

# ARQUIVOS

---

Wesley Lima

# Arquivos

- Até agora nossos dados desapareciam ao sair do IDLE
- Arquivos servem para armazenamento permanente
- Um arquivo é uma área em disco onde podemos ler ou gravar informações
- Acessamos o arquivo pelo seu nome
- Para acessar um arquivo é preciso abri-lo

# Arquivos

- Ao abrir o arquivo informamos seu nome, diretório onde fica (se necessário) e que operações iremos executar: leitura e/ou escrita
- A função que abre os arquivo é open e os modos são: r – leitura, w – escrita, a – append, b – binário, + (atualização)
- Os métodos para ler ou escrever são read e write.
- Os arquivos devem ser fechados com close.

# Arquivos

```
arquivo = open('números.txt', 'w')
for linha in range(1, 101):
    arquivo.write('%d\n' % linha)
arquivo.close()
```

- Caso você execute este programa nada aparecerá na tela
- Procure no diretório onde salvou o programa, o arquivo números.txt
- O modo w cria o arquivo se ele não existir, caso exista ele será apagado e reescrito

# Arquivos

```
arquivo = open('números.txt', 'r')
for linha in arquivo.readlines():
    print(linha)
arquivo.close()
```

- readlines gera uma lista onde cada elemento é uma linha lida
- Arquivos textos são simples e possuem um caracter de controle no final para pular linha
- Se quisermos tirar esse caracter do final podemos usar `print(linha.rstrip())`

# Modo Python

```
with open('números.txt') as f:  
    print(f.read())
```

- O código acima faz o mesmo da forma pythônica
- No slide anterior vimos como programadores normais fazem a leitura
- Python é legal, pois sempre você pode se aprofundar mais
- Python é simples, mas difícil de esgotar

# Exemplo Crypto

- Leia um arquivo mensagem.txt e grave cripto.txt com todas as vogais trocadas por '\*'

```
texto = open('mensagem.txt')
saida = open('cripto.txt', 'w')
for linha in texto.readlines():
    for letra in linha:
        if letra in 'aeiou':
            saida.write('*')
        else:
            saida.write(letra)
texto.close()
saida.close()
```

# Validar endereços IP

- Crie um programa que leia um arquivo texto contendo uma série de endereços Ips e faça sua validação. A regra de validação é que cada octeto de um endereço deve ser um valor de no máximo 255. O programa deverá criar dois novos arquivos. Valido.txt que conterà os IPs válidos e Invalidos.txt que conterà aqueles endereços que não seguem o padrão.

## **IPS.txt**

200.135.80.9  
192.168.1.1  
8.35.67.74  
257.32.4.5  
85.345.1.2  
1.2.3.4  
9.8.284.5  
192.168.0.256

## **Válidos.txt**

200.135.80.9  
192.168.1.1  
8.35.67.74  
1.2.3.4

## **Inválidos.txt**

257.32.4.5  
85.345.1.2  
9.8.284.5  
192.168.0.256



# Validar endereço IP

```
def ip_ok(ip):  
    ip = ip.split('.')  
    for byte in ip:  
        if int(byte) > 255:  
            return False  
    return True  
arq = open('IPS.txt')  
validos = open('Válidos.txt', 'w')  
invalidos = open('Inválidos.txt', 'w')  
for linha in arq.readlines():  
    if ip_ok(linha):  
        validos.write(linha)  
    else:  
        invalidos.write(linha)  
arq.close()  
validos.close()  
invalidos.close()
```

# Arquivos e diretórios

- `import os`
- `os.getcwd()` – Retorna o diretório atual.
- `os.mkdir()` – Cria um novo diretório.
- `os.makedirs("p1/p2/p3")` – Cria todos os diretórios no caminho especificado.
- `os.chdir()` – muda o diretório atual.
- `os.chdir(..)` – muda para o diretório pai.
- `os.rename("velho", "novo")` – renomeia um diretório ou arquivo. Também pode mover um arquivo.
- `os.rmdir()` – apaga um diretório.
- `os.remove()` – apaga um arquivo.
- `os.listdir()` – Lista os arquivos e diretórios.

# Arquivos e diretórios

- `import os.path`
- `os.path.isdir()` – verifica se é um diretório.
- `os.path.isfile()` – verifica se é um arquivo.
- `os.path.exists()` – verifica se um arquivo ou diretório já existe.
- Informações dos arquivos
  - `os.path.getsize()` – Tamanho do arquivo
  - `os.path.getctime()` – Data de criação do arquivo.
  - `os.path.getmtime()` – Data de modificação do arquivo.
  - `os.path.getatime()` – Data de acesso ao arquivo.

