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
    - Tuplas
    - **Strings**
    - Dicionários

Strings
Definição

- Uma **string** é uma coleção ordenada de zero ou mais letras, números e outros símbolos.

- Cada caractere componente da string pode ser acessado por um índice dentro da string.

- Como tuplas, strings são **imutáveis**: uma vez definidas, elas permanecem iguais por toda execuçaõ do programa.

Strings
Criando strings

- Para criar uma string, utilizam-se aspas (simples ou duplas)

- String vazia:
  - `S1 = ''`
- String não vazia:
  - `S2 = 'estrutura de dados'`
- Strings a partir de valores
  - `S3 = str(1)`
  - `S4 = str('Ola')`
  - `S5 = str([1,2,3])#Observação: cria uma #string com a representação #textual da lista.`

Strings

Acessando strings

- `S[i]` retorna o iésimo item da string `S`.

```
S = 'algoritmos'
caractere = S[4]
```

- `S[inicio:fim]` retorna os elementos do início ao fim de `S`. Isto chama-se **fatiamento** de strings.

```
S = 'algoritmos'
seq = S[1:4]
seq2 = S[:3]
seq3 = S[:]
seq4 = S[:-1]
seq5 = S[::2] #S[inicio:fim:n]
```

Strings

Manipulando strings

- Não é possível modificar um caractere de uma string.

```
S = 'algoritmos'
S[4]='R' #Erro!!!
```

- É possível criar strings a partir de strings.

```
S1 = 'estrutura'
S2 = 'dados'
S3 = S1 + S2 #concatenação de strings!
print(S3)
S4 = 'oi! '*4 # repetição de strings!
```

Strings

Manipulando strings

- `replace`: substitui strings para criar uma **nova** string.

```
S1 = 'um aluno, dois alunos, tres alunos.'
S2 = S1.replace('aluno','estudante')
print(S1)
print(S2)
```

Strings

Manipulando strings

- Não é possível remover um elemento de uma tupla.

```
S = 'algoritmos'
del S[2] #Erro!!!
```

- É possível criar uma **nova** string sem um determinado caractere.

```
S1 = 'algoritmos'
S2 = S1[:2] + S1[3:]
print(S2)
```

Strings

Manipulando strings

- Iteração em strings

```
S = 'algoritmos'
for x in S:
  print(x)
```

- `split`: particiona uma string de acordo com algum critério

```
S = 'algoritmos e estrutura de dados'
L = S.split(' ')
print(L)
```

Strings

Encontrando elementos em strings

- `in`: verifica se uma string está em outra

```
print('a' in 'algorimos')
print('ri' in 'algoritmos')
```

- `not in`: verifica se uma string não está em outra

```
print('h' not in 'algorimos')
print('dados' not in 'algoritmos')
```

- `find`: encontra o índice da primeira ocorrência de uma string em outra

```
S = 'algoritmos'
print(S.find('o'))
print(S.find('o',5))
print(S.find('tm'))
print(S.find('dados'))
```

Strings

Formatando strings

- Faça um programa que peça ao usuário que digite o nome e a idade de uma pessoa e imprima:

```
Fulano tem tantos anos.
```

- Possível solução

```
nome = input('Digite um nome: ')
idade = input('Digite uma idade: ')
print(nome + ' tem ' + str(idade) + '
 anos.')
```

Strings

Formatando strings

- Faça um programa que peça ao usuário que digite o nome e a idade de uma pessoa e imprima:

```
Fulano tem tantos anos.
```

- Melhor solução: operador `%`

```
nome = input('Digite um nome: ')
idade = input('Digite uma idade: ')
print('%s tem %d anos.' %(nome,idade))
```

Strings

Formatando strings

- Outros exemplos do uso do operador %

```
print('Seus nomes são %s %s' %('Pedro','Lara'))
print('Hoje é dia %d' %(5))
print('Hoje é dia %02d' %(5))
print('Pi vale %f' %(math.pi))
print('Pi vale %0.2f' %(math.pi))
```

- Caracteres especiais

```
print('Oi, tudo bem?')
print('Oi,\n tudo bem?')
print('Oi,\t tudo bem?')
```

Strings

Operações básicas sobre strings

- Tamanho de uma string

```
S = 'algoritmos'
print(len(S))
```

- Contagem em uma string

```
S = 'algoritmos'
print(S.count('o'))
print(S.count('h'))
```

- Coloca a primeira letra em maiúscula

```
S1 = 'algoritmos'
S2 = S1.capitalize()
print(S2)
```

Strings

Operações básicas sobre strings

- Verifica se todos os caracteres são alfanuméricos

```
S1 = 'Oi, tudo bem?'
S2 = 'Eutenho10anos'
print(S1.isalnum())
print(S2.isalnum())
```

- Verifica se todos os caracteres são letras

```
S1 = 'Eu estudo na Ufma'
S2 = 'EuestudonaUfma'
print(S1.isalpha())
print(S2.isalpha())
```

Strings

Operações básicas sobre strings

- Verifica se todos os caracteres são números

```
S1 = '10-12-2016'
S2 = '10122016'
print(S1.isdigit())
print(S2.isdigit())
```

- Retorna a string em maiúscula e em minúscula

```
S1 = 'Algoritmos'
S2 = S1.upper()
S3 = S1.lower()
print(S1)
print(S2)
print(S3)
```

Material complementar

http://www3.ifrn.edu.br/~jurandy/fdp/doc/aprenda-python/capitulo_07.html#id1

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

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

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

Dúvidas?