Programação Estruturada de Computadores

M.Sc Nairon Viana

nairon.viana@ifpi.edu.br

Programação Estruturada de Computadores

Roteiro – Próximos conteúdos de aula

- Tipos de Dados em Python
  - Tipos Compostos mutáveis e imutáveis
    - Imutáveis – tuplas = ()
    - Mutáveis – listas = [] e dicionários = {}
- Manipulando Strings em Python
- Manipulando Tuplas
- Manipulando Listas e Dicionários
- Introdução às Estruturas de Repetição em Python

Programação em Python

Linguagem Python

– **Variáveis e Tipos da Linguagem**

Variáveis Compostas – Tuplas, Listas e Dicionários

Programação em Python

Linguagem Python

– **Definições**

Tipos e Variáveis Compostas em Python

Variáveis compostas: também chamadas sequências são variáveis que armazenam um conjunto de dados de tipos simples (primitivos, como float, str, int, bool, etc)

Características

1 única variável representa mais de 1 valor

Valores são representados sequência, separados por vírgula

Cada valor pode ser identificado por um índice

Índice: um inteiro ou um str, float, etc (no caso de dicionários)

Programação em Python

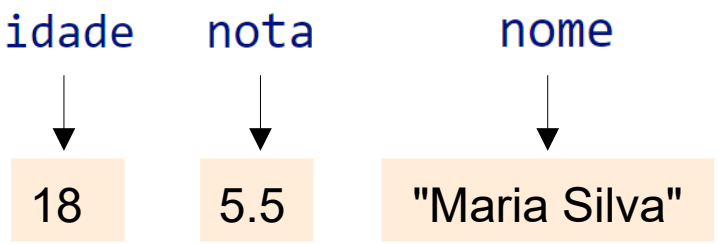
Linguagem Python

– **Definições**

Tipos e Variáveis Compostas em Python

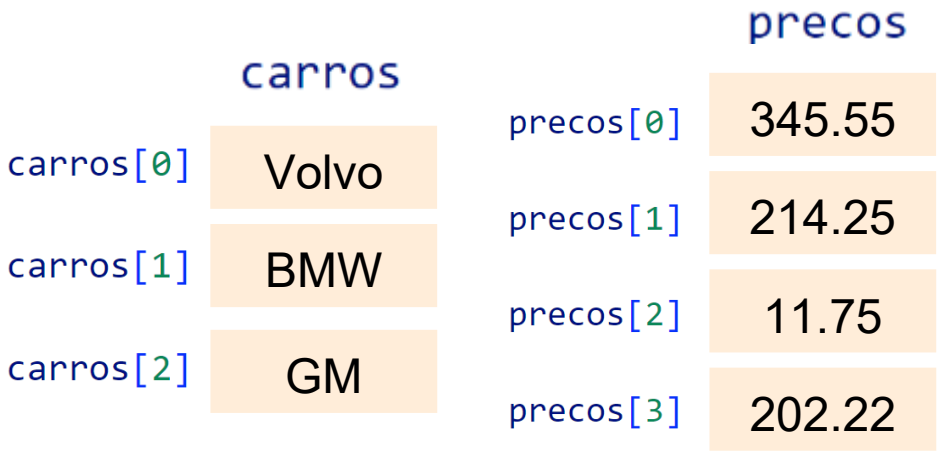
Variável Simples

```
idade = 18
nota = 5.5
nome = "Maria Silva"
```



Variável Composta

```
carros = ["Volvo", "BMW", "GM"]
precos = [345.55, 214.25, 11.75, 202.22]
```

