Algoritmos e Estrutura de Dados

Bruno Feres de Souza
bferes@gmail.com

Universidade Federal do Maranhão
Bacharelado em Ciência e Tecnologia

1° semestre de 2016

Na aula anterior...

Dados e Tipos de Dados

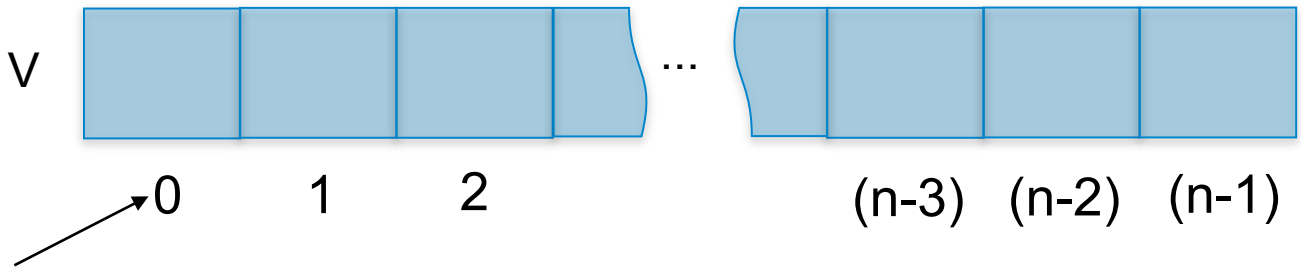
- Em Python:
  - Tipos de dados **atômicos**:
    - int e float: +, -, *, /, %, **
    - bool: and, or, not
  - Tipos de dados de **coleção**:
    - Listas
    - Strings
    - Tuplas e **Dicionários**

Tuplas
Definição

- Uma **tupla** é uma coleção ordenada de zero ou mais valores, de um mesmo tipo ou não. Ela serve para agrupar dados.

- Cada valor pode ser acessado por um índice dentro da tupla.

- **Semelhante a Listas**, exceto que Tuplas são **imutáveis**: uma vez definidas, elas permanecem iguais por toda execuçaõ do programa.



Tuplas

Criando tuplas

- Para criar uma tupla, utilizam-se parênteses ()

- Tupla vazia:

```
T1 = ()
```

- Tupla homogênea:

```
T2 = (1,2,3,4,5)
```

- Tupla heterogênea:

```
T3 = ('fisica', 'quimica', 1997, 2000)
```

- Tupla de um elemento:

```
T4 = (50,)
```

- Tupla a partir de uma lista

```
T5 = tuple([3,2,7,4,1])
```

Tuplas

Acessando tuplas

- T[i] retorna o iésimo item da tupla T.

```
T = (3,2,7,4,1)
item = T[4]
```

- T[inicio:fim] retorna os elementos do início ao fim de T. Isto chama-se **fatiamento** de tuplas.

```
T = (3,2,7,4,1)
seq = T[1:4]
seq2 = T[:3]
seq3 = T[:]
seq4 = T[:-1]
seq5 = T[::2] #T[inicio:fim:n]
```

Tuplas

Manipulando tuplas

- Não é possível modificar elemento de uma tupla.

```
T = (3,2,7,4,1)
T[4]=8 #Erro!!!
```

- É possível criar tuplas a partir de tuplas.

```
T1 = (3,2,7,4,1)
T2 = ('a','b','c')
T3 = T1 + T2 #concatenação de tuplas!
print(T3)
```

Tuplas

Manipulando tuplas

- Não é possível remover um elemento de uma tupla.

```
T = (3,2,7,4,1)
del T[2] #Erro!!!
```

- É possível criar uma nova tupla sem um determinado item.

```
T1 = (3,2,7,4,1)
T2 = T1[:2] + T1[3:]
print(T2)
```

