PYTHON FILES, PACKAGE AND MODULES

By http://www.HadoopExam.com

Note: These instructions should be used with the HadoopExam Apache OOzie: Professional Trainings. Where it is executed and you can do hands on with trainer.

Content:

- Create a file using Python
- Read a file using Python
- Python OS Module
- String Formatting
- OS Modules
- PIPES and Subprocesess
- Python Module
- __name_ frames

Step 1: Create a file

- If the file already exists, opening it in write mode clears out the old data and starts fresh, so be careful!
- If the file doesn't exist, a new one is created.
- Open returns a file object that provides methods for working with the file.
- The write method puts data into the file.

```
fout = open('C:/test/hadoopexam.txt', 'w')
fout.write('Hello and Welcome to \n')
fout.write('HadoopExam Learning Resources')
fout.close()
```

Step 2: The argument of write has to be a string, so if we want to put other values in a file, we have to convert them to strings.

```
fout = open('C:/test/hadoopexam.txt', 'w')
fee=3500
fout.write(str(fee))
fout.flush()
```

Step 3: String formatting

```
fee = 3900
'%d' % fee

'Total fee of Hadoop Course is %d INR' %fee
```

Note: If there is more than one format sequence in the string, the second argument has to be a tuple. Each format sequence is matched with an element of the tuple.

```
'%d' to format an integer
'%g' to format a floating-point number
'%s' to format a string
```

Step 4: Using tuple as well having % sign in final string.

```
num_of_courses=2
discount=10.5
course_name='Hadoop & Spark'

'Total fee for %d courses comes with %g%% discount e.g %s' %(num_of_courses , discount ,
course_name)
```

Step 5: The number of elements in the tuple has to match the number of format sequences in the string. And, also the types of the elements have to match the format sequences.

```
'%d %d %d' % (2900, 3900)
'%d' % 'Discount'
```

Step 6: OS Module

- The os module provides functions for working with files and directories.
- **os.getcwd** returns the name of the current directory.

```
import os
cwd = os.getcwd()
cwd
```

Step 7: Getting the absolute path of a file.

```
os.path.abspath('hadoopexam.txt')
os.path.exists('C:\\TempWork\\eclipse\\neon\\eclipse\\')
os.path.isdir('C:\\TempWork\\eclipse\\neon\\eclipse\\')
os.path.isfile('C:\\TempWork\\eclipse\\neon\\eclipse\\')
os.listdir(os.getcwd()) #list current working directory
```

Exception Handling

Step 8: When you try to open a file and it does not exists.

```
try:
        fin = open('c:/hadoopexam.txt')
except:
        print('There is a problem, while opening the file')
```

Python starts by executing the try clause. If all goes well, it skips except clause and proceeds. If an exception occurs, it jumps out of the try clause and runs an except clause. Handling an exception with a try statement is called catching an exception.

Pipes: It helps you to execute os command. Now let say we want to list the content of current directory.

We can use following program

```
Import os
command='dir' # OS Commnad
pipeopen = os.popen(command) # execute OS command
result = pipeopen.readlines(10) #Read first 10 lines of command output, check other methods as well
print (result) #Print the result
pipeopen.close() #It is the best practice to close pipe
```

Opening Notepad using PIPES

```
import os
command='notepad' # OS Commnad
pipeopen = os.popen(command) # execute OS command
pipeopen.close() #It is the best practice to close pipe
```

subprocess Module

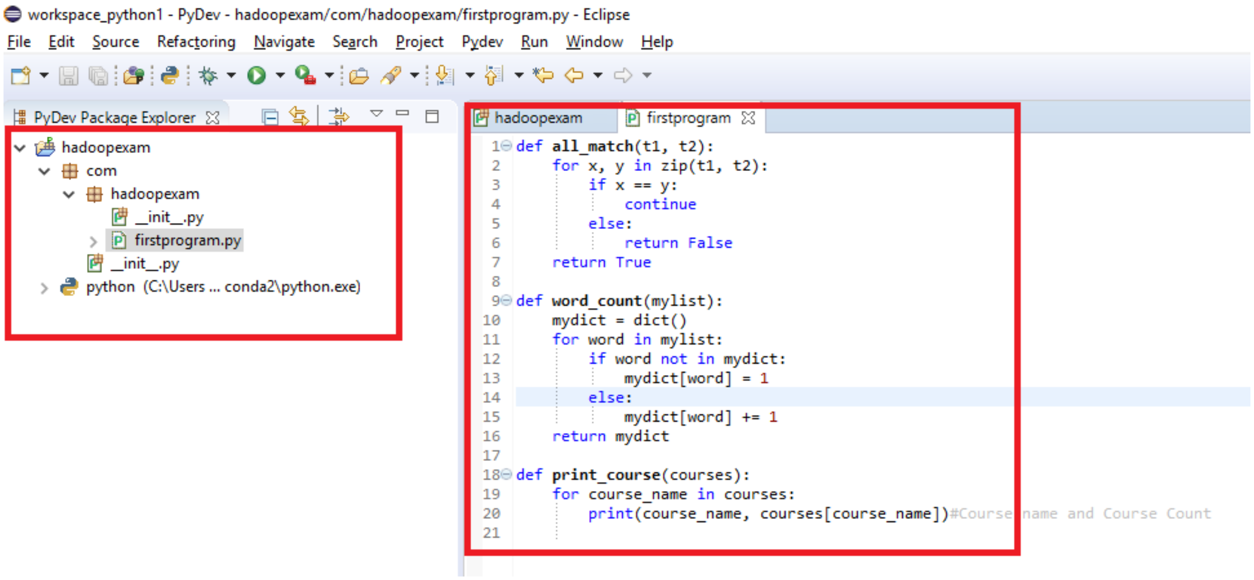
```
import subprocess
subprocess.call('notepad')
```

