STRINGS

Autoria de Carolina Soares

STRINGS

Uma string é um tipo de dado que armazena uma sequência de caracteres.

```
texto = 'Olá mundo'
print(texto)
print(type(texto))
```

Saída do código:

```
Olá mundo
<class 'str'>
```

STRINGS DE DOCUMENTAÇÃO

```
texto = """

Está é uma String em Python com multiplas linhas. O Python me permite guardar uma String com várias linhas de caracteres. As Strings são textos que podem conter letras e números. Além disso, as Strings podem está dentro de aspas simples ou duplas."""
```

ÍNDICE DA STRING

```
nome = "Maria"
print(nome[0])
print(nome[1])
print(nome[2])
print(nome[3])
print(nome[4])
```

Saída do código:

```
M
a
r
i
a
```

ÍNDICE DA STRING

```
nome = "Maria"
print(nome[-1])
print(nome[-2])
print(nome[-3])
print(nome[-4])
print(nome[-5])
```

Saída do código:

```
a
i
r
a
M
```

INTERVALO DA STRING

índice	0	1	2	3	4	5	6	7	8	9
nome	J	o	ã	o		S	i	l	v	a

```
nome[ início : fim : pulo]
```

INTERVALO DA STRING

índice	-10	-9	-8	-7	-6	-5	-4	-3	-2	-1
nome	J	o	ã	o		S	i	l	v	a

```
nome[ início : fim : pulo]
```

INTERVALO DA STRING

```
nome = "João Silva"
print(nome[5:])
print(nome[-3:])
print(nome[3:8])
print(nome[-10:4])
print(nome[:7:2])
```

Saída do código:

```
Silva
lva
o Sil
João
Jã i
```

CONCATENANDO STRINGS

```
texto1 = 'Aprendendo'
texto2 = 'Python'
print(texto1 + ' ' + texto2)
```

Saída do código:

```
Aprendendo Python
```

MULTIPLICANDO A STRING

```
linha = '-'
print(linha * 20)
print('Bloco de código')
print(linha * 20)
```

Saída do código:

```
--------------------
Bloco de código
--------------------
```

OBTENDO O TAMANHO DA STRING

```
texto = "abc"
print(len(texto))
```

Saída do código:

```
3
```

STRINGS EM ESTRUTURAS DE REPETIÇÃO

```python
txt = 'Python'
for item in txt:
    print(item)
```

```python
txt = 'Python'
indice = 0
while indice < len(txt):
    print(txt[indice])
    indice += 1
```

Saída do código:

```
P
y
t
h
o
n
```

VERIFICAR SE STRING EXISTE EM OUTRA

```
texto = "Tenha um ótimo dia"
if 'dia' in texto:
    print("A sequência de caractere existe na string!")
```

Saída: A sequência de caractere existe na string!

VERIFICAR SE STRING EXISTE EM OUTRA

```
texto = "Tenha um ótimo dia"
if 'ótimo' not in texto:
    print("A palavra 'ótimo' não existe no texto")
else:
    print("Sim, 'ótimo' existe no texto!")
```

Saída: Sim, 'ótimo' existe no texto!

FUNÇÕES DA STRING

```
palavra = "Céu Limpo"

palavra.upper() # CÉU LIMPO

palavra.lower() # céu limpo

palavra.replace("Limpo", "Nublado") # Céu Nublado

palavra.split(" ") # ['Céu', 'Limpo']

palavra.startswith("Céu") # True

palavra.endswith(".") # False
```

CARACTERES DE ESCAPE NA STRING

```
print("Programação em \"Python\"")
```

Saída: Programação em "Python"

```
print("Programação \n em Python")
```

Saída: Programação

em Python

```
print("Programação \t em Python")
```

Saída: Programação	em Python

```
print("Programação e\bm\b Python")
```

Saída: Programação Python

CARACTERES DE ESCAPE

Sequência de escape	Significado
`\<newline>`	A barra invertida e a nova linha foram ignoradas
`\\`	Contrabarra (`\`)
`\'`	Aspas simples (`'`)
`\"`	Aspas duplas (`"`)
`\a`	ASCII Bell (BEL) - um sinal audível é emitido
`\b`	ASCII Backspace (BS) - apaga caractere à esquerda
`\f`	ASCII Formfeed (FF) - quebra de página
`\n`	ASCII Linefeed (LF) - quebra de linha
`\r`	ASCII Carriage Return (CR) - retorno de carro
`\t`	ASCII Horizontal Tab (TAB) - tabulação horizontal
`\v`	ASCII Vertical Tab (VT) - tabulação vertical
`\`*ooo*	Caractere com valor octal *ooo*
`\x`*hh*	Caractere com valor hexadecimal *hh*

https://docs.python.org/pt-br/3/reference/lexical_analysis.html#strings

EXERCÍCIOS

1. Utilizando estrutura de repetição imprima cada letra da palavra Python de trás para frente.

2. Um palíndromo é uma palavra que quando lida de trás para frente permanece igual. Com a lista a seguir percorra cada elemento e mostre se a palavra é palíndromo ou não: ["ana", "pelo", "ovo", "reviver", "telhado", "abacate", "radar", "osso", "reler", "sol" , "rever", "salas"]. Ao final exiba a quantidade total de palavras que são palíndromos.

3. Considere a seguinte string com nomes de frutas: "banana uva maçã melão abacaxi". Separe as palavras em uma lista com o split, verifique se uva está na lista e troque por morango. Por fim, coloque todas as palavras em maiúsculo e imprima a lista.

4. Receba uma entrada que pede o nome completo de um usuário, verifique se o nome completo do usuário possui "de" se sim utilize o replace para remover da string, conte o tamanho total do nome completo, depois pegue o intervalo do primeiro nome e o último nome do usuário e concatene em outra string. Exiba o nome e sobrenome do usuário, as suas iniciais e o tamanho do nome completo.

5. Receba uma data do usuário no formato dd/mm/aaaa e imprima a data com o mês escrito por extenso.