# Suponhamos que temos a seguinte string:

frase = 'uma FRASE'

# Podemos acessar individualmente cada caractere em uma frase.

# A ideia é a mesma de acessar uma lista:

print(frase[0])

print(frase[1])

print(frase[2])

tamanho = len(frase)

print("A frase possui", tamanho, "caracteres")

# Porém, strings são imutáveis: não podemos alterar caracteres individuais

# A linha abaixo, se for descomentada, dará erro no programa:

# frase[4] = 'C'

# Podemos converter strings para listas:

listafrase = list(frase)

print(listafrase)

# A função join() intercala cada elemento de uma lista com uma string.

stringfinal = '-py'.join(listafrase)

print(stringfinal)

# Usar um join() com uma string vazia é útil para transformar a lista de volta

# em string:

stringfinal = ''.join(listafrase)

print(stringfinal)

# Existem algumas funções interessantes que retornam a string "tratada":

s1 = frase.capitalize() # 1a letra maiúscula, restante minúscula

s2 = frase.title() # todo início de palavra em maiúscula, resto minúscula

s3 = frase.upper() # string inteira em maiúscula

s4 = frase.lower() # string inteira em minúscula

s5 = frase.replace('F', 'C') # substitui a primeira substring pela segunda

print(s1)

print(s2)

print(s3)

print(s4)

print(s5)

# Note que NENHUMA delas ALTERA a string original, elas sempre retornam

# a string nova.

print('String original:', frase)

# Outra possibilidade com strings é quebrar a string em uma lista de substrings

# Sempre que o caractere especificado é encontrado, a string é quebrada

quebra1 = frase.split(' ') # quebra a frase no caractere espaço em branco

quebra2 = s3.split('A') # quebra a frase em maiúsculas no caractere 'A'

print(quebra1)

print(quebra2)

# Podemos inserir quebras de linha com '\n'

frase = 'uma\nFRASE'

print(frase)

# Podemos inserir tabulação com '\t'

frase = 'uma\n\tFRASE'

print(frase)

# Para conseguir representar a barra '\', precisamos de 2 barras:

frase = 'uma\\FRASE'