Data Science & Big Data

UFPR
UNIVERSIDADE FEDERAL DO PARANÁ

Strings!

Prof. André Grégio

DSBD

Strings

String é uma cadeia de caracteres

Por ex. podemos representar a string entre aspas duplas ou simples:

- “Aqui temos uma string”
- ‘Aqui temos uma string’

```
>>> print("Aqui temos uma string")
Aqui temos uma string

>>> print('Aqui temos uma string')
Aqui temos uma string
```

Strings

Podemos atribuir uma string à uma variável e manipular essa variável com uma das diversas funções disponíveis:

texto = “Aqui temos uma string”

```
>>> texto = "Aqui temos uma string"

>>> print(texto)
Aqui temos uma string

>>> print(texto.title())
Aqui temos uma string
```

Strings

Podemos acessar:

- Um caractere de cada vez pela sua posição no string.
- Um intervalo de caracteres [n:m] incluindo o inicio “n” e excluindo o último “m”

(a posição inicial em python começa com 0)

```
>>> texto = "Aqui temos uma string"
>>> texto[0]
'A'
>>> texto[3]
'I'
>>> texto[0:4]
'Aqui'
```

Strings

Podemos acessar:

- Um caractere de cada vez pela sua posição no string.
- Um intervalo de caracteres [n:m] incluindo o inicio “n” e excluindo o último “m”

(a posição inicial em python começa com 0)

```
>>> texto = "Aqui temos uma string"
>>> texto[0]
'A'
>>> texto[3]
'I'
>>> texto[0:4]
'Aqui'
```

exclui

Strings

Podemos acessar:

- Um intervalo omitindo [n:m] início ou fim
- Um intervalo com passo entre cada caracter [n:m:passo]

```
>>> texto = "Aqui temos uma string"
>>> texto[:4]
'Aqui'
>>> texto[15:]
'string'
>>> texto[0:4:2]
'Au'
>>> texto[0:4:3]
'Ai'
>>> texto[0::3]
'Aiesmsi'
```

Strings

Algumas funções úteis disponíveis:

- Minúsculo usando “lower()”
- Maiúsculo usando “upper()”
- Primeira letra maíuscula “capitalize()”

```
>>> print(texto.lower())
aqui temos uma string

>>> print(texto.upper())
AQUI TEMOS UMA STRING

>>> print(texto.lower().capitalize())
Aqui temos uma string
```

https://wiki.python.org.br/ManipulandoStringsComPython

Strings

Algumas funções úteis disponíveis:

- Retirar espaços em branco (esquerda “lstrip()”, direita “rstrip()” ou ambos “strip()”)

```
>>> texto = " Aqui temos uma string "

>>> print(texto.lstrip())
Aqui temos uma string

>>> print(texto.rstrip())
 Aqui temos uma string

>>> print(texto.strip())
Aqui temos uma string
```

Strings

Algumas funções úteis disponíveis:

- Separar a *string* por **palavras**: `"split()"`

```
>>> texto = "Aqui temos uma string"

>>> print(texto.split())

['Aqui', 'temos', 'uma', 'string']
```

Lista de palavras!

Strings

- Encontrar posição de caracter: "`find()`"
- Tamanho do string: "`len()`"
- Frequência de um caracter: "`count()`"

```
>>> texto = " Aqui temos uma string "
>>> texto.find("t")
6
>>> texto.find("m")
8
>>> texto[8]
'm'
>>> texto.find("m",10)
13
>>> len(texto)
23
>>> texto.count("m")
2
```

Strings

Concatenação de strings:

- Combinar diferentes variáveis do tipo *string*

```
>>> p_nome = "Maria"

>>> u_nome = "da Silva"

>>> nome_completo = p_nome + " " + u_nome

>>> print(nome_completo)
Maria da Silva
```

Strings

Concatenação de strings:

- Combinar variáveis com constantes

```
>>> p_nome = "Maria"

>>> u_nome = "da Silva"

>>> nome_completo = "Oi, " + p_nome + " " + u_nome + "!"

>>> print(nome_completo)
Oi, Maria da Silva!
```

Strings

Adicionando caracter não “imprimível”

- Nova linha ‘\n’
- Tabulação ‘\t’

```
>>> print("\nOlá!")

Olá!
>>> print("\tOlá!")
    Olá!
>>> print("\n\tOlá!")

    Olá!
>>> print("\t\nOlá!")

Olá!
```

Strings

Algumas funções úteis disponíveis:

- Remover **caracteres arbitrários** com [l|r]strip()

```
texto = "\tAqui temos uma string\n"


print(texto)
print(texto.strip("\n\t"))
print(texto.lstrip("\n"))
print(texto.rstrip("\t"))
print(texto.lstrip("\t"))
print(texto.rstrip("\n"))
```

???

