

Lecture-23

Renaming a Directory or a File:

The rename() method can rename a directory or a file.

For renaming any directory or file, the rename() method takes in two basic arguments: the old name as the first argument and the new name as the second argument.

GALGOTIAS
UNIVERSITY

School of Basic and Applied Sciences

Course Code : BSCM 304

Course Name: Programming Using Python

Example:

```
import os
test=os.listdir("D:\\Python directory" )
print(test)
print("\n")
```

Directory list:

```
['12.py', '12345.py', 'abj.txt', 'Arti.txt', 'foo.txt', 'Love', 'naman.txt', 'pawan.py', 'pawan.txt', 'rr.py', 'test']
```

```
os.rename('test','test_new')
```

```
test=os.listdir("D:\\Python directory" )
print(test)
```

updated Directory list:

```
['12.py', '12345.py', 'abj.txt', 'Arti.txt', 'foo.txt', 'Love', 'naman.txt', 'pawan.py', 'pawan.txt', 'rr.py', 'test_new']
```

Move or Copy Files and Directories:

Let's say we want to copy or move files and directories around, but don't want to do it by calling out to shell commands. The **shutil module** has portable implementations of functions for copying files and directories.

GALGOTIAS
UNIVERSITY

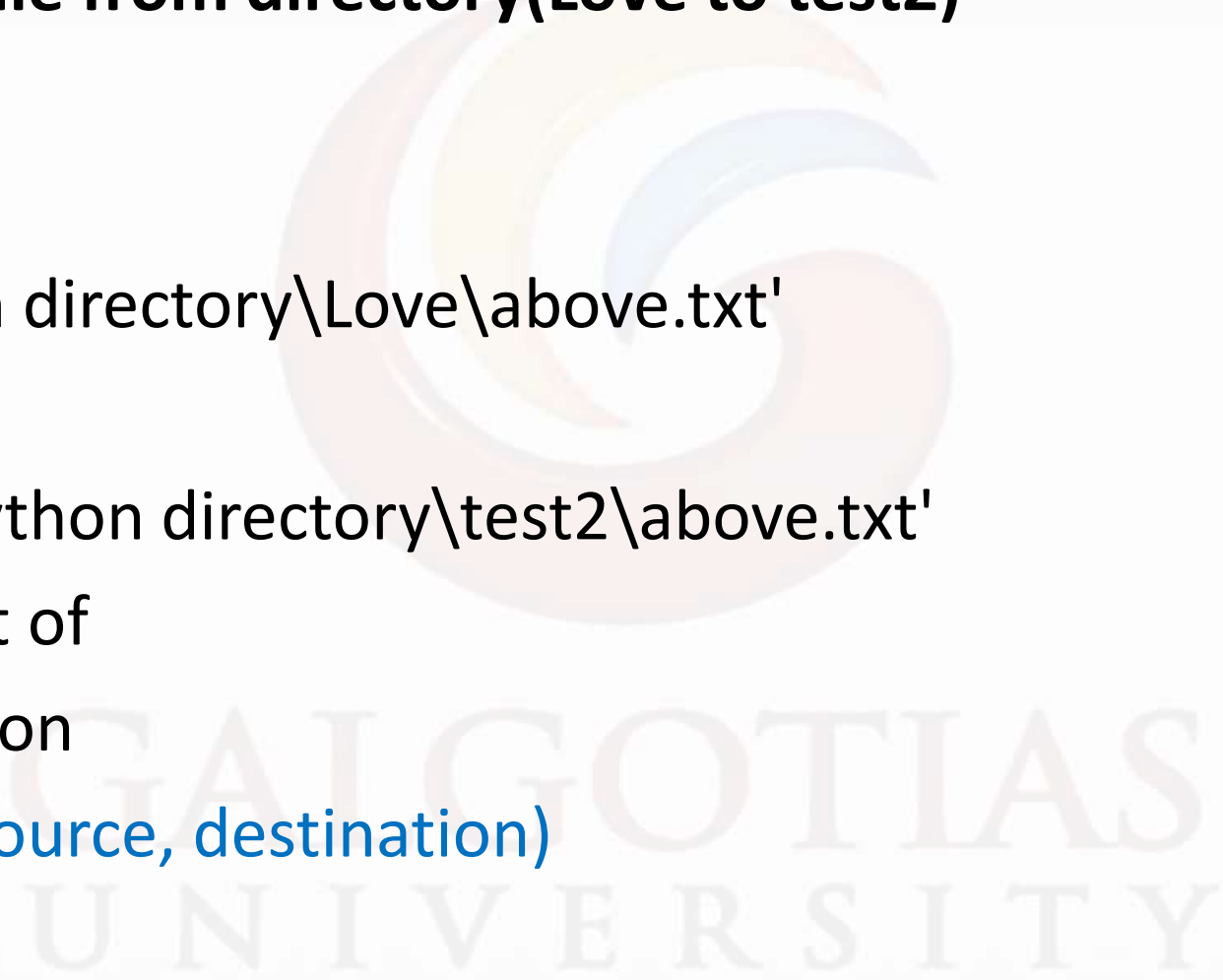
Example1: Move directory Love\new to test2\ram.

```
import shutil  
  
# Source path  
source = r'D:\Python directory\Love\New'  
  
# Destination path  
destination = r'D:\Python directory\test2\ram'  
  
# Move the content of  
# source to destination  
dest = shutil.move(source, destination)
```