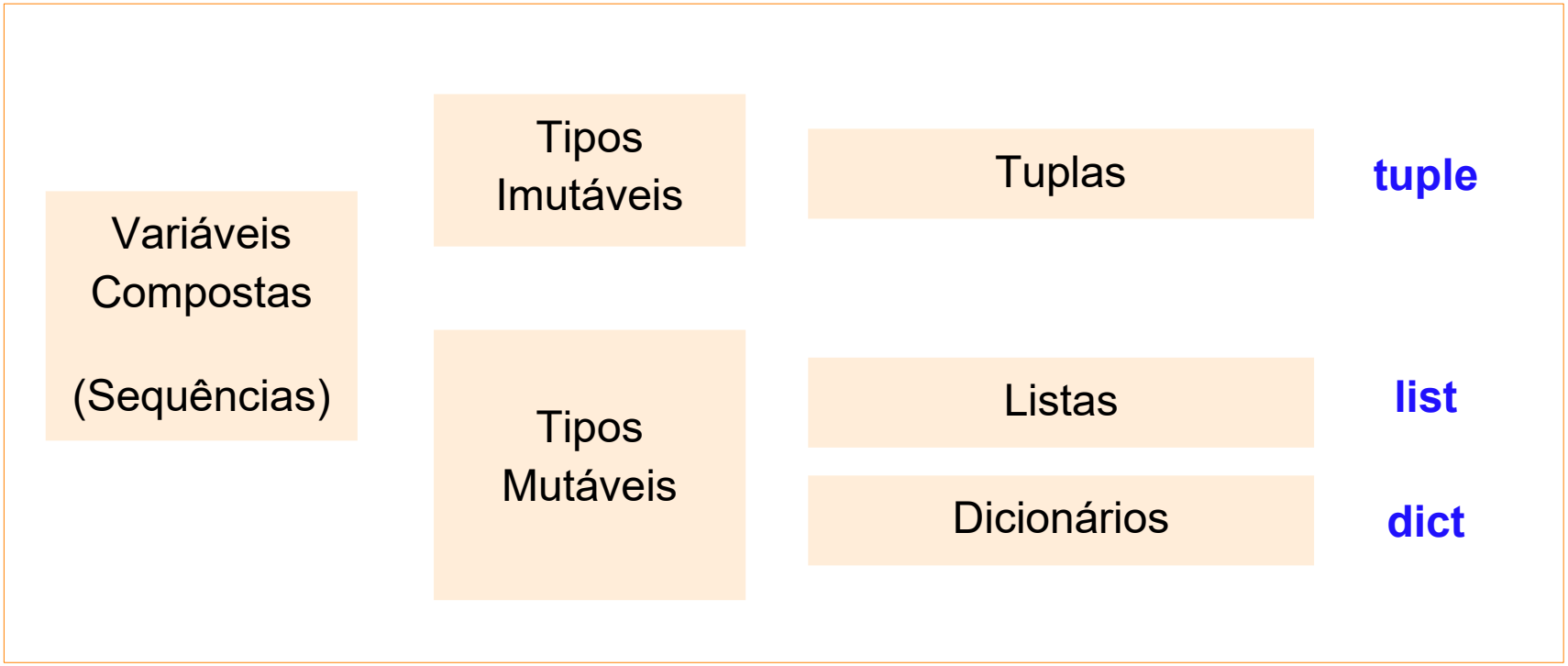


Programação em Python

Linguagem Python

– **Definições**

Tipos e Variáveis Compostas em Python



Programação em Python

Linguagem Python

– **Definições**

Tipo Tupla

Tipo imutável (1x definido, seus elementos não podem ser modificados)

Classe **tuple**

A sequência de itens é representada entre **parênteses**

tupla = ()

Cada item pode ser acessado por meio de um índice inteiro

A sequência do índice é **crescente** começando com 0 e aumentando da esquerda para a direita

O tamanho da tupla (quantidade de itens) é verificado usando a função **len**

Programação em Python

Linguagem Python

– **Definições**

Tipo Tupla – Operações

```
# Criando a variável tupla (4 elementos)
disc =('MATE', 'HIST', 'FISI', 'QUIM')

# Mostrando o tipo da variável
print(type(disc))

# Acessando o primeiro elemento
print(disc[0])

# Acessando o último elemento da tupla
# O índice também pode ser uma variável
ultimo = 3
print(disc[ultimo])

# Mostrando o tamanho da tupla
print(len(disc))
```

