

# LENGUAJES y HERRAMIENTA PARA CIENCIAS DE DATOS I

## Formateo de cadenas



# Formatted String Litteral

- f'Cadena{expresion:<especificadorFormato>}'
  - ◆ Expresión puede ser una variable o valor literal

```
mLuque@hydrogen: ~  
File Edit View Search Terminal Help  
mLuque@hydrogen:~$ python3  
Python 3.6.9 (default, Jan 26 2021, 15:33:00)  
[GCC 8.4.0] on linux  
Type "help", "copyright", "credits" or "license" for more information.  
>>> aciertos=55  
>>> fallos=34  
>>> f'El algoritmo ha clasificado correctamente {aciertos} patrones. Un {aciertos/(aciertos+fallos)}'  
'El algoritmo ha clasificado correctamente 55 patrones. Un 0.6179775280898876'  
>>> 
```

# Formatted String Litteral

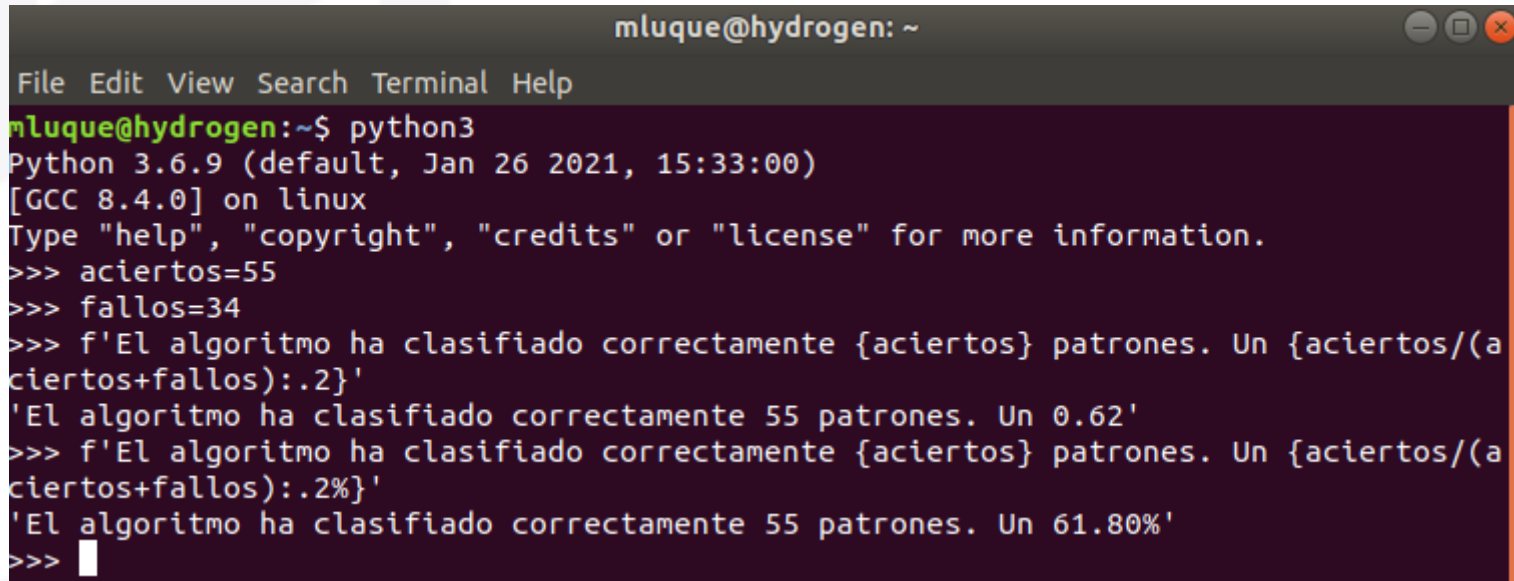
- Especificador de formato especial
  - ◆ :ancho.precision%
    - ancho → n.º de caracteres del campo
    - Precisión → n.º de decimales en reales o n.º de caracteres en cadenas.
    - % → expresa en porcentaje