Strings

Algumas funções úteis disponíveis:

- Separar por **caracteres arbitrários** com split()

```
>>> texto = "primeira linha do texto\nsegunda linha do texto\nterceira linha do texto\n"

>>> print(texto)

>>> print(texto.split("\n"))
```

???

Strings

Algumas funções úteis disponíveis:

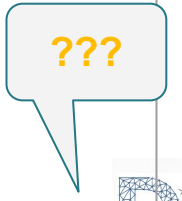
- Separar por **caracteres arbitrários** com split()

```
>>> texto = "primeira linha do texto\nsegunda linha do texto\nterceira linha do
texto\n"

>>> print(texto)

primeira linha do texto
segunda linha do texto
terceira linha do texto

>>> print(texto.split("\n"))
['primeira linha do texto', 'segunda linha do texto', 'terceira linha do texto', '']
```



Formatação de strings

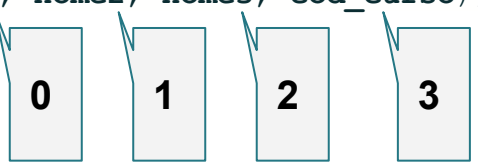
```
nome1 = "André"

nome2 = "Eduardo"

nome3 = "Tamy"

cod_curso = 240
```

```
print("{}, {} e {} estão responsáveis pela disciplina CI{}...".format(nome1, nome2, nome3, cod_curso))
```




```
print("{2}, {1} e {0} estão responsáveis pela disciplina CI{3}!".format(nome1, nome2, nome3, cod_curso))

print("Prof. {sobrenome} foi ao médico hoje.".format(sobrenome = "Almeida"))
```

Formatação de strings

```
msg1 = "Linguagem"
msg2 = "Progr"
msg3 = "Python 3"
'A {:<10} de {:^10} é {:>10}'.format(msg1, msg2, msg3)
```

Mais informações em:

https://pyformat.info/

L	i	n	g	u	a	g	e	m	
		P	r	o	g	r			
		P	y	t	h	o	n		3

Strings

Erros comuns na impressão de *strings*:

- Aspas simples e duplas no texto

```
>>> texto = "Willie's really weary"
>>> print(texto)
Willie's really weary

>>> texto = 'Willie's really weary'
  File "<stdin>", line 1
    texto = 'Willie's really weary'
                    ^
SyntaxError: invalid syntax

>>> texto = '"Confusão com aspas", Curso de Python'
>>> print(texto)
"Confusão com aspas", Curso de Python
```

Strings

Erros comuns na impressão de strings:

- Operando String com tipos diferentes, função str()

```
>>> print(1+1)
2

>>> print(1+'1')
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
TypeError: unsupported operand type(s) for +: 'int' and 'str'


>>> print(str(1)+'1')
11
```

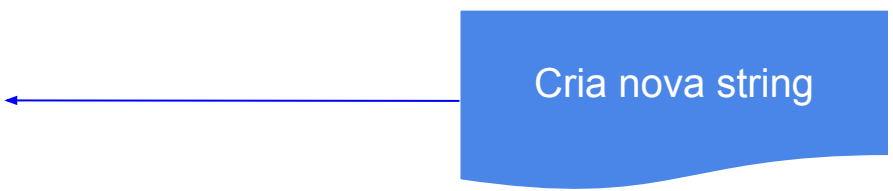
Indica concatenação de strings

Strings

Strings são imutáveis e não conseguimos alterar o valor

```
>>> texto = "HAqui temos uma string"
>>> texto[0]= ''
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
TypeError: 'str' object does not support item assignment


>>> novo = texto[1:]
>>> print(novo)
Aqui temos uma string
```



Exercícios

1. Defina a variável “nome” e atribua a ela o nome de uma pessoa. Depois imprima o nome da pessoa no lugar de ‘fulano’ na mensagem:
   a. “Oi fulano, vamos aprender Python?”
   b. Imprima a mensagem em maiúsculo.
2. Imprima a citação a seguir em uma linha e seu autor na linha seguinte, deslocado três *tabs* à direita: **“No Brasil, quando o feriado é religioso, até ateu comemora”, Jô Soares**
3. Imprima a mensagem **“O resultado da fórmula de Bhaskara é:”** e o resultado
   a. Em seguida da mensagem
   b. Abaixo da mensagem
   c. Como uma *string* formatada com a, b, c, e as raízes, cada um em uma linha
4. Faça os exercícios 1, 2 e 3 com impressão “simples” e usando *format*.