Resultado

```
<class 'tuple'>
MATE
QUIM
4
```

Programação em Python

Linguagem Python

– **Definições**

Tipo Tupla

Observações – Possíveis erros no uso de **tuplas**

```
# Não tentar alterar um elemento (erro!)
disc[0] = "BIOL"
```

```
   TypeError: 'tuple' object does not support item assignment
```

```
# Não acessar um elemento com índice inexistente
print('Posição 5:', disc[4])
```

```
      print('Posição 5:', disc[4])
                          ~~~~^^^
   IndexError: tuple index out of range
```

Programação em Python

Linguagem Python

– **Definições**

Tipo Lista

Tipo mutável (1x definido, seus elementos **podem** ser modificados)

Classe **list**

A sequência de itens é representada entre **colchetes**

lista = []

Cada item pode ser acessado por meio de um índice inteiro

A sequência do índice é **crescente** começando com 0 e aumentando da esquerda para a direita

O tamanho da tupla (quantidade de itens) é verificado usando a função **len**

Programação em Python

Linguagem Python

– **Definições**

Tipo Lista – Operações

```
# Criando a variável lista (3 elementos)
livros =["Algoritmos", "Redes", "Web Design"]

# Mostrando o tipo da variável
print(type(livros))

# Mostrando toda a lista
print(livros)

# Acessando os elementos
print(livros[0]) # Algoritmos
print(livros[1]) # Redes
print(livros[2]) # Web Design
```

Resultado

```
<class 'list'>
Algoritmos
Redes
Web Design
['Algoritmos', 'Redes', 'Web Design']
3
```

```
# Mostrando o tamanho da lista
print(len(livros))
```

Programação em Python

Linguagem Python

– **Definições**

Tipo Lista – Operações

```
['Algoritmos', 'Redes', 'Web Design']
```

1
```
# Acessando um elemento qualquer da lista
# O índice também pode ser uma variável
indice = 2
print(livros[indice])
```

2
```
# Alterando os elementos 1 e 3
livros[0] = "Prog. Mobile"
livros[2] = "Lógica de Prog."
print(livros)
```

3
```
# Removendo elemento
livros.remove("Redes")
print(livros)
```

4
```
# Adicionando elemento ao final
livros.append("Redes II")
print(livros)
```

Resultado

```
1 Web Design
2 ['Prog. Mobile', 'Redes', 'Lógica de Prog.']
3 ['Prog. Mobile', 'Lógica de Prog.']
4 ['Prog. Mobile', 'Lógica de Prog.', 'Redes II']
```

Programação em Python

Linguagem Python

– **Definições**

Tipo Lista – Operações mais importantes

```
livros =["Algoritmos", "Redes", "Web Design"]   elemento = "Segurança"
```

Operação	Comando
Adicionar um elemento ao final da lista	livros.**append**(elemento)
Estender a lista com uma segunda lista	livros.**extend**(novalista)
Contar a quantidade de vezes que um elemento se repete na lista	livros.**count**("Web Design")
Posição de um elemento na lista	livros.**index**("Redes")
Inserir um elemento em uma posição **pos** posição da lista	livros.**insert**(**pos**, "Redes II")

Programação em Python

Linguagem Python

– **Definições**

Tipo Lista – Operações mais importantes

```
livros =["Algoritmos", "Redes", "Web Design"]   elemento = "Segurança"
```

Operação	Comando
Remover e elemento da posição **pos** da lista	livros.**pop**([**pos**])
Remover um elemento de uma posição **pos** da lista (outra forma – comando **del**	**del** livros[**pos**]
Remover um elemento qualquer da lista	livros.**remove**(**elemento**)
Inverter a lista	livros.**reverse**()
Ordenar a lista (quando a lista é numérica)	livros.**sort**()
Soma dos elementos (lista numérica)	**sum**(**livros**)

