

Introdução à Python

Altino Dantas

PARTE I

Objetivos



- Aprender a manipular arquivos em Python
 - Conhecer os diferentes modos de abertura de arquivos
 - Como percorrer um arquivo lendo de ou escrevendo em;

Manipulando arquivos



- Para trabalharmos com arquivos, teremos que criar um objeto (que nos disponibilizará métodos como read ou write).
- O acesso ou leitura ao arquivo dependerá do modo em que o objeto foi criado.
- Existem duas categorias de objetos de arquivo: arquivos binários e arquivos de texto. Vamos trabalhar com os arquivos de texto.

Manipulando arquivos

- A sintaxe será:

```
open(nome_do_arquivo, 'modo')
```

“**open()**” retornará um objeto de arquivo, e é sempre utilizado seguindo o padrão: nome do arquivo e o modo (leitura ou escrita).



Modos disponíveis

Os modos disponíveis para trabalhar com arquivos são:

Modo	Significado
'r'	Abre o arquivo para leitura
'w'	Abre o arquivo para escrita (sobrescreve)
'x'	Abre para criação (falha caso o arquivo exista)
'a'	Abre para escrita (acrescenta no arquivo)
'+'	Abre um arquivo para atualização (leitura e escrita)

Podemos combinar alguns modos, como por exemplo: r+ (irá abrir o arquivo e permitir a leitura e escrita).



Escrevendo em arquivos

```
with open('arquivo', 'w') as f:  
    f.write('Curso de Python')
```

Desta forma os dados serão sobrescritos, ou seja, este modo é indicado para gravar os dados inicialmente.



Escrevendo

Podemos escrever em um determinado ponto do arquivo do arquivo:

```
with open('arquivo', 'w') as f:  
    f.seek(0)  
    f.write('Curso de Python')
```

O “**seek**” é quem definirá a posição de meu ponteiro, onde o 0 (zero) indica a primeira linha.



Lendo arquivos



```
with open('arquivo', 'r+') as f:  
  
    print f.read()
```

O read permite que você leia o arquivo todo. Temos também o **readlines** que trará todo o conteúdo em forma de lista.



Exercício



- Faça um programa que leia um arquivo texto contendo uma lista de endereços IP e gere um outro arquivo, contendo um relatório dos endereços IP válidos e inválidos.

Entrada

```
200.135.80.9
192.168.1.1
8.35.67.74
257.32.4.5
85.345.1.2
1.2.3.4
9.8.234.5
192.168.0.256
```

Saída

```
[Endereços válidos:]
200.135.80.9
192.168.1.1
8.35.67.74
1.2.3.4

Endereços inválidos:]
257.32.4.5
85.345.1.2
9.8.234.5
192.168.0.256
```

Possibilidade de checar a validade de um endereço IP

```
import socket

addr = '257.32.4.5'

try:
    socket.inet_aton(addr) # Legal
except socket.error:
    print("IP incorreto")
```

IP incorreto

Obrigado

altinobasilio@inf.ufg.br

Dúvidas ou sugestões?

