

# String e fatiamento

**Guilherme Arthur de Carvalho**

Analista de sistemas

**@decarvalhogui**

# Objetivo Geral

Conhecer métodos úteis para manipular objetos do tipo string, como interpolar valores de variáveis e entender como funciona o fatiamento.

# Pré-requisitos

- Python 3
- VSCode

# Percurso

## **Etapa 1**

**Conhecendo métodos úteis da classe string**

## **Etapa 2**

Interpolação de variáveis

## **Etapa 3**

Fatiamento de string

## **Etapa 4**

String múltiplas linhas

## Etapa 1

# Conhecendo métodos úteis da classe string

# Introdução

A classe String do Python é famosa por ser rica em métodos e possuir uma interface muito fácil de trabalhar.

Em algumas linguagens manipular sequências de caracteres não é um trabalho trivial, porém, em Python esse trabalho é muito simples.

# Maiúscula, minúscula e título

```
curso = "pYtHon"  
  
print(curso.upper())  
>>> PYTHON  
  
print(curso.lower())  
>>> python  
  
print(curso.title())  
>>> Python
```

# Eliminando espaços em branco

```
curso = "  Python "  
  
print(curso.strip())  
>>> "Python"  
  
print(curso.lstrip())  
>>> "Python "  
  
print(curso.rstrip())  
>>> "  Python"
```



# Junções e centralização

```
curso = "Python"

print(curso.center(10, "#"))
>>> "##Python##"

print(".".join(curso))
>>> "P.y.t.h.o.n"
```

# Percurso

~~Etapa 1~~

~~Conhecendo métodos úteis da classe string~~

**Etapa 2**

**Interpolação de variáveis**

**Etapa 3**

**Fatiamento de string**

**Etapa 4**

**String múltiplas linhas**

## Etapa 2

# Interpolação de variáveis

# Introdução

Em Python temos 3 formas de interpolar variáveis em strings, a primeira é usando o sinal %, a segunda é utilizando o método format e a última é utilizando f strings.

A primeira forma não é atualmente recomendada e seu uso em Python 3 é raro, por esse motivo iremos focar nas 2 últimas.

# Old style %

```
nome = "Guilherme"
idade = 28
profissao = "Progamador"
linguagem = "Python"

print("Olá, me chamo %s. Eu tenho %d anos de idade, trabalho como %s e
estou matriculado no curso de %s." % (nome, idade, profissao, linguagem))

>>> Olá, me chamo Guilherme. Eu tenho 28 anos de idade, trabalho como
Progamador e utilizo e estou matriculado no curso de Python.
```

# Método format

```
nome = "Guilherme"
idade = 28
profissao = "Programador"
linguagem = "Python"

print("Olá, me chamo {}. Eu tenho {} anos de idade, trabalho como {} e
estou matriculado no curso de {}".format(nome, idade, profissao,
linguagem))

print("Olá, me chamo {3}. Eu tenho {2} anos de idade, trabalho como {1} e
estou matriculado no curso de {0}.".format(linguagem, profissao, idade,
nome))
```

# Método format

```
print("Olá, me chamo {nome}. Eu tenho {idade} anos de idade, trabalho como  
{profissao} e estou matriculado no curso de  
{linguagem}.".format(nome=nome, idade=idade, profissao=profissao,  
linguagem=linguagem))
```

```
print("Olá, me chamo {nome}. Eu tenho {idade} anos de idade, trabalho como  
{profissao} e estou matriculado no curso de  
{linguagem}.".format(**pessoa))
```

```
>>> Olá, me chamo Guilherme. Eu tenho 28 anos de idade, trabalho como  
Progamador e estou matriculado no curso de Python.
```

# f-string

```
nome = "Guilherme"
idade = 28
profissao = "Programador"
linguagem = "Python"

print(f"Olá, me chamo {nome}. Eu tenho {idade} anos de idade, trabalho
como {profissao} e estou matriculado no curso de {linguagem}.")

>>> Olá, me chamo Guilherme. Eu tenho 28 anos de idade, trabalho como
Progamador e utilizo e estou matriculado no curso de Python.
```



# Formatar strings com f-string

```
PI = 3.14159

print(f"Valor de PI: {PI:.2f}")
>>> "Valor de PI: 3.14"

print(f"Valor de PI: {PI:10.2f}")
>>> "Valor de PI:          3.14"
```

# Percurso

~~Etapa 1~~

~~Conhecendo métodos úteis da classe string~~

~~Etapa 2~~

~~Interpolação de variáveis~~

**Etapa 3**

**Fatiamento de string**

**Etapa 4**

**String múltiplas linhas**

## Etapa 3

# Fatiamento de string

# Introdução

Fatiamento de strings é uma técnica utilizada para retornar substrings (partes da string original), informando início (start), fim (stop) e passo (step): `[start: stop[, step]]`.

# Fatiamento

```
nome = "Guilherme Arthur de Carvalho"

nome[0]
>>> "G"

nome[:9]
>>> "Guilherme"

nome[10:]
>>> "Arthur de Carvalho"

nome[10:16]
>>> "Arthur"

nome[10:16:2]
>>> "Atu"

nome[:]
>>> "Guilherme Arthur de Carvalho"

nome[::-1]
>>> "oHlavraC ed ruhtrA emrehliuG"
```

# Percurso

~~Etapa 1~~

~~Conhecendo métodos úteis da classe string~~

~~Etapa 2~~

~~Interpolação de variáveis~~

~~Etapa 3~~

~~Fatiamento de string~~

**Etapa 4**

**String múltiplas linhas**

## Etapa 4

# String multiplas linhas

# Introdução

Strings de múltiplas linhas são definidas informando 3 aspas simples ou duplas durante a atribuição. Elas podem ocupar várias linhas do código, e todos os espaços em branco são incluídos na string final.



# Strings triplas

```
nome = "Guilherme"

mensagem = f"""
Olá meu nome é {nome},
Eu estou aprendendo Python
"""

>>>
```

```
Olá meu nome é Guilherme,
Eu estou aprendendo Python
```

# Strings triplas

```
nome = "Guilherme"

mensagem = f'''
    Olá meu nome é {nome},
    Eu estou aprendendo Python.
    Essa mensagem tem diferentes recuos.
'''

>>>

    Olá meu nome é Guilherme,
    Eu estou aprendendo Python.
    Essa mensagem tem diferentes recuos.
```

# Percurso

~~Etapa 1~~

~~Conhecendo métodos úteis da classe string~~

~~Etapa 2~~

~~Interpolação de variáveis~~

~~Etapa 3~~

~~Fatiamento de string~~

~~Etapa 4~~

~~String múltiplas linhas~~

Hands On!

*“Falar é fácil.  
Mostre-me o código!”*

Linus Torvalds

# Links Úteis

- <https://github.com/guicarvalho/trilha-python-dio>
- <https://docs.python.org/pt-br/3/library/string.html>
- <https://docs.python.org/pt-br/3/library/stdtypes.html#textseq>

# Dúvidas?

- > Fórum/Artigos
- > Comunidade Online (Discord)