- PIPE: represent a running program, however, you should use **subprocess** module for such activity.
- Popen, behaves very similar to opening a file.

Python Modules

- Any file that contains Python code can be imported as a module.
- Modules in Python are simply Python files with the .py extension, which implement a set of functions. Modules are imported from other modules using the import command.
- The first time a module is loaded into a running Python script, it is initialized by executing the code in the module once. If another module in your code imports the same module again, it will not be loaded twice but once only
- Packages are namespaces which contain multiple packages and modules themselves. They are simply directories.
- Each package in Python is a directory which MUST contain a special file called __init__.py
- This file can be empty, and it indicates that the directory it contains is a Python package, so it can be imported the same way a module can be imported.

Step 9: Create a Python Module Name “firstprogram” as shown below.


workspace_python1 - PyDev - hadoopexam/com/hadoopexam/firstprogram.py - Eclipse
File Edit Source Refactoring Navigate Search Project Pydev Run Window Help
PyDev Package Explorer
hadoopexam
com
hadoopexam
__init__.py
firstprogram.py
__init__.py
python (C:\Users ... conda2\python.exe)
hadoopexam
firstprogram

```
def all_match(t1, t2):
    for x, y in zip(t1, t2):
        if x == y:
            continue
        else:
            return False
    return True

def word_count(mylist):
    mydict = dict()
    for word in mylist:
        if word not in mydict:
            mydict[word] = 1
        else:
            mydict[word] += 1
    return mydict

def print_course(courses):
    for course_name in courses:
        print(course_name, courses[course_name])#Course name and Course Count
```

- Create a pydev project hadoopexam
- Create a package com.hadoopexam
- Create a Module firstprogram
- Write below code in Module

```
def all_match(t1, t2):
    for x, y in zip(t1, t2):
        if x == y:
            continue
        else:
            return False
    return True

def word_count(mylist):
    mydict = dict()
    for word in mylist:
        if word not in mydict:
            mydict[word] = 1
        else:
            mydict[word] += 1
    return mydict

def print_course(courses):
    for course_name in courses:
    print(course_name, courses[course_name])#Course name and Course Count
```

Step 10: Now use this module in Python console.

```
from com.hadoopexam.firstprogram import word_count
from com.hadoopexam.firstprogram import print_course


site1 = ['Hadoop', 'Spark', 'Python','Hadoop', 'Spark', 'Python','Hadoop', 'Spark', 'Python','Hadoop',
'Spark', 'Python','Hadoop', 'Spark', 'Python' , 'Java' , 'SQL' , 'Unix' , 'Cloud Computing' , 'Java' , 'SQL' ,
'Unix' , 'Cloud Computing','Hadoop', 'Spark', 'Python','Hadoop', 'Spark', 'Python','Hadoop', 'Spark',
'Python','Hadoop', 'Spark', 'Python','Hadoop', 'Spark', 'Python' , 'Java' , 'SQL' , 'Unix' , 'Cloud
Computing' , 'Java' , 'SQL' , 'Unix' , 'Cloud Computing']


#Call course count
course_count = word_count(site1)
print(course_count)

print_course(course_count)
```

Step 11: Get all the internal information about the module.

```
dir(firstprogram)
```

What does *if __name__ == “__main__”*: do?

