

Guia completo com exemplos, resultados e explicações

GitHub: github.com/GustaFranz

- ✓ Strings são sequências de caracteres (textos).
- ✓ Podem usar aspas simples ' ', duplas " " ou triplas """
- ✓ Strings são **imutáveis** (não mudam diretamente; geram novas).
- ✓ Possuem diversos métodos úteis para manipulação.
- ✓ Muito usadas em validações, formatações e processamento de dados.

Código	Resultado	Explicação
'Python'	Python	String com aspas simples.
"Python"	Python	String com aspas duplas.
'''Python'''	Python	Aspas triplas (permite multilinha).
"""L1\nL2"""	L1 / L2	Strings em várias linhas.
str(123)	123	Converte um número para string.

Em todos os exemplos abaixo usamos `texto = " Python é Incrível! "` (repare nos espaços no início e no fim).

Comando	Resultado	Explicação
<code>texto.upper()</code>	PYTHON É INCRÍVEL!	Converte tudo para maiúsculas.
<code>texto.lower()</code>	python é incrível!	Converte tudo para minúsculas.
<code>texto.capitalize()</code>	Python é incrível!	Deixa só o primeiro caractere maiúsculo.
<code>texto.title()</code>	Python É Incrível!	Primeira letra de cada palavra maiúscula.
<code>texto.swapcase()</code>	pYTHON É iNCRÍVEL!	Inverte maiúsculas e minúsculas.

Remoção de espaços

Comando	Resultado	Explicação
<code>texto.strip()</code>	'Python é Incrível!'	Remove espaços do início e do fim.
<code>texto.lstrip()</code>	'Python é Incrível! '	Remove espaços apenas à esquerda.
<code>texto.rstrip()</code>	' Python é Incrível!'	Remove espaços apenas à direita.

Busca e contagem

Comando	Resultado	Explicação
<code>texto.find('Python')</code>	2	Índice da primeira ocorrência (há 2 espaços antes).
<code>texto.rfind('!')</code>	19	Índice da última ocorrência do caractere.
<code>texto.count('o')</code>	1	Quantas vezes o caractere/substring aparece.
<code>'é' in texto</code>	True	Verifica se a substring está presente.
<code>'xyz' not in texto</code>	True	Verifica se a substring NÃO está presente.

Substituição

Comando	Resultado	Explicação
<code>texto.replace('Python', 'Top')</code>	Top é Incrível!	Substitui todas as ocorrências.
<code>texto.replace('o', '0')</code>	Pyth0n é Incrível!	Substitui caractere por caractere.

Divisão e junção

Comando	Resultado	Explicação
<code>texto.split()</code>	['Python', 'é', 'Incrível!']	Divide por espaços em uma lista.
<code>texto.split('é')</code>	['Python ', ' Incrível!']	Divide usando o separador informado.
<code>'- '.join(lista)</code>	Python-é-Incrível	Junta itens de uma lista com o separador.

Verificações

Comando	Resultado	Explicação
<code>texto.isalpha()</code>	False	Há espaços e '!', então nem tudo são letras.
<code>texto.isalnum()</code>	False	Nem tudo são letras ou números.
<code>texto.isdigit()</code>	False	Não são todos dígitos.
<code>texto.isspace()</code>	False	Não são apenas espaços.
<code>texto.startswith('Py')</code>	False	Começa com espaços, não com 'Py'.
<code>texto.endswith('!')</code>	False	Termina com espaços, não com '!'.

Atenção

startswith/endswith deram False por causa dos espaços nas pontas. Aplique .strip() antes para comparar só o conteúdo: texto.strip().startswith('Py') -> True.

Exercício: analisador de frases

O usuário digita frases até escrever 'fim'. Para cada frase, removemos espaços extras, contamos palavras e vogais e verificamos se começa com maiúscula. No fim, exibimos um relatório com totais e médias.

analisador.py

```
frases = []
while True:
    entrada = input('Digite uma frase (ou "fim"): ')
    if entrada.lower() == 'fim':
        break
    frases.append(entrada.strip())

total = len(frases)
total_palavras = sum(len(f.split()) for f in frases)
media = total_palavras / total if total else 0
vogais = sum(c.lower() in 'aeiou' for f in frases for c in f)
maiusculas = sum(1 for f in frases if f[:1].isupper())
mais = max(frases, key=lambda f: len(f.split()))
menos = min(frases, key=lambda f: len(f.split()))

print('Total de frases:', total)
print('Média de palavras:', round(media, 2))
print('Frase com mais palavras:', mais)
print('Frase com menos palavras:', menos)
print('Total de vogais:', vogais)
print('Começam com maiúscula:', maiusculas)
```

Dica final + boas práticas

- ✓ Sempre remova espaços desnecessários com .strip().
- ✓ Use .lower() para comparações seguras (sem diferenciar caixa).
- ✓ Evite repetir código: aproveite os métodos prontos.
- ✓ Lembre: strings são imutáveis — os métodos geram novas strings.

Prof. Gustavo Franz (Science/Biology)*Python Developer in Progress**Building Educational Solutions with Python*

Professor de Ciencias e Biologia desde 2013 — atualmente estudando Programacao.

Eu crio estes resumos para organizar os assuntos e estudar de forma clara e objetiva. Estou compartilhando porque eles tambem podem ajudar outros desenvolvedores que estao comecando. Bons estudos — vamos aprender juntos!

GitHub: github.com/GustaFranz

Exercicios Python: github.com/GustaFranz/python_exercises

Para quem quiser ver a resolucao dos meus exercicios em Python.

EasyAnsi: github.com/GustaFranz/easyansi

Para quem quiser codificar personalizando com cores e estilos de forma mais facil, pratica e intuitiva.