Tuplas

Operações básicas sobre tuplas

- Tamanho de uma tupla

```
T = (3,2,7,4,1)
print(len(T))
```

- Repetição de tuplas

```
T = ('Oi',)*4
print(T)
```

- Pertencer a uma tupla

```
T = (3,2,7,4,1)
print(3 in T)
```

Tuplas

Operações básicas sobre tuplas

- Máximo em uma tupla

```
T = (3,2,7,4,1)
print(max(T))
```

- Mínimo em uma tupla

```
T = (3,2,7,4,1)
print(min(T))
```

- Comparar tuplas: compara elemento por elemento

```
T1 = (3,2,7,4,1)
T2 = (5,6,8)
cmp(T1,T2) #Retorna -1 se T1 for menor, #0 se T1 e T2 forem iguais e #1 se T2 for menor
```

Tuplas

Operações básicas sobre tuplas

- Iteração em tuplas

```
T = (3,2,7,4,1)
for x in T:
  print x
```

- **Observação**: compreensão de **listas** funciona apenas em **listas**!

```
T1 = (x**2 for x in range(10)) #Não produz o efeito esperado!

T2 = tuple([x**2 for x in range(10)]) #Produz o efeito esperado
```

Tuplas

Empacotamento/desempacotamento de tuplas

- Empacotar significa agrupar valores.
- É utilizado para permitir múltiplas atribuições de valores simultaneamente.
- Empacotamento

```
T = ('Carlos', 'BCT', '2015.2')
```

- Desempacotamento:

```
(nome, curso, ingresso) = T
```

Tuplas

Empacotamento/desempacotamento de tuplas

- Como trocar o valor de duas variáveis a e b?

- E como fazer a troca utlizando empacotamento?

Tuplas

Empacotamento/desempacotamento de tuplas

- Como trocar o valor de duas variáveis a e b?

```
a = 10
b = 5
temp = a
a = b
b = temp
```

- E como fazer a troca utlizando empacotamento?

```
a = 10
b = 5
(a,b) = (b,a)
```

Tuplas
Retorno de funções

- Funções podem retornar até um valor apenas

```
def f(r):
    c = 2 * math.pi * r
    return c
```

- E se quisermos retornar mais de um valor?

Tuplas

Retorno de funções

- Funções podem retornar até um valor apenas

```
def f(r):
    c = 2 * math.pi * r
    return c
```

- E se quisermos retornar mais de um valor?

```
def f(r):
    c = 2 * math.pi * r
    a = math.pi * r * r
    return (c,a)
```

Material complementar

http://www3.ifrn.edu.br/~jurandy/fdp/doc/aprenda-python/capitulo_09.html#capitulo-9-tuplas

http://www.tutorialspoint.com/python/python_tuples.htm

http://interactivepython.org/runestone/static/pythonds/Introduction/GettingStartedwithData.html

Dicionários

Definição

- Um **dicionário** é uma coleção de zero ou mais pares de itens, onde cada par consiste de uma chave e um valor.

- Diferentemente das demais coleções estudadas (listas, tuplas e strings), os índices dos dicionários podem ser também quaisquer tipos imutáveis de dados.

Produto	Preço
alface	R$0.45
batata	R$1.20
tomate	R$2.30
feijão	R$1.50

```
tabela={"alface":0.45,
        "batata":1.20,
        "tomate":2.30,
        "feijao":1.50}
```

chave *valor*

Dicionários

Criando dicionários

- Para criar um dicionários, utilizam-se chaves **{ }**

- Dicionário vazio:
  - D1 = {}
- Dicionário com elementos homogêneos:
  - D2 = {'MA': 'Sao Luis', 'Para':'Belem'}
- Dicionários com elementos heterogêneos:
  - D3 = {'um':1,'dois':2,'tres':3}
  - D4 = {1:'oi',2:'ola',3:'tchau'}
  - D5 = {(1,2):'tupla!',True:'bool!','t':'string!'}
  - D6 = {[3,4]:'lista!'} #Erro!

Observação: **valor** pode ser de qualquer tipo; **chave** pode ser números, booleanos, Strings, tuplas.

