados hoy con el lenguaje.

#### Datos numéricos

las variables numéricas se utilizan para representar cifras constantes o variables, numéricas, hoy se pueden hacer operaciones entre ellas, se pueden hacer conversión de tipo o “casting”, hola los números pueden ser enteros de cómo flotante o decimales y números imaginarios. Hoy la precisión de los datos es tan importante en las operaciones como en la visualización para la comprensión humana.

## Ejercicios

Codificar los ejercicios por separado para visualizar las diferentes formas de presentación o formateo de los Strings.

## FORMATO DE VARIABLES

Posibilitan la salida de información con cadenas de texto

### Ejercicio

```
year = 2016
event = 'Referendum'
print(f'Results of the {year} {event}')
# cada variable se reemplaza entre {}
# siempre y cuando el string anteceda con f
```

## Método format()

#El uso básico del método str.format(), sintaxis:

```
cadena_formateada = cadena_original.format(valor1, valor2, ...)
```

### Ejercicio:

```
votos = 42_572_654 #numero entero
novotos = 12_333_300
print(type(votos))
print(votos)
#variables
print(f"votantes: {votos} no votantes {novotos}")

print("-----")
no_votos = 43_132_495
perc = votos / (votos + novotos)
print('{:-10} YES votes{:2.3%}'.format(votos, perc))
#{: -10} Numero de espacios para la variable
```

- : Indica el inicio del formato.
- : indica alineación a la izquierda

10 tamaño del campo ancho mínimo de 10

{ } : Delimitan la expresión que se va a formatear. En este caso, no hay una variable dentro de las llaves, por lo que se asume que el valor a formatear se proporcionará como argumento a la función de formato.

2 : Indica el ancho total del campo formateado. En este caso, el campo tendrá 2 caracteres de ancho.

. : Separa la parte entera de la parte decimal.

3 : Indica el número de dígitos decimales que se mostrarán. En este caso, se mostrarán 2 dígitos decimales.

% : Indica que el valor se debe formatear como un porcentaje.

Ejercicios:

```
print("hola {} como esta el{}".format("juan", "trabajo"))
```

-----

```
numero = 0.5
```

```
# Se formatea y mostrara el valor como un porcentaje
```

```
print("El valor es: {:.2%}".format(numero))
```

```
#-----
```

```
#Asigna el valor en el momento de formatear
```

```
txt = "Por solo {price:.2f} pesos!"
```

```
print(txt.format(price = 49))
```

## Separadores de 1000

```
area = 1973.9208802178716
print('El area es: {:.2f}'.format(area))
#separadores de 1000 y 2 decimales
```

## Mas Formatos.

```
numero = 0.5
# Se formatea el valor como un porcentaje
print("El valor es: {:.2%}".format(numero))
area = 1973.9208802178716
print('El area es: {:.2f}'.format(area))
#separadores de 1000 y 2 decimales

print('El area es: {:.2f}'.format(area))
#reemplaza la coma (,) por asterisco (*)
print('El area es: {:.2f}'.format(area).replace(",", ""))
#El area es: 1.973,92
```

Desde la version 3.6, las expresiones más usadas como se muestra a continuación:

```
print(f'El area es: {area:,.2f}')
```

Fecha Creación	Enero 23 2024
Responsable	Plinio Neira Vargas
Revisado por	Sonia Escobar
Fecha Revisión	Enero 25 2024