

Aula 06

- *Operating System Module (OS Module)*
 - Manipulação de arquivos e diretórios
 - Compactação e descompactação de arquivos
- Platform
 - Coleta de dados da *plataforma*.
- Programação em Rede
 - URLLib
- Atividade Prática

- O **módulo OS**, fornece um conjunto de funções para interação com o sistema operacional.
- As funcionalidades, na grande maioria, podem ser utilizadas independentemente do sistema operacional.

- ***os.name***
 - Retorna o nome do sistema operacional.
- ***os.getcwd()***
 - Retorna o diretório corrente *Current Working Directory (CWD)*
- ***os.error***
 - Todas as funções deste módulo podem produzir exceções do tipo *OSError*, as quais são capturadas com o bloco de tratamento.

```
1  import os
2  os.system("cls")
3
4  print(os.name)
5  print(os.getcwd())
6
7  #tratamento de exceções
8  try:
9      f = open('xyzw.txt','r')
10     txt = f.read()
11     f.close()
12 except OSError as e:
13     print("Erro ao abrir o arquivo")
14
```

...

2: Python Debu ▾



nt

G:\src

Erro ao abrir o arquivo

Operating System (OS)

- ***os.path.join***
 - Especifica um caminho (*path*) multiplataforma utilizando o separador apropriado.
- ***os.sep***
 - Indica o separador utilizado pelo SO.

```
18     dir = os.path.join(os.sep, 'g:\\', 'src', 'data')
19     print(os.listdir(dir))
```

```
... 2: Python Debu ▼ + [ ] [ ] [ ] [ ] [ ] [ ]
['countries.txt', 'exemplos_arquivos.zip', 'livraria.csv',
output.csv', 'output.json', 'output.xml', 'sample.csv', 'sa
e.json', 'sample.xml']
```

- ***os.mkdir***
 - Cria um diretório no diretório atual
- ***os.makedirs***
 - Cria um diretório a partir de um caminho (path)
- ***os.chdir***
 - Altera do diretório atual

- ***Criação de diretórios***

```
21  dir = os.path.join(os.sep, 'c:\\', 'temp')
22  if (not os.path.exists(dir)):
23      |    os.makedirs(dir)
24
25  os.chdir(dir)
26  for i in range(1,6):
27      |    os.mkdir(str(i))
```


- ***os.rename***
 - Alterar o nome de um diretório ou arquivo

```
30 for i in os.listdir(dir):  
31     atual = os.path.join(os.sep,os.getcwd(), i)  
32     novo  = os.path.join(os.sep,os.getcwd(), "dir_%s" %i)  
33     os.rename(atual,novo)
```

- ***os.rmdir***
 - Apaga um diretório

```
36     for i in os.listdir(dir):  
37         |     os.rmdir(str(i))
```

- ***os.environ.keys()***
 - Retorna as variáveis de ambiente

```
print(os.environ.keys())  
print(os.environ["HOMEPATH"])
```

- *Percorrer arquivos e diretórios*

```
65     dir = os.path.join(os.sep, "g:\\", "src")
66     for root,dirs,files in os.walk(dir):
67         for name in files:
68             print(os.path.join(root, name))
69         for name in dirs:
70             print(os.path.join(root, name))
```

Compactando arquivos e diretórios

```
76 import os
77 import zipfile as zip
78 path_zip = os.path.join(os.sep, "g:\\", "output.zip")
79 path_dir = os.path.join(os.sep, "g:\\", "src")
80
81 zf = zip.ZipFile(path_zip, "w")
82 for dirname, subdirs, files in os.walk(path_dir):
83     zf.write(dirname)
84     for filename in files:
85         zf.write(os.path.join(dirname, filename))
86 zf.close()
```

Descompactando arquivos e diretórios

```
88     #diretório para extração do arquivo
89     dir = os.path.join(os.sep, "g:\\", "teste")
90     if not os.path.isdir(dir):
91         os.makedirs(dir)
92
93     #extrair o arquivo
94     zf = zip.ZipFile("g:\\output.zip", "r")
95     zf.extractall(dir)
96     zf.close()
97
```

- Utilizado para extrair informações específicas a respeito da plataforma.
- O módulo contém funções:
 - *Cross platform*
 - *Windows Platform*
 - *Mac Os Platform*
 - *Java Platform*
 - *Unix Platform*

docs.python.org/3/library/platform.html

```
39 import platform as p
40 print(p.architecture())
41
42 #Tipo da máquina 32 ou 64 bits
43 print(p.machine())
44
45 #Nome do nó da rede
46 print(p.node())
47
48 #Detalhes sobre o processador
49 print(p.processor())
50
51 #Nome da plataforma
52 print(p.platform())
53
54 #Versão do SO
55 print(p.release())
56
57 #Nome do SO
58 print(p.system())
```


- Módulo ***urllib***
 - Fornece uma interface simples para acesso a recursos de rede
 - Permite acessar sites, fazer o download de dados, modificar seus cabeçalhos e fazer qualquer requisição GET ou POST.

```
1  import urllib.request
2
3  URL = "https://www.google.com/"
4
5  response = urllib.request.urlopen(URL)
6  print(response.read())
```

- Visualizando os cabeçalhos da requisição

```
1  import urllib.request
2
3  URL = "https://www.google.com/"
4
5  response = urllib.request.urlopen(URL)
6
7  #cabeçalhos da requisição
8  headers = response.info()
9  print('DATA      : %s' %headers['date'])
10 print(headers)
11
```

- Visualizando detalhes sobre a resposta

```
1  import urllib.request
2
3  URL = "https://www.google.com/"
4
5  response = urllib.request.urlopen(URL)
6
7  #dados retornados pela requisição
8  dados = response.read()
9  print('LENGTH : %d' %len(dados))
10 print(dados)
```

ATIVIDADE PRÁTICA

- ***Exercício 1:***
 - Escreva um programa capaz de:
 - Contar a quantidade de arquivos em um determinado *path*.
 - Contar a quantidade de diretórios em um determinado *path*.