Create a file name: pydev_thread.py with following content

```
import os

def open_notepad(string, sleeptime, lock, *args):
    command='notepad' # OS Commnad
    pipeopen = os.popen(command) # execute OS command
    pipeopen.close() #It is the best practice to close pipe

if __name__ == "__main__":
    command='dir' # OS Commnad
    pipeopen = os.popen(command) # execute OS command
    result = pipeopen.read() #Read first 10 lines of command output, check other
methods as well
    print (result) #Print the result
    pipeopen.close() #It is the best practice to close pipe


print(__name__)
```

- When the Python interpreter reads a source file, it executes all of the code found in it.
- Before executing the code, it will define a few special variables. For example, if the python interpreter is running that module (the source file) as the main program, it sets the special **__name__ variable to have a value "__main__".** If this file is being imported from another module, **__name__ will be set to the module's name**.
- In the case of your script, let's assume that it's executing as the main function, e.g. you said something like

    python threading_example.py

    on the command line. After setting up the special variables, it will execute the import statement and load those modules. It will then evaluate the def block, creating a function object and creating a variable called **open_notepad** that points to the function object. It will then read the if statement and see that **__name__ does equal "__main__",** so it will execute the block shown there.

- One reason for doing this is that sometimes you write a module (a .py file) where it can be executed directly. Alternatively, it can also be imported and used in another module. By doing the main check, you can have that code only execute when you want to run the module as a program and not have it execute when someone just wants to import your module and call your functions themselves.
- Add import statement in a module and test it.

All Products List of www.HadoopExam.com

TRAINING'S (AVAILABLE)

- Hadoop BigData Professional Training
- HBase (NoSQL) Professional Training
- Apache Spark Professional Training
- Apache OOZie (Hadoop workflow) Professional Training
- Beginner AWS Training Course- **(HETRNAWS101)**
- Core Java 1z0-808 Exam training
- JAX-WS (Java WebService HandsOn Training)
- Scala Programming Training
- Python Programming Training
- Hortonworks Administration Professional Trainings

MAPR HADOOP AND NOSQL CERTIFICATION (AVAILABLE)

- **MapR Hadoop Developer Certification**
- **MapR HBase NoSQL Certification**
- **MapR Spark Developer Certification (In Progress)**

HORTONWORKS HADOOP AND NOSQL CERTIFICATION (AVAILABLE)

- HDPCD : NO Java (Hortonworks Developer Certification)
- HDPCD : Spark (Spark Developer Certifications)
- HDPCA : Hortonworks Administration Certification
- Hortonworks Administration Professional Trainings

CLOUDERA HADOOP AND SPARK CERTIFICATION (AVAILABLE)

- CCA131 : Hadoop Administrator
- CCA-175 Cloudera® (Hadoop and Spark Developer)
- CCP:DE575 : Cloudera® Data Engineer Certification
- CCA159 : Cloudera Data Analyst Certifications

DATABRICKSA OREILLY SPARK CERTIFICATION (AVAILABLE)

- Apache Spark Developer

AWS: AMAZON WEBSERVICE CERTIFICATION (AVAILABLE)

- AWS Solution Architect : Associate
- AWS Solution Architect: Professional
- AWS Developer : Associate
- AWS Sysops Admin : Associate

MICROSOFT AZURE CERTIFICATION (AVAILABLE)

- Azure 70-532
- Azure 70-533

DATA SCIENCE CERTIFICATION (AVAILABLE)

- EMC E20-007

EMC CERTIFICATIONS (AVAILABLE)

- EMC E20-007

SAS ANALYTICS CERTIFICATION (AVAILABLE)

- SAS Base A00-211
- SAS Advanced A00-212
- SAS Analytics : A00-240
- SAS Administrator : A00-250

ORACLE JAVA CERTIFICATION (AVAILABLE)

- Java 1z0-808
- Java 1z0-809

- Java 1z0-897 (Java WebService Certification)

ORACLE DATABASE CLOUD CERTIFICATION (AVAILABLE)

- 1z0-060 (Oracle 12c)
- 1z0-061 (Oracle 12c)

Subscribe Here for Regular Updates: Like New Training Module launched

Become Author and Trainer: We are looking for Author (Writing Technical Books) and Trainer (Creating Training Material): **No Compromise on Quality.**

Benefit: You will get very good revenue sharing. Please drop us an email to hadoopexam@gmail.com (For the skills, you feel you are master)

We are sure, you are good at least one technology. Don’t limit your potential, contact us immediately with your skill. Our expert team will contact you with more detail. You training and Books will reach to all our existing network and with our expert marketing team we will help you to reach as much as technical professional, with our Smart Advertising network. Contact us with sending an email hadoopexam@gmail.com

Opportunity to share your knowledge with all learners who are in need. We are helping 1000's of learners since last 4 years and established ourselves with Quality low cost material.