# Formatted String Litteral

- Especificador de formato especial
  - ◆ :ancho.precision%
    - ancho → n.º de caracteres del campo
    - Precisión → n.º de decimales en reales o n.º de caracteres en cadenas.
    - % → expresa en porcentaje
  - ◆ otros modificadores
    - !a, !r, !s

# Formatted String Litteral

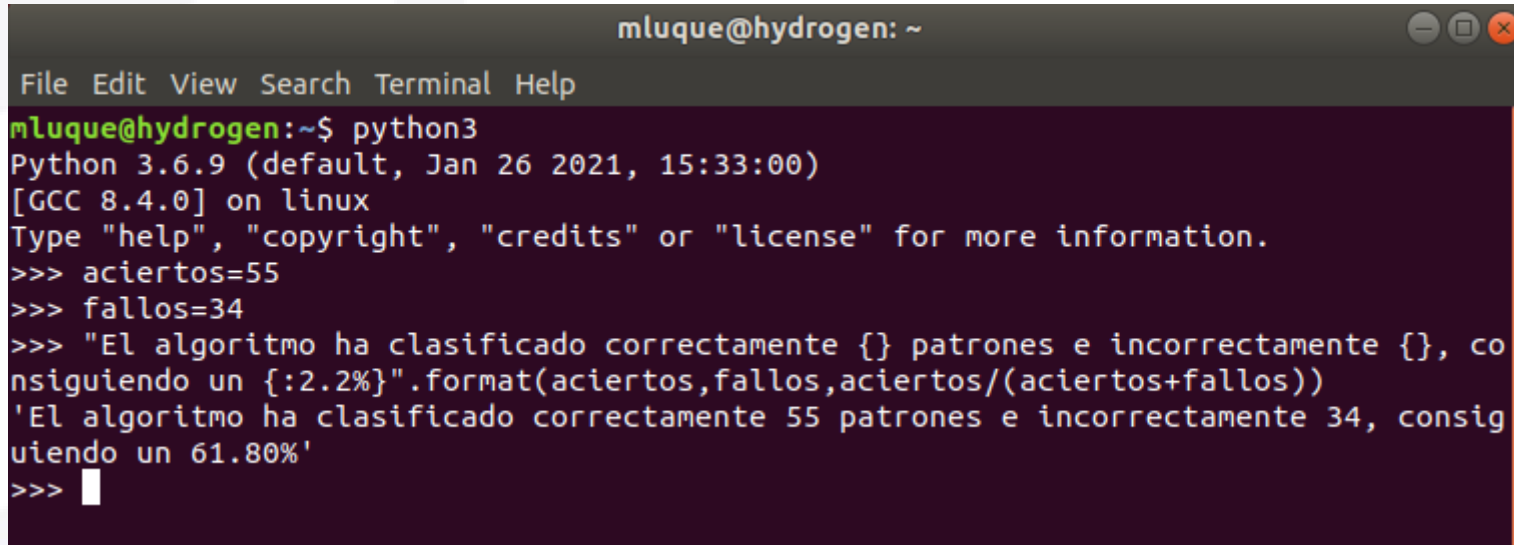
- Especificador de formato especial



```
mluque@hydrogen: ~  
File Edit View Search Terminal Help  
mluque@hydrogen:~$ python3  
Python 3.6.9 (default, Jan 26 2021, 15:33:00)  
[GCC 8.4.0] on linux  
Type "help", "copyright", "credits" or "license" for more information.  
>>> aciertos=55  
>>> fallos=34  
>>> f'El algoritmo ha clasificado correctamente {aciertos} patrones. Un {aciertos/(a  
ciertos+fallos):.2}'  
'El algoritmo ha clasificado correctamente 55 patrones. Un 0.62'  
>>> f'El algoritmo ha clasificado correctamente {aciertos} patrones. Un {aciertos/(a  
ciertos+fallos):.2%}'  
'El algoritmo ha clasificado correctamente 55 patrones. Un 61.80%'  
>>> 
```

