

DESNEVOLVIMENTO PARA INTERNET

DANIEL CORRÊA DA SILVA

2015-1

CAPÍTULO 1

SCRIPT E ADMINISTRAÇÃO DE SISTEMAS EM PYTHON

ADMINISTRAÇÃO DE SISTEMAS EM PYTHON

1.1 – SCRIPT EM PYTHON QUE RECEBE COMANDOS DO TERMINAL:

```
1  #!/usr/local/bin/python3
2  # Arquivo: pipelab.py
3  # Objetivo: receber o comando por pipe
4
5  import fileinput
6
7  with fileinput.input() as f_input:
8      for line in f_input:
9          print(line, end='')
10
```

ADMINISTRAÇÃO DE SISTEMAS EM PYTHON

1.2 – SCRIPT QUE AUTENTICA USUARIO E SENHA POR TERMINAL:

```
1  #!/usr/local/bin/python3
2  # Arquivo: passwordLab.py
3  # Autenticar com o usuario e senha por terminal
4  import getpass
5
6  user = getpass.getuser()
7  passwd = getpass.getpass()
8  print(user,passwd)
```

ADMINISTRAÇÃO DE SISTEMAS EM PYTHON

1.3 – SCRIPT QUE EXECUTA UM COMANDO DO TERMINAL E CONVERTE EM STRING:

```
1  #!/usr/local/bin/python3
2  # Arquivo: comandoLab.py
3  # Objetivo: executar um comando e obter sua saída em uma string
4
5  import subprocess
6
7  #out_bytes = subprocess.check_output(['netstat', '-a'])
8  out_bytes = subprocess.check_output(['ls', '-la'])
9
10 #comando para interpretar os bytes como texto
11 out_bytes = out_bytes.decode('utf-8')
12 print (out_bytes)
```

ADMINISTRAÇÃO DE SISTEMAS EM PYTHON

1.4 – SCRIPT QUE COPIA E MOVE ARQUIVOS E DIRETÓRIOS

```
1  #!/usr/local/bin/python3
2  # Arquivo: arquivoDiretorioLab.py
3  # Objetivo: Copiar ou mover diretório utilizando python
4
5  import shutil
6
7  src = ''
8  dest = ''
9
10 # Copia src para dest
11 shutil.copy(src,dest)
12
13 # Copia arquivos, porém preserva os metadados (cp -p src dest)
14 shutil.copy2(src,dest)
15
16 # Copia árvore de diretório (cp -R src dest)
17 shutil.copytree(src,dest)
18
19 # Move src para dest (mv)
20 shutil.move(src,dest)
21
22 # Preservar links simbólicos em arquivos diretórios copiados
23 shutil.copy2(src,dest, follow_symlinks=False)
24 shutil.copytree(src,dest, symlinks=True)
```

ADMINISTRAÇÃO DE SISTEMAS EM PYTHON

1.5 – SCRIPT QUE COMPACTA E DESCOMPACTA ARQUIVOS

```
>>> import shutil
>>> shutil.get_archive_formats()
[('bztar', "bzip2'ed tar-file"), ('gztar', "gzip'ed tar-file"), ('tar', 'uncompressed tar file'), ('xztar', "xz'ed tar-file"), ('zip', 'ZIP file')]
```

```
1  #!/usr/local/bin/python3
2  # Arquivo: compactDescompactLab.py
3  # Objetivo: compactar e descompactar arquivos
4  #           e diretórios utilizando python
5
6  import shutil
7
8  shutil.unpack_archive('arqCompact.tar.gz')
9  shutil.make_archive('arqCompact', 'zip', 'dirTeste')
```

ADMINISTRAÇÃO DE SISTEMAS EM PYTHON

1.6 – SCRIPT QUE OBTÉM INFORMAÇÕES DE ARQUIVOS DE CONFIGURAÇÃO

```
;config.ini
;Exemplo de arquivo de configuração

[installation]
library=%(prefix)s/lib
include=%(prefix)s/include
bin=%(prefix)s/bin
prefix=/usr/local

#definições relacionadas com configuração de debug
[debug]
log_errors=true
show_warnings=False

[server]
port:8080
nworkers: 32
pid-file=/tmp/spam.pid
root=/www/root
```


ADMINISTRAÇÃO DE SISTEMAS EM PYTHON

1.6 – SCRIPT QUE OBTÉM INFORMAÇÕES DE ARQUIVOS DE CONFIGURAÇÃO

```
[>>> from configparser import ConfigParser
[>>> cfg=ConfigParser()
[>>> cfg.read('config.ini')
['config.ini']
[>>> cfg.sections()
['installation', 'debug', 'server']
[>>> cfg.get('installation','library')
'/usr/local/lib'
[>>> cfg.getboolean('debug','log_errors')
True
[>>> cfg.getint('server','port')
8080
[>>>
```