



Universidade Federal do Rio Grande
Centro de Ciências Computacionais



Algoritmos e Estruturas de Dados I

Listas e *Strings*

Profs. Drs. Cleo Billa, Rafael Penna e Thiago da Silveira

1º Semestre de 2020

Roteiro

- Conjuntos de valores
- Listas
- *Strings*
- Função `len` e *casting* para lista
- Iterando sobre listas e *strings*
- Listas de listas e listas de *strings*

Conjuntos de valores

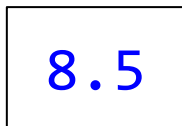
- E se eu quisesse armazenar os nomes e médias dos alunos?

```
numNotas = 2
numAlunos = 10
for aluno in range(numAlunos):
    nome = input('Digite seu nome: ')
    soma = 0
    for nota in range(numNotas):
        valor = float(input('Digite uma nota: '))
        soma = soma + valor
    print(nome, 'teve média', soma/numNotas)
```

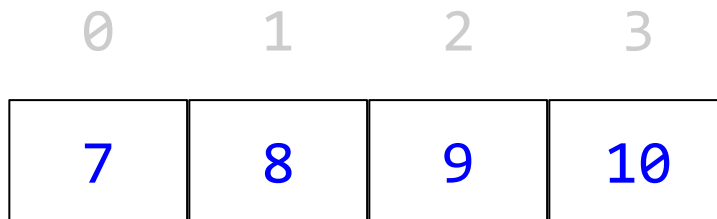
Listas

- Contrapõem variáveis escalares;
- São estruturas de dados que permitem o armazenamento de conjuntos de informações quaisquer.

m



n



Listas

- Contrapõem variáveis escalares;
- São estruturas de dados que permitem o armazenamento de conjuntos de informações quaisquer.

m


8.5

n

0	1	2	3
'a'	1	True	3.14

Operações com listas

- Instanciação

`var` = `[`  `]`
Valores

```
lista1 = []  
lista2 = [7, 8, 9, 10]  
lista3 = ['a', 1, True, 3.14]
```

Operações com listas

- Acesso

var[índice]

[0, tamanho da lista)

```
lista1 = []
lista2 = [7, 8, 9, 10]
lista3 = ['a', 1, True, 3.14]

print(lista1[0])
print(lista2[0])
print(lista2[1])
print(lista2[2])
print(lista2[3])
print(lista2[4])
print(lista3[0], type(lista3[0]))
print(lista3[1], type(lista3[1]))
print(lista3[2], type(lista3[2]))
print(lista3[3], type(lista3[3]))
print(lista3[4], type(lista3[4]))
```

Operações com listas

- Acesso

var[índice]

[0, tamanho da lista)

```
lista1 = []
lista2 = [7, 8, 9, 10]
lista3 = ['a', 1, True, 3.14]

print(lista1[0])
print(lista2[0])
print(lista2[1])
print(lista2[2])
print(lista2[3])
print(lista2[4])
print(lista3[0], type(lista3[0]))
print(lista3[1], type(lista3[1]))
print(lista3[2], type(lista3[2]))
print(lista3[3], type(lista3[3]))
print(lista3[4], type(lista3[4]))
```


Operações com listas

- Atribuição

var **[** índice **]** = ...
 └──┬──┘
 [0, tamanho da lista)
 Valor

```
lista1 = []  
lista2 = [7, 8, 9, 10]  
lista3 = ['a', 1, True, 3.14]
```

```
print(lista1)  
print(lista2)  
print(lista3)
```

```
lista1[0] = 1  
lista2[1] = 8.5  
lista3[0] = 0  
lista3[2] = 1
```

```
print(lista1)  
print(lista2)  
print(lista3)
```

Operações com listas

- Atribuição

var **[** índice **]** = ...



[0, tamanho da lista)

Valor

```
lista1 = []  
lista2 = [7, 8, 9, 10]  
lista3 = ['a', 1, True, 3.14]
```


```
print(lista1)  
print(lista2)  
print(lista3)
```

```
lista1[0] = 1  
lista2[1] = 8.5  
lista3[0] = 0  
lista3[2] = 1
```

```
print(lista1)  
print(lista2)  
print(lista3)
```

Operações com listas

- Concatenação

`var += [`  `]`

Valores

```
lista1 = []  
lista2 = [7, 8, 9, 10]  
lista3 = ['a', 1, True, 3.14]  
  
lista1 += [1]  
lista2 += [11]  
lista3 += [False]  
  
print(lista1)  
print(lista2)  
print(lista3)
```

Veja mais sobre *lists* em: <https://docs.python.org/3/tutorial/datastructures.html>

Strings

- São estruturas de dados que permitem o armazenamento de conjuntos de informações do tipo caractere.

m 0

'c'

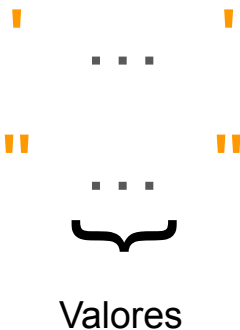
n 0 1 2 3

'c'	'a'	's'	'a'
-----	-----	-----	-----

Operações com *strings*

- Instanciação

`var = ' ... '`
`var = " ... "`



Valores

```
string1 = ""  
string2 = 'a'  
string3 = 'abacaxi'
```

Operações com *strings*

- Acesso

var[índice]

