

ANÁLISE E DESENVOLVIMENTO DE SISTEMAS

Computational Thinking

PROF. EDUARDO GONDO

String

As Strings são tipos de dados presentes na maioria das linguagens. Apesar de vocês já conhecerem algumas características, sempre vale a pena lembrá-las:

- ▶ representam uma informação através dos caracteres (não é um número)
- ▶ dentro do python, identificamos as Strings por `"""` (aspas duplas) ou `''` (aspas simples)
- ▶ também podemos representar como String caracteres localizados através de linhas (texto)

```
1 texto = '''If plan A does not work,  
2 the alphabet has twenty five more  
3 letters'''  
4 print(texto)
```

String

- ▶ Strings representam uma **sequência de caracteres**
- ▶ uma sequência representa uma coleção de informações organizadas de acordo com uma ordem
- ▶ em nossa disciplina, as Strings não serão usadas apenas para representar dados de entrada
- ▶ elas serão usadas dentro do algoritmo como sendo uma **coleção** de informações
- ▶ sobre essas informações criaremos alguns algoritmos

str = "HELLO"

H	E	L	L	O
0	1	2	3	4

- ▶ veja um exemplo:

String

- ▶ cada um dos caracteres é armazenado em uma posição da String
- ▶ parecido com uma palavra, onde as letras devem obedecer uma ordem relativa
- ▶ podemos acessar cada um dos caracteres através do seu índice:

```
1 st = "HELLO"  
2 print(st[0])  
3 print(st[2:4])
```

- ▶ execute o código acima, veja que na linha 2 é impresso o H
- ▶ e na linha 3, o LL

String - função len

- ▶ quando temos o código abaixo:

```
1  st = "DON'T WORRY, BE HAPPY"
2  outra_st = st[4:8]
3  tam = len(outra_st)
4  print(outra_st)
5  print(tam)
```

- ▶ estamos criando uma outra string com uma parte dos caracteres de st
- ▶ lembre-se que Strings são imutáveis
- ▶ a função `len` devolve a quantidade de caracteres na String

String - métodos

- ▶ lembre que tudo em Python são objetos e as Strings não fogem à regra
- ▶ vamos ver alguns métodos dos objetos String, lembrando que todos eles retornam **uma outra String**:
- ▶ `strip`: remove espaços em branco da String
- ▶ `lower`: converte os caracteres para minúscula
- ▶ `upper`: converte os caracteres para maiúscula
- ▶ `replace`: substitui uma substring por outra
- ▶ veja alguns exemplos desses métodos:

String - métodos exemplos

```
1 s = " Just keep swimming "  
2 t = s.strip()  
3 print(t)  
4 t = s.lower()  
5 print(t)  
6 t = s.upper()  
7 print(t)  
8 t = s.replace("swimm", "runn")  
9 print(t)
```

String - percorrendo os caracteres

para percorrer todos os caracteres de uma String podemos fazer:

```
1 s = "This is the beginning of anything you want"
2 i = 0
3 while i < len(s):
4     print(s[i])
5     i = i + 1
6
7 for c in s:
8     print(c)
```

Em Python podemos usar o comando `while` combinado com a função `len` ou usarmos o comando `for`. O comando `for` percorre a String do início até o fim dela, percorrendo caracter por caracter.

Pequeno Parênteses - for parecido com o do Java

- ▶ também podemos usar o comando `for` com o comportamento parecido com o `for` do Java
- ▶ veja o código abaixo:

```
1  for x in range(1, 100):  
2      print(x)
```

- ▶ a função `range` cria um intervalo de números
- ▶ no exemplo, é um intervalo de números de 1 a 99
- ▶ em Java, seria algo como:

```
1  for (int i = 1; i < 100; i++) {  
2      System.out.println(i);  
3  }
```

String - verificando conteúdo

- ▶ temos dois meios para verificar o conteúdo de uma String em Python
- ▶ usando o operador `in` ou o método `find` da String
- ▶ veja um exemplo de uso do `in`:

```
1 s = "Every journey begins with a single step"
2 resultado = "th" in s
3 print(resultado) #True
4 resultado = not "th" in s
5 print(resultado) #False
```

- ▶ veja se fosse o `find`

```
1 s = "Don't lose hope. When the sun goes down, the
   stars come out."
2 resultado = s.find("lose")
3 print(resultado) #6
4 resultado = s.find("he", 20)
5 print(resultado) #23
```

Exercícios

- ▶ Refaça 4 exercícios da lista de repetição usando o comando de repetição `for`
- ▶ Façam os exercícios da lista de exercícios

Referência Bibliográfica

- ▶ Puga e Riseti - Lógica de Programação e Estrutura de Dados
- ▶ Ascêncio e Campos - Fundamentos da Programação de Computadores
- ▶ Forbelone e Eberspacher - Lógica de programação: a construção de algoritmos e estruturas de dados
- ▶ Documentação do Python - <https://docs.python.org/3.8/>
- ▶ Python Programming For Beginners: Learn The Basics Of Python Programming (Python Crash Course, Programming for Dummies) (English Edition). Kindle
- ▶ Python: 3 Manuscripts in 1 book: - Python Programming For Beginners - Python Programming For Intermediates - Python Programming for Advanced (English Edition). Kindle

| Copyleft

Copyleft © 2025 Prof. Eduardo Gondo Todos direitos liberados.
Reprodução ou divulgação total ou parcial deste documento é liberada.