INSTITUTO FEDERAL DE
EDUCAÇÃO, CIÊNCIA E TECNOLOGIA
SUL-RIO-GRANDENSE
Campus Sapiranga

Linguagem de Programação III

Aula 13

Arquivos Zip
Funções Data e Hora

Prof. Guilherme Xavier

Compactando arquivos com o módulo zipfile

Seus programas Python podem criar e abrir (ou extrair) arquivos ZIP usando funções do módulo zipfile.

Lendo arquivos ZIP

Para ler o conteúdo de um arquivo ZIP, inicialmente você deve criar um objeto ZipFile (observe as letras Z e F maiúsculas). Conceitualmente, os objetos ZipFile são semelhantes aos objetos File retornados pela função open() que vimos no capítulo anterior: são valores por meio dos quais o programa interage com o arquivo

Para criar um objeto ZipFile, chame a função zipfile.ZipFile() passando-lhe uma string com o nome do arquivo .zip.
Observe que zipfile é o nome do módulo Python e ZipFile() é o nome da função.

Por exemplo, digite o seguinte no shell interativo:

```
>>> import zipfile, os
>>> os.chdir('C:\\temp\\documentos\')    # vai para a pasta que contém example.zip
>>> exampleZip = zipfile.ZipFile('example.zip')
>>> exampleZip.namelist()
['spam.txt',  'catnames.txt', 'zophie.jpg']
>>> spamInfo = exampleZip.getinfo('spam.txt')
>>> spamInfo.file_size
```

Criando arquivos ZIP e adicionando itens

Para criar seus próprios arquivos ZIP compactados, abra o objeto ZipFile em modo de escrita passando 'w' como segundo argumento.

(Isso é semelhante a abrir um arquivo-texto em modo de escrita passando 'w' à função open().)

Ao passar um path para o método write() de um objeto ZipFile, o Python compactará o arquivo nesse path e o adicionará ao arquivo ZIP.

O primeiro argumento do método write() é uma string que contém o nome do arquivo a ser adicionado.

O segundo argumento é o parâmetro referente ao tipo de compactação, que diz ao computador qual algoritmo deverá ser usado para compactar os arquivos; você poderá simplesmente definir esse valor sempre com zipfile.ZIP_DEFLATED. (Isso especifica o algoritmo de compressão *deflate*, que funciona bem para todos os tipos de dados.)

Digite o seguinte no shell interativo:

```
>>> import zipfile
>>> newZip = zipfile.ZipFile('new.zip', 'w')
>>> newZip.write('spam.txt', compress_type=zipfile.ZIP_DEFLATED)
>>> newZip.close()
```

Esse código cria um novo arquivo ZIP chamado new.zip que contém o conteúdo compactado de spam.txt.
Tenha em mente que, assim como na escrita em arquivos, o modo de escrita apagará qualquer conteúdo existente em um arquivo ZIP.

Se quiser simplesmente adicionar arquivos em um arquivo ZIP existente, passe 'a' como o segundo argumento de zipfile.ZipFile() para que o arquivo ZIP seja aberto em modo de adição

Extraindo itens de arquivos ZIP

O método extractall() de objetos ZipFile extrai todos os arquivos e as pastas de um arquivo ZIP no diretório de trabalho atual.

```
>>> import zipfile, os
>>> os.chdir('C:\\')  # vai para a pasta que contém example.zip
>>> exampleZip = zipfile.ZipFile('example.zip')
>>> exampleZip.extractall()
>>> exampleZip.close()
```

Após executar esse código, o conteúdo de example.zip será extraído para C:\. Opcionalmente, um nome de pasta pode ser passado para extractall() para que esse método extraia os arquivos em uma pasta que não seja o diretório de trabalho atual.

Se a pasta passada para o método extractall() não existir, ela será criada.

Por exemplo, se a chamada for substituída por exampleZip.extractall('C:\\delicious'), o código extrairá os arquivos de example.zip em uma nova pasta C:\delicious que será criada.

O método extract() dos objetos ZipFile extrai um único arquivo do arquivo ZIP.

.

Prossiga com o exemplo no shell interativo:

```
>>> exampleZip.extract('spam.txt')
'C:\\spam.txt'
>>> exampleZip.extract('spam.txt', 'C:\\some\\new\\folders')
254
'C:\\some\\new\\folders\\spam.txt'
>>> exampleZip.close()
```

A string passada para extract() deve coincidir com uma das strings da lista retornada por namelist().

Opcionalmente, um segundo argumento pode ser passado a extract() para extrair o arquivo em uma pasta que não seja o diretório de trabalho atual.

Se esse segundo argumento for uma pasta que ainda não exista, o Python a criará.