[0, tamanho da *string*)

```
string1 = ""  
string2 = 'a'  
string3 = 'abacaxi'  
  
print(string1[0])  
print(string2[0], string2[1])  
print(string3[0])  
print(string3[1])  
print(string3[2])
```

Operações com *strings*

- Acesso

var[índice]



[0, tamanho da *string*)

```
string1 = ""  
string2 = 'a'  
string3 = 'abacaxi'  
  
print(string1[0])  
print(string2[0], string2[1])  
print(string3[0])  
print(string3[1])  
print(string3[2])
```

Operações com *strings*

- Atribuição

```
string1 = ""  
string2 = 'a'  
string3 = 'abacaxi'
```

```
string1[0] = "a"  
string2[0] = "b"  
string3[5] = "t"  
string3[6] = "e"
```

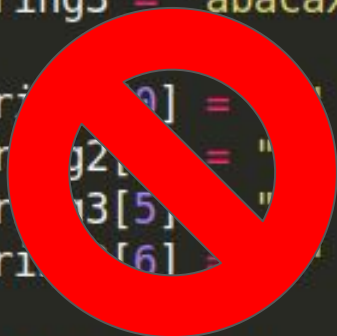
```
print(string1)  
print(string2)  
print(string3)
```

Veja mais sobre *strings* em: <https://docs.python.org/3/library/string.html>

Operações com *strings*

- Atribuição


```
string1 = ""  
string2 = 'a'  
string3 = 'abacaxi'  
  
string1[0] = 'a'  
string2[0] = 'A'  
string3[5] = 'X'  
string3[6] = 'Y'  
  
print(string1)  
print(string2)  
print(string3)
```



Veja mais sobre *strings* em: <https://docs.python.org/3/library/string.html>

Operações com *strings*

- Concatenação

`var += ' ... '`
`var += " ... "`

Valores

```
string1 = ""  
string2 = 'a'  
string3 = 'abacaxi'  
  
string1 += 'a'  
string2 += "bacaxi"  
string3 += ''  
  
print(string1)  
print(string2)  
print(string3)
```

Veja mais sobre *strings* em: <https://docs.python.org/3/library/string.html>

Função len e *casting* para lista

- len(*lista ou string*)

```
lista1 = []  
lista2 = [7, 8, 9, 10]  
lista3 = ['a', 1, True, 3.14]  
  
string1 = ""  
string2 = 'a'  
string3 = 'abacaxi'  
  
print(len(lista1), len(lista2), len(lista3))  
print(len(string1), len(string2), len(string3))
```

Função len e *casting* para lista

- `list(string)`
- `list(range(...))`

```
string1 = ""  
string2 = 'a'  
string3 = 'abacaxi'  
  
lista4 = list(string1)  
lista5 = list(string2)  
lista6 = list(string3)  
lista7 = list(range(11))  
  
print(lista4)  
print(lista5)  
print(lista6)  
print(lista7)
```

Iterando sobre listas e *strings*

- Uso de laços de repetição para percorrer os elementos de uma lista ou *string*;

```
lista3 = ['a', 1, True, 3.14]
string3 = 'abacaxi'

for indice in range(len(lista3)):
    print(lista3[indice])

for elemento in string3:
    print(elemento)
```

Listas de listas e listas de *strings*

- Podemos ter listas de listas;
- Podemos ter listas de *strings*;
- Podemos ter listas de listas de listas/*strings*;
- Podemos ter listas de listas de ... de listas/*strings*.

```
lista2 = [7,8,9,10]
lista3 = ['a', 1, True, 3.14]

string2 = 'a'
string3 = 'abacaxi'

lista8 = [[]]
lista9 = [lista2, lista3]
lista10 = [string2, string3]
lista11 = [string2, list(string3)]

print(lista8)
print(lista9)
print(lista10)
print(lista11)
```

Conjuntos de valores... de novo

- E se eu quisesse armazenar os nomes e médias dos alunos?

```
numNotas = 2
numAlunos = 10
for aluno in range(numAlunos):
    nome = input('Digite seu nome: ')
    soma = 0
    for nota in range(numNotas):
        valor = float(input('Digite uma nota: '))
        soma = soma + valor
    print(nome, 'teve média', soma/numNotas)
```

Conjuntos de valores... de novo

- E se eu quisesse armazenar os nomes e médias dos alunos?

```
nomes = []
medias = []

numNotas = 2
numAlunos = 3
for aluno in range(numAlunos):
    nome = input('Digite seu nome: ')
    nomes += [nome]
    soma = 0
    for nota in range(numNotas):
        valor = float(input('Digite uma nota: '))
        soma = soma + valor
    media = soma/numNotas
    medias += [media]

for indice in range(len(medias)): # ou len(nomes)
    print(nomes[indice], 'teve média', medias[indice])
```




Universidade Federal do Rio Grande
Centro de Ciências Computacionais



Algoritmos e Estruturas de Dados I

Listas e *Strings*

Profs. Drs. Cleo Billa, Rafael Penna e Thiago da Silveira

1º Semestre de 2020