# Uso de caminhos

- `import os`
- `caminho="dir1/dir2/dir3"`
- `os.path.abspath(caminho)`
- `os.path.basename(caminho)`
- `os.path.dirname(caminho)`
- `os.path.splitext("teste.txt")`
- `os.path.splitdrive("c:\windows")`
- `os.path.join("c:", "windows", "system32")`

- `os.walk()`: Percorre todos os subdiretórios.

```
import os
```

```
import sys
```

```
for raiz, diretorios, arquivos in
```

```
os.walk(sys.argv[1]):
```

```
    print(raiz)
```

```
    For d in diretorios:
```

```
        Print("%s\n",d)
```

```
    For f in arquivos:
```

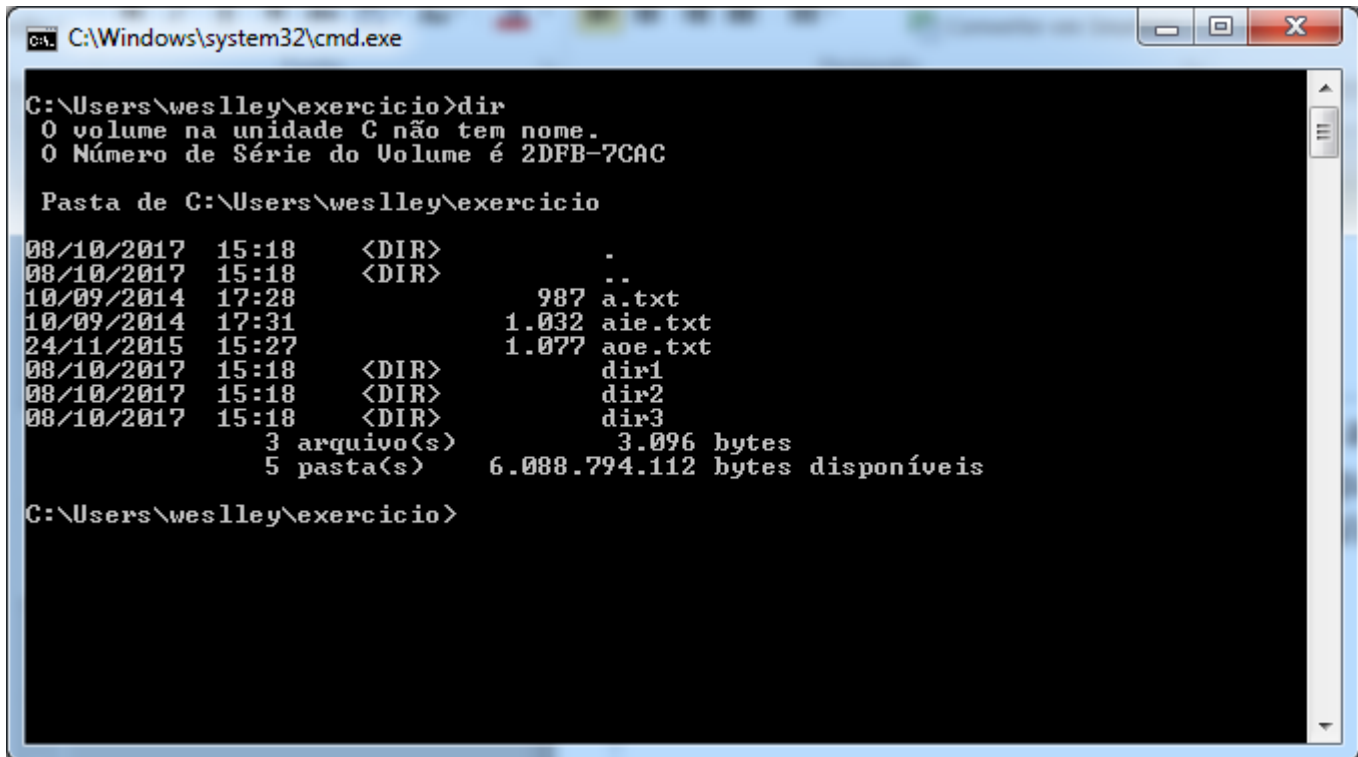
```
        print("%s\n",f)
```

# A biblioteca de tempo

- `import time`
- `agora=time.time()`
- `time.ctime(agora)`
- `tm=time.localtime()`
- `tm`
- `time.strftime("%A ,%H:%M ",tm)` – Personaliza a data/hora

# Exercício

- Desenvolva o um programa que faça a listagem de arquivos e diretórios de um diretório qualquer. A saída deve ser igual a do comando dir do Windows conforme ilustrado na tela abaixo

A screenshot of a Windows command prompt window. The title bar shows the path 'C:\Windows\system32\cmd.exe'. The command prompt shows the current directory as 'C:\Users\weslley\exercicio' and the command 'dir' has been executed. The output displays a directory listing with columns for date, time, type, and name. It shows three files: 'a.txt' (987 bytes), 'aie.txt' (1,032 bytes), and 'aoe.txt' (1,077 bytes). It also shows three subdirectories: 'dir1', 'dir2', and 'dir3'. At the bottom, it summarizes the contents: '3 arquivo(s) 3.096 bytes' and '5 pasta(s) 6.088.794.112 bytes disponíveis'.

```
C:\Windows\system32\cmd.exe

C:\Users\weslley\exercicio>dir
0 volume na unidade C não tem nome.
0 Número de Série do Volume é 2DFB-7CAC

Pasta de C:\Users\weslley\exercicio

08/10/2017  15:18    <DIR>          .
08/10/2017  15:18    <DIR>          ..
10/09/2014  17:28             987 a.txt
10/09/2014  17:31          1.032 aie.txt
24/11/2015  15:27          1.077 aoe.txt
08/10/2017  15:18    <DIR>          dir1
08/10/2017  15:18    <DIR>          dir2
08/10/2017  15:18    <DIR>          dir3
               3 arquivo(s)          3.096 bytes
               5 pasta(s)       6.088.794.112 bytes disponíveis

C:\Users\weslley\exercicio>
```