Cadeia de Caracteres

“”” “”” / aspas triplas → usado para textos longos.

Exemplos ⤵
print(**“”””**Lorem Ipsum is simply dummy text of the printing and typesetting industry. Lorem Ipsum has been the industry's standard dummy text ever since the 1500s, when an unknown printer took a galley of type and scrambled it to make a type specimen book**“”””**)

[ ] / Listas → As **listas** são **representadas** por **[ ]**.

Fatiamento → usado para fatiar (separar) caracteres de uma variável.

Exemplo ⤵
frase = Curso em Vídeo Python {total de 21 caracteres – começando de 0}
frase[9] / *frase[9:13]* → apenas os caracteres de 9 a 13 (porém, o 13 não entra)
↳ *apenas o caractere 9*

frase[9:21:2] → selecionará caracteres de 9 até 21 (removendo o caractere 21) pulando de 2 em 2.

frase[: 5] → selecionará caracteres de 0 a 5 (removendo o caractere 5)

frase[15 :] → selecionará caracteres a partir do 15

frase[9::3] → selecionará caracteres a partir do 9, pulando de 3 em 3

Análise → utilizado para analisar uma string, obtendo algumas informações:

- Com qual letra começa
- Qual o tamanho
- Com qual letra termina
- Qual é a primeira palavra inteira

1. **len** → utilizado para medir o comprimento da frase.

Exemplo ⤵
len(frase) → vai **analisar** a variável frase e **verificar** quantos caracteres têm.

2. **count** → utilizado para contar a quantidade de uma letra.

Exemplo ⤵
frase.**count**(**‘o’**) → faz a **contagem** de quantas vezes aparece a letra **‘o’** na **variável selecionada**.

frase.**count**(**'o'**, **0**, **13**) → faz a **contagem** letras já **com** o **fatiamento**. Ou seja, **contará** quantas letras **'o'** terá **entre** o **caracter 0** e o **caracter 13** *(13 não entrará na contagem)*

3. **find** → utilizado para encontrar uma determinada letra dentro da string.

Exemplo ⤵
frase.**find**('deo') → quantas vezes ele **encontrou 'deo'** na variável. **Mostrará** a partir de qual momento apareceu 'deo', ex: 11 (a partir do caractere 11).

frase.**find**('Android') → se **dentro** da **string não** tiver a **palavra procurada**, o **programa irá retornar** como **-1**, pois as strings **começam sempre na posição 0**.

4. **in** → utilizada para saber se dentro da string contém uma determinada palavra.

Exemplo ⤵
'Curso' in frase → está perguntando se **existe a palavra** Curso na frase *(Curso em Vídeo Python)*
O programa retornará True

Transformação → Fazer transformação através de métodos.

1. **replace** → substituir uma palavra/letra por outra.

Exemplo ⤵
frase.**replace**('Python', 'Android') → A palavra Python será substituida por Android.

2. **upper()** → utilizado para deixar as letras maiúsculas.

Exemplo ⤵
frase.**upper()** → **todas** as **letras** da variante frase, ficarão **maiúsculas**.

3. lower() → usado para deixar as letras minúsculas.

Exemplo ⤵
frase**.lower()** → **todas** as **letras** da variante frase, ficarão **minúsculas**.

4. **capitalize()** → utilizado para deixar apenas a primeira letra da string maiúscula.

Exemplo ⤵
frase.**capitalize()** → Curso em vídeo python

5. **title()** → usado para deixar toda primeira letra de cada palavra em maiúsculo.

Exemplo ⤵
frase.**title()** → Curso Em Vídeo Python

6. **strip()** → usado para **remover** os **espaços inúteis** no **começo** e no **fim** da **string**.

7. **rstrip()** → usado para **remover** os **espaços inúteis apenas** da **direita (final)** da string.

8. **lstrip()** - usado para **remover** os **espaços inúteis** apenas da **esquerda (começo)** da string.

9. **split() / dividir** → usado para dividir a string (onde tiver espaço, será dividido), criando uma lista automaticamente

Exemplo ↴
frase.**split()** → Curso em Vídeo Python

10. join / juntar → usado para juntar as palavras da string, utilizando "-" como separador.

Exemplo ↴
'-'.**join(**frase**)** → Curso-em-Vídeo-Python