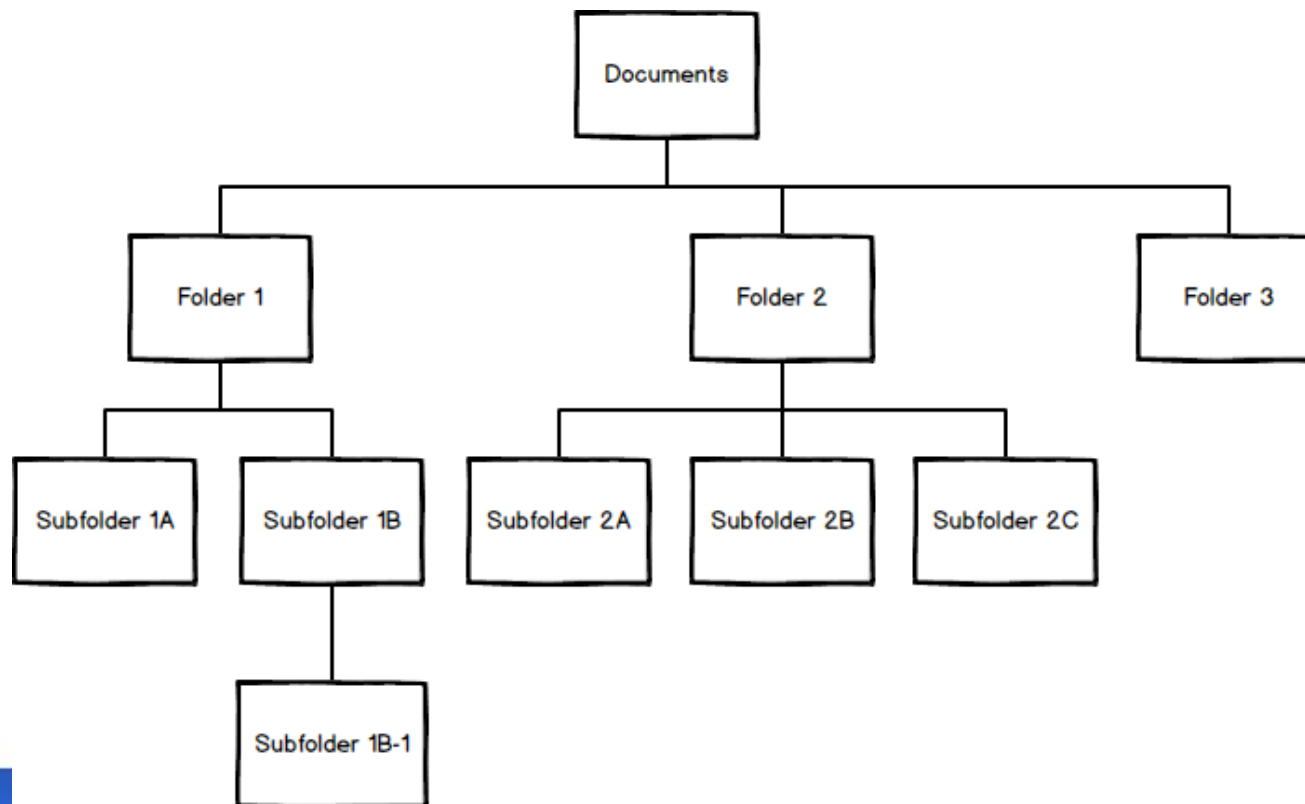
- ***Exercício 2:***

- Escreva um programa capaz de:

- Contar a quantidade de arquivos de uma determinada extensão. O usuário informará o *path* e também a extensão desejada.

- **Exercício 3:**

- Escreva um programa capaz de criar a seguinte hierarquia de diretórios:



- **Exercício 4:**

- Escreva um programa capaz de:

- Criar 10 diretórios identificados por *diretorio_XX*, em que XX é o número do diretório.
 - Em cada diretório criar dois arquivos identificados por *arquivo01.txt* e *arquivo02.txt*
 - Realize a compactação dos diretórios criados em um arquivo denominado ***backup.zip***.

- ***Exercício 5:***

- Escreva um programa capaz de:

- Criar um diretório e adicionar 100 arquivos aleatórios, em que estes arquivos podem ser de três extensões específicas: ABC, XYZ ou QWE.
 - Crie uma função para apagar apenas os arquivos com a extensão XYZ.

- **Exercício 6:** Escreva uma aplicação Python capaz de contar a quantidade de tags: `<p>`, `<h1>` e `<a>` e `` de uma página web.

- ***Exercício 7:*** Escreva uma aplicação Python capaz gravar um arquivo HTML contendo o resultado da requisição.

- **Exercício 8:** Escreva uma aplicação Python para:
 - Identificar o caminho de todas as imagens de uma página Web.
 - Realizar o download das imagens e salvar no disco.

- **Exercício 9:** Escreva uma aplicação Python para:
 - Extrair todos os hiperlinks de uma página Web e salvar o resultado em um arquivo texto.