# str.format

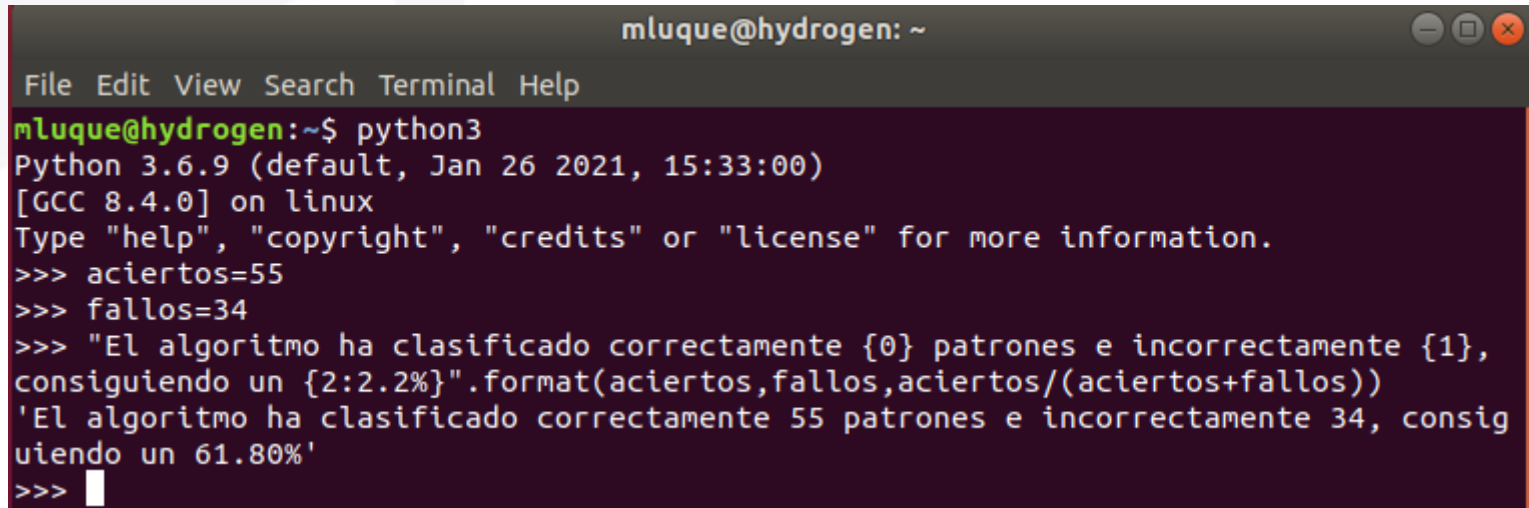
- `cadena.format(valor1, valor2 , ...)`
  - ◆ La cadena tendrá tantos `{ }` como argumentos format



```
mluque@hydrogen: ~  
File Edit View Search Terminal Help  
mluque@hydrogen:~$ python3  
Python 3.6.9 (default, Jan 26 2021, 15:33:00)  
[GCC 8.4.0] on linux  
Type "help", "copyright", "credits" or "license" for more information.  
>>> aciertos=55  
>>> fallos=34  
>>> "El algoritmo ha clasificado correctamente {} patrones e incorrectamente {}, co  
nsiguiendo un {:.2%}".format(aciertos,fallos,aciertos/(aciertos+fallos))  
'El algoritmo ha clasificado correctamente 55 patrones e incorrectamente 34, consig  
uiendo un 61.80%'  
>>> █
```