Programação em Python

Tipo Lista – Operações

```
1 notas = [7.5, 2.5, 5.5]
2 notas[0] = notas[0] + notas[1]
3 notas[len(notas)-1] = 9.8
4 notas.append(2.5)
5 notas2 = [8.0, 1.1]
6 notas.extend(notas2)
7 print('Quant. 2.5?', notas.count(2.5))
8 print('Pos. 9.8: ', notas.index(9.8))
9 notas.insert(5, 0.5)
10 notas.pop(0)
11 del notas[0]
12 notas.remove(8.0)
13 notas.reverse()
14 notas.sort()
15 print('Soma: ', sum(notas))
```

Resultado

```
[7.5, 2.5, 5.5]
[10.0, 2.5, 5.5]
[10.0, 2.5, 9.8]
[10.0, 2.5, 9.8, 2.5]
[10.0, 2.5, 9.8, 2.5, 8.0, 1.1]
Quant. 2.5? 2
Pos. 9.8:  2
[10.0, 2.5, 9.8, 2.5, 8.0, 0.5, 1.1]
[2.5, 9.8, 2.5, 8.0, 0.5, 1.1]
[9.8, 2.5, 8.0, 0.5, 1.1]
[9.8, 2.5, 0.5, 1.1]
[1.1, 0.5, 2.5, 9.8]
[0.5, 1.1, 2.5, 9.8]
Soma:  13.9
```

Programação em Python

Linguagem Python

– **Definições**

Observação – Tipo Lista e Tupla

Tanto Listas como Tuplas, os valores dos itens podem ser definidos não necessariamente do mesmo tipo, podemos ter uma tupla/lista com itens de **TIPOS DISTINTOS**

```
# Exemplo de Tupla com tipos diferentes
aluno    = ('Maria Silva', 7.5, 18, 'APROVADO')
produto1 = ('Arroz', 2.55, 'Alimentos', 'OK', False)
produto2 = ('Sabonete', 1.55, 'Higiene', 'OK', True)

# Exemplo de Lista com tipos diferentes
funcionario = ['Pedro José', 'MASC', True, 'Teresina', 2590.55]
disc = ['Matematica', 'Exatas', False, 250, 255.65]
```

Programação em Python

Linguagem Python

– **Definições**

Listas com Entrada/Saída

Como Listas são tipos **MUTÁVEIS** podemos inicializar seus elementos com valores recebidos por meio da entrada padrão **input**

Primeira Forma

```
# Lista com 3 elementos str vazios
alunos = ['', '', '']

alunos[0] = str(input('Nome do primeiro aluno: '))
alunos[1] = str(input('Nome do segundo aluno: '))
alunos[2] = str(input('Nome do terceiro aluno: '))

print('Primeiro: ', alunos[0])
print('Segundo: ', alunos[1])
print('Terceiro: ', alunos[2])
```

Programação em Python

Linguagem Python

– **Definições**

Listas com Entrada/Saída

Como Listas são tipos **MUTÁVEIS** podemos inicializar seus elementos com valores recebidos por meio da entrada padrão **input**

Segunda Forma

```
# Lista vazia
alunos = []

al1 = str(input('Nome do primeiro aluno: '))
al2 = str(input('Nome do segundo aluno: '))
al3 = str(input('Nome do terceiro aluno: '))

# Inserindo cada elemento no início
alunos.insert(0, al1)
alunos.insert(0, al2)
alunos.insert(0, al3)

print('Primeiro: ', alunos[0])
print('Segundo: ', alunos[1])
print('Terceiro: ', alunos[2])
```

Programação em Python

Linguagem Python

– **Definições**

Listas com Entrada/Saída

Como Listas são tipos **MUTÁVEIS** podemos inicializar seus elementos com valores recebidos por meio da entrada padrão **input**