O valor retornado por extract() é o path absoluto em que o arquivo foi extraído.

```
#todos os arquivos - definindo terminação
import os
import zipfile

arquivo_zip = zipfile.ZipFile("c:\\temp\\documentos\\ifsul\\todos.zip", 'w')


for folder, subfolders, files in os.walk('C:\\temp\\documentos\\ifsul'):

    for file in files:
        if file.endswith('.pdf') or file.endswith('.docx'):
            arquivo_zip.write(os.path.join(folder, file),
                              os.path.relpath(os.path.join(folder, file), 'c:\\temp\\documentos\\if
                              compress_type=zipfile.ZIP_DEFLATED)


arquivo_zip.close()
```

Adicionar arquivos conforme a terminação
Defina o path do arquivo

```
arquivo_zip = zipfile.ZipFile("c:\\temp\\documentos\\todos.zip", 'w')
```

Criar um for para leitura do arquivo

```
for folder, subfolders, files in os.walk('C:\\temp\\documentos\\ifsul'):

    for file in files:
        if file.endswith('.pdf') or file.endswith('.docx'):
            arquivo_zip.write(os.path.join(folder, file),
                              os.path.relpath(os.path.join(folder, file), 'c:\\temp\\documentos'),
                              compress_type=zipfile.ZIP_DEFLATED)

arquivo_zip.close()
```

Exercícios
Acesse a pasta:
AulaPython\Programas\Apostilas\Python

1.1 Crie o arquivo varios.zip com todos os arquivos que estão na pasta
1.2 Liste todos os arquivos que estão compactados em varios.zip
1.3 Informe o tamanho do arquivo Python.docx que esta no arquivo varios.zip
1.4 Ciar um arquivo chamado *variospdf.zip* com todos os arquivos .pdf contidos na pasta
1.5 Extraia os arquivos python.docx na pasta AulaPython\Arquivos\Documentos

INSTITUTO FEDERAL DE
EDUCAÇÃO, CIÊNCIA E TECNOLOGIA
SUL-RIO-GRANDENSE
Campus Sapiranga

Um pouco sobre o Tempo

Módulo time traz varias funções para manipular o tempo

```
>>> import time
>>> agora=time.time()
>>> agora
1534256326.931621
```

Posição	Nome	Descrição
0	tm_year	Ano
1	tm_mon	Mês
2	tm_mday	Dia
3	tm_hour	Hora
4	tm_min	Minutos
5	tm_sec	Segundos
6	tm_wday	Dias da semana 0 a 6 – 0=segunda-feira
7	tm_yday	Dias do ano de 1 a 366
8	tm_isdst	Horario de verão, 1 = Horario de verão

Obtenção de informações sobre a hora

```
>>> import time
>>> agora=time.time()
>>> agora
1534256326.931621
>>> time.gmtime(agora)
time.struct_time(tm_year=2018, tm_mon=8, tm_mday=14, tm_hour=14,
tm_min=18, tm_sec=46, tm_wday=1, tm_yday=226, tm_isdst=0)
>>>
```

```
import os
import os.path
import time
 agora=time.time()
print("Ano: %d"  % agora.tm_year)
print("Mes: %d"  % agora.tm_year)
```

Faça um programa que mostre Dia, Hora, Minuto, Segundo,
Dia da Semana, Dia do Ano e se estamos ou não no horario de verão

Código de formatação strftime

Código	Descrição
%a	Dia da semana abreviado
%A	Nome do dia da semana
%b	Nome do mês abreviado
%B	Nome do mês completo
%c	Data e hora conforme configuração regional
%d	Dia do mês (01-31)
%H	Hora no formato 24H (00-23)
%I	Hora no formato 12H
%J	Dia do ano 0001-366

Código	Descrição
%m	Mês 01-12
%M	Minutos (00-59)
%p	AM ou PM
%S	Segundos (00-61)
%U	Numero de semana (00-53) aonde a semana 1 começa após o primeiro domingo .
%w	Dia da semana 90-6) onde 0 é domingo
%W	Numero de semana (00-53) aonde a semana 1 começa após o primeiro segunda-feira .
%x	Representação regional da data
%X	Representação regional da hora
%y	Ano(00-99)
%Y	Ano com 4 dígitos
%Z	Nome do fuso horário
%	Símbolo de %

Obtenção de informações sobre o arquivo

```
import os
import os.path
import time
import sys
nome = sys.argv[0]
print("Nome: %s" % nome)
print("Tamanho: %d" % os.path.getsize(nome))
print("Criado: %s" % time.ctime(os.path.getctime(nome)))
print("Modificado: %s" % time.ctime(os.path.getmtime(nome)))
print("Acessado: %s" % time.ctime(os.path.getatime(nome)))
```

Exercícios

2.1 Sabendo-se que o ano letivo termina em 21/12/2018, faça um programa que informe:
O total de dias letivos no ano
O total de dias transcorrido até a data atual.