Dicionários

Criando dicionários

- Para criar um dicionários, utilizam-se chaves **{ }**

- Dicionários por adição de elementos

```
D = {}
D['um'] = 'one'
D['dois'] = 'two'
D['tres'] = 'three'
print(D)
```

- **Observação**: chaves devem ser únicas

```
D = {'Nome': 'Ana', 'Idade': 27, 'Nome': 'Maria'}
print(D)
```

Dicionários

Criando dicionários

- Referenciando o mesmo dicionário

  D = {'MA': 'Sao Luis', 'Para':'Belem'}

  D2 = D

  Observação: D e D2 referem-se a mesma posição de memória!

- Copiando um dicionário

  D = {'MA': 'Sao Luis', 'Para':'Belem'}

  D2 = D.copy()

  Observação: D e D2 possuem os mesmos elementos mas não se referem a mesma posição de memória!

Dicionários

Acessando dicionários

- D[**chave**] retorna o **valor** correspondent a **chave**.

```
Registro = {'Nome':'Ana', 'CPF':'1234567', 'Idade':27}
print(Registro['Nome'])
print(Registro['CPF'])
print(Registro['Idade'])
```

- Se a chave não estiver presente, um erro é gerado.

```
D = {'Codigo':'1234567','Prova':7.5}
print(D['Prova'])
print(D['Trabalho']) #Erro!
```

Dicionários

Manipulando dicionários

- Modifica-se o valor de um elemeno por sua chave

```
D = {'Codigo':'1234567','Prova':7.5}
D['Prova'] = 9.0 #Atualiza elemento existente!
D['Prova'] += 1.0
D['Trabalho'] = 5.5 #Inclui a nova chave 'Trabalho'!
```

- Removendo dicionários

```
D = {'Brasil':'Real','EUA':'Dolar', 'Japao':'Iene'}
del D['EUA'] #Remove um elemento específico
D.clear() #Remove todas os elementos
del D #Remove o dicionário
```

Dicionários

Operações básicas sobre dicionários

- Tamanho de um dicionário

```
D = {'Brasil':'Real','EUA':'Dolar', 'Japao':'Iene'}
print(len(D))
```

- Chaves de um dicionário

```
D = {'Brasil':'Real','EUA':'Dolar', 'Japao':'Iene'}
print(D.keys())
```

- Valores em um dicionário

```
D = {'Brasil':'Real','EUA':'Dolar', 'Japao':'Iene'}
print(D.values())
```

Dicionários

Operações básicas sobre dicionários

- Elementos em um dicionários

```
D = {'Brasil':'Real','EUA':'Dolar', 'Japao':'Iene'}
print(D.items())
```

- Verifica se a chave pertence a um dicionário

```
D = {'Brasil':'Real','EUA':'Dolar', 'Japao':'Iene'}
print('EUA' in D)
print(D.has_key('EUA')) #Soh funciona no Python 2.7
```

- Adiciona um dicionário a outro

```
D = {'Brasil':'Real','EUA':'Dolar', 'Japao':'Iene'}
D2 = {'Argentina':'Peso','Reino Unido':'Libra'}
D.update(D2)
print(D)
```

Dicionários

Iterando dicionários

- Utilizando o for-in: iterar pelos valores

```
D = {'Brasil':'Real','EUA':'Dolar', 'Japao':'Iene'}
for m in D.values():
  print('A moeda eh: ' + m)
```

- Utilizando o for-in: iterar pelas chaves

```
D = {'Brasil':'Real','EUA':'Dolar', 'Japao':'Iene'}
for p in D.keys():
  print('O pais eh: ' + p)
```

- Utilizando o for-in: iterar pelos elementos

```
D = {'Brasil':'Real','EUA':'Dolar', 'Japao':'Iene'}
for (p,m) in D.items():
  print('O pais ' + p + ' tem a moeda ' + m)
```

Material complementar

http://www3.ifrn.edu.br/~jurandy/fdp/doc/aprenda-python/capitulo_10.html#id3

http://www.tutorialspoint.com/python/python_dictionary.htm

http://interactivepython.org/runestone/static/pythonds/Introduction/GettingStartedwithData.html

http://openbookproject.net/thinkcs/python/english3e/dictionaries.html

Dúvidas?