Terceira Forma

```
# Lista vazia
alunos = []

# Inserindo cada elemento no início
alunos.insert(0, str(input('Nome do primeiro aluno: ')))
alunos.insert(0, str(input('Nome do segundo aluno: ')))
alunos.insert(0, str(input('Nome do terceiro aluno: ')))

print(alunos)
```

Programação em Python

Linguagem Python

– **Definições**

Tipo Dicionário

Tipo mutável (1x definido, seus elementos **podem** ser modificados)

Classe **dict**

A sequência de itens é representada entre **chaves**

dicionario = {}

A associação com índices é diferente, uma vez que o índice pode ser qualquer valor, não somente inteiro

Dicionários são elementos em pares do tipo **[chave: valor]**, onde a **chave** é usada como referência para acessar um valor do conjunto

Em **Python** são representados pelo par **[key: value]**

Programação em Python

Linguagem Python

– **Definições**

Tipo Dicionário – Operações

```
# Criando a variável lista (3 elementos)
alunos = {"Sérgio":8.5, "Márcia":4.5, "Mauro":6.5 }

# Mostrando o tipo da variável
print(type(alunos))

# Mostrando as chaves (keys)
print(alunos.keys())

# Mostrando os valores (value)
print(alunos.values())

# Tamanho do dicionário
print(len(alunos))
```

Resultado

```
<class 'dict'>
dict_keys(['Sérgio', 'Márcia', 'Mauro'])
dict_values([8.5, 4.5, 6.5])
3
```

Programação em Python

Linguagem Python

Tipo Dicionário – Operações

```
# Mostrando a variável
print(alunos)

# Acessando os elementos
print(alunos["Sérgio"])
print(alunos["Márcia"])
print(alunos["Mauro"])

# Adicionando um valor
# Sempre ao final
alunos["Pedro"] = 2.5

# Alterando um valor
alunos["Sérgio"] = 1.5

# Removendo um valor/chave
del alunos["Mauro"]
```

Resultado

```
{'Sérgio': 8.5, 'Márcia': 4.5, 'Mauro': 6.5}
8.5
4.5
6.5
{'Sérgio': 8.5, 'Márcia': 4.5, 'Mauro': 6.5, 'Pedro': 2.5}
{'Sérgio': 1.5, 'Márcia': 4.5, 'Mauro': 6.5, 'Pedro': 2.5}
{'Sérgio': 1.5, 'Márcia': 4.5, 'Pedro': 2.5}
```

Programação em Python

Linguagem Python

– **Definições**

Tipo Tupla

Observações – Possíveis erros no uso de **tuplas**

```
# Não tentar alterar um elemento (erro!)
disc[0] = "BIOL"
```

```
   TypeError: 'tuple' object does not support item assignment
```

```
# Não acessar um elemento com índice inexistente
print('Posição 5:', disc[4])
```

```
      print('Posição 5:', disc[4])
                          ~~~~^^^
   IndexError: tuple index out of range
```

Programação em Python

Linguagem Python

– **Definições**

Dict com Entrada/Saída

Dicionários são tipos **MUTÁVEIS** logo podemos inicializar seus elementos com valores recebidos da entrada padrão **input**

```
# Criando a variável lista (3 elementos)
alunos = {}

al1 = str(input('Nome 1o Aluno: '))
alunos[al1] = float(input('Nota 1o Aluno: '))

al2 = str(input('Nome 2o Aluno: '))
alunos[al2] = float(input('Nota 2o Aluno: '))

al3 = str(input('Nome 3o Aluno: '))
alunos[al3] = float(input('Nota 3o Aluno: '))

print('Nomes (Chaves)', alunos.keys())
print('Notas (Valores)', alunos.values())
```

Programação em Python

Linguagem Python

– **Variáveis e Tipos da Linguagem**

Próximo: Exemplos em Python e Listas de Exercícios Práticos em Laboratório

Programação Estruturada de Computadores

M.Sc Nairon Viana

nairon.viana@ifpi.edu.br