2.2 Faça um programa que receba uma data qualquer e informe o dia da semana.

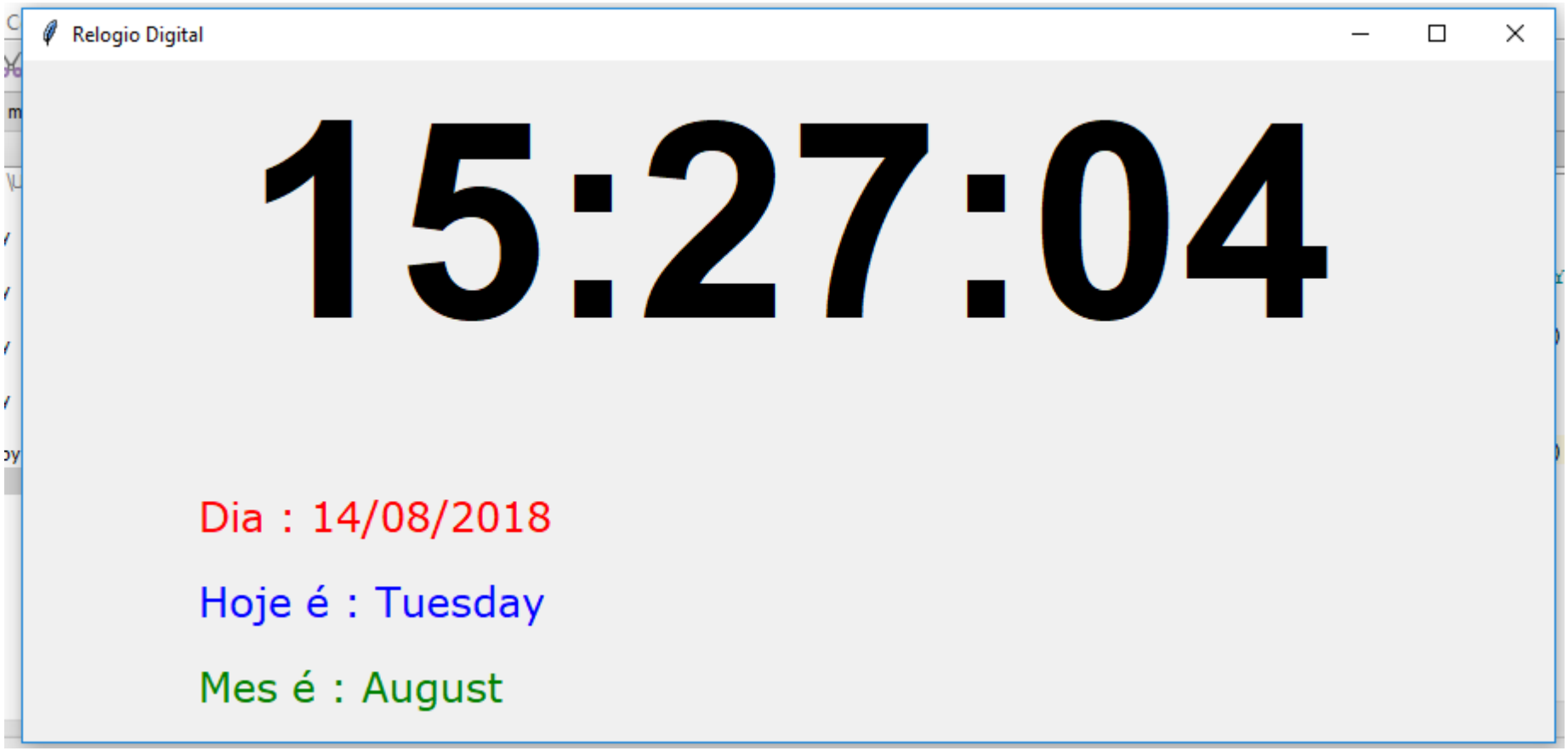
2.3 Faça um programa que receba uma data qualquer e informe o dia da semana e o mês por extenso.

Exercícios

2.4. Faça um programa que receba a data de nascimento de uma Pessoa e informe sua idade e o dia da semana que a pessoa nasceu.

2.5 Crie um relógio digital que, alem da hora, contenha

- ✓ Data atual
- ✓ O dia da Semana
- ✓ Mês por extenso


Relogio Digital
15:27:04
Dia : 14/08/2018
Hoje é : Tuesday
Mes é : August