

# Linguagem de Programação I

**Prof. Orlando Saraiva Júnior**  
**orlando.nascimento@fatec.sp.gov.br**

# **String**

---

Dados textuais em Python são manipulados com objetos str, ou strings. Strings são sequências imutáveis de pontos de código Unicode. Literais de string são escritas de diversas maneiras:

Aspas simples: 'permite aspas duplas'

Aspas duplas: "permite aspas 'simples'"

Aspas triplas: '''Três aspas simples''', """Três aspas duplas"""

Strings entre aspas triplas podem abranger várias linhas - todos os espaços em branco associados serão incluídos na literal de string.

---

Retorna uma versão em string de object. Se object não for fornecido, retorna a string vazia. Caso contrário, o comportamento de str()

str(1)

str([1,2,3])

Uma string formatada, ou f-string, é uma string prefixada com 'f' ou 'F'.

Essas strings podem conter campos de substituição, que são expressões delimitadas por chaves {}.

Enquanto outras string sempre têm um valor constante, strings formatadas são, na verdade, expressões avaliadas em tempo de execução.

```
nome = "Ana"  
idade = 40  
mensagem = f"Olá, meu nome é {nome} e eu tenho {idade} anos."  
print(mensagem)
```

# **Métodos para String**

Existe uma poderosa função em Python chamada **len**. O valor retornado é igual ao número de caracteres na string.

```
S = 'oi mundo'
```

```
len(S)
```

Quando você precisa verificar se uma string começa ou termina com alguns caracteres, pode usar os métodos **startswith** e **endswith**.

```
S = 'Fatec Rio Claro'
```

```
S.startswith('Fatec')
```

```
S.startswith('Universidade')
```

```
S.endswith('Claro')
```

```
S.endswith('claro')
```



Uma forma de verificar se uma palavra pertence a uma string é utilizar o operador **in**. Este operador pode ser utilizado com listas.

S = 'Fatec Rio Claro'

"Fatec" in S

"Rio" in S

---

Se você precisa contar as ocorrências de letras ou palavras em uma string, use o método **count**

S = 'um tigre, dois tigres, três tigres'

S.count('tigre')

S.count('tigres')

Para pesquisar se uma string está dentro de outra e obter a posição da primeira ocorrência, use o método **find**.

S = 'Fatec Rio Claro'

S.find('Rio')

S.find('Claro')

Python apresenta métodos que ajudam a apresentar strings:

```
S = 'Fatec'
```

```
'X' + S.center(20) + 'X'
```

```
'X' + S.center(20, '.') + 'X'
```

```
S.ljust(20)
```

```
S.rjust(20)
```

# Quebra ou separação de strings

---

O método **split** quebra uma string a partir de um caracter passado como parâmetro, retornando uma lista com as substrings separadas

S = “um tigre, dois tigres, três tigres”

S.split(',')

S.split(' ')

Para isso, use o método **replace**. O primeiro parâmetro é a string a substituir. O segundo parâmetro o conteúdo que substituirá.

S = “um tigre, dois tigres, três tigres”

S.replace('tigre','gato')

S.replace('tigre','gato', 1)

S.replace('tigre','gato', 2)

S.replace('',' -')

Strings em Python podem ter seu conteúdo analisado e verificado:

```
S = '123'
```

```
S.isalnum()
```

```
S.isalpha()
```

Usando o format, podemos substituir valores usando números entre chaves

```
"{0} {1}".format("Oi","Mundo")
```

```
S = "{0} {1}!".format("Oi", "Mundo")
```

```
S = "{0}, {0}, {1}!".format("Vamos", "começar")
```

```
S = "Olá, {nome}. Você tem {idade} anos.".format(nome="Maria", idade=30)
```



# Dúvidas

**Prof. Orlando Saraiva Júnior**

[orlando.nascimento@fatec.sp.gov.br](mailto:orlando.nascimento@fatec.sp.gov.br)