# str.format

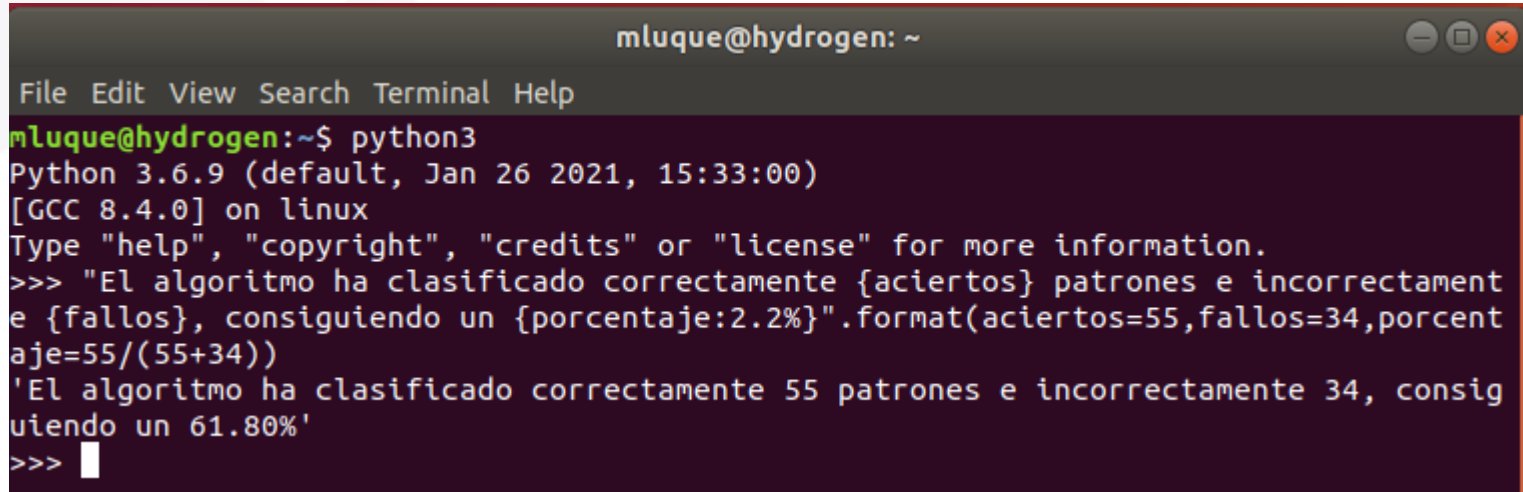
- Argumentos posicionales



```
m luque@hydrogen: ~  
File Edit View Search Terminal Help  
m luque@hydrogen:~$ python3  
Python 3.6.9 (default, Jan 26 2021, 15:33:00)  
[GCC 8.4.0] on linux  
Type "help", "copyright", "credits" or "license" for more information.  
>>> aciertos=55  
>>> fallos=34  
>>> "El algoritmo ha clasificado correctamente {0} patrones e incorrectamente {1},  
consiguiendo un {2:2.2%}".format(aciertos,fallos,aciertos/(aciertos+fallos))  
'El algoritmo ha clasificado correctamente 55 patrones e incorrectamente 34, consig  
uiendo un 61.80%'  
>>> 
```

# str.format

- Argumentos nombrados



```
m luque@hydrogen: ~  
File Edit View Search Terminal Help  
m luque@hydrogen:~$ python3  
Python 3.6.9 (default, Jan 26 2021, 15:33:00)  
[GCC 8.4.0] on linux  
Type "help", "copyright", "credits" or "license" for more information.  
>>> "El algoritmo ha clasificado correctamente {aciertos} patrones e incorrectament  
e {fallos}, consiguiendo un {porcentaje:2.2%}".format(aciertos=55,fallos=34,porcent  
aje=55/(55+34))  
'El algoritmo ha clasificado correctamente 55 patrones e incorrectamente 34, consig  
uiendo un 61.80%'  
>>> 
```



# Ajuste manual

- Métodos proporcionados por str
  - ◆ rjust → alinea el texto a la derecha
  - ◆ ljust → alinea el texto a la izquierda
  - ◆ center → centra el texto

```
>>> nombre="Maria"
>>> nombre.rjust(10)
'      Maria'
>>> nombre.ljust(10)
'Maria      '
>>> nombre.center(10)
'  Maria  '
>>> 
```