GALGOTIAS
UNIVERSITY

Example2: Move a file from directory(Love to test2)

```
import shutil
# Source path
source = r'D:\Python directory\Love\above.txt'
# Destination path
destination = r'D:\Python directory\test2\above.txt'
# Move the content of
# source to destination
dest = shutil.move(source, destination)
import shutil
```



```
# Copy source to destination. (cp source to destination)
```

```
#shutil.copy(source, destination)
```

```
import shutil
```

```
# Source path
```

```
source = 'D:\Python directory\Love'
```

```
# Destination path
```

```
destination = 'D:\Python directory\test2'
```

```
dest =shutil.copy(source,destination)
```

GAJAGOTIAS
UNIVERSITY

How to Traverse a Directory Tree in Python – Guide to os.walk

When you use a scripting language like Python, one thing you will find yourself doing over and over again is walking a directory tree, and processing files. While there are many ways to do this, Python offers a built-in function that makes this process a breeze.

Basic Python Directory Traversal

Here's a really simple example that walks a directory tree, printing out the name of each directory and the files contained:

GALGOTIAS
UNIVERSITY

School of Basic and Applied Sciences

Course Code : BSCM 304

Course Name: Programming Using Python

Let's say we have a directory tree that looks like this:

```
+--- test.py
|
+--- [subdir1]
|   |
|   +--- file1a.txt
|   +--- file1b.png
|
+--- [subdir2]
|
+--- file2a.jpeg
+--- file2b.html
```



GALGOTIAS
UNIVERSITY

Example1:

```
# Import the os module, for the os.walk function
import os

# Set the directory you want to start from
rootDir = 'D:\Python directory'

t= os.walk(rootDir)

for dirName, subdirList, fileList in t:
    print('Found directory: %s' % dirName)
    for fname in fileList:
        print('\t%s' % fname)
```

GALGOTIAS
UNIVERSITY

School of Basic and Applied Sciences

Course Code : BSCM 304

Course Name: Programming Using Python

OUTPUT:

Found directory: D:\Python directory

12.py
12345.py
Aarti.txt
abj.txt
copy.py
direcotry tree change.py
direcotry tree.py
file type.py
foo.txt
Love foo.txt
naman.txt
pawan.py
pawan.txt
pawan2.py



GALGOTIAS
UNIVERSITY

Name of the Faculty: Dr. O P Verma

Program Name: B.Sc. (Mathematics)

School of Basic and Applied Sciences

Course Code : BSCM 304

Course Name: Programming Using Python

OUTPUT Continue...

```
print dir and sub dirlist.py
```

```
  Revrse number.py
```

```
  rr.py
```

```
Found directory: D:\Python directory\ test2
```

```
  abj.txt
```

```
  fool.txt
```

```
Found directory: D:\Python directory\copy1
```

```
Found directory: D:\Python directory\Love
```

```
  abj.txt
```

```
  above.txt
```

```
  direcotry tree.py
```

```
  foo.txt
```

```
  ram.txt
```

```
  suraj.txt
```

GALGOTIAS
UNIVERSITY

Name of the Faculty: Dr. O P Verma

Program Name: B.Sc. (Mathematics)

Example2:

```
# Import the os module, for the os.walk function
import os

# Set the directory you want to start from
rootDir = 'D:\Python directory\Love'
for dirName, subDirList, fileList in os.walk(rootDir):
    print('Found directory: %s' % dirName)
    for fname in fileList:
        print('\t%s' % fname)
```

OUTPUT:

Found directory: D:\Python directory\love

abj.txt

above.txt

direcotry tree.py

foo.txt

ram.txt

suraj.txt



GALGOTIAS
UNIVERSITY

Changing the Way the Directory Tree is Traversed

By default, Python will walk the directory tree in a top-down order (a directory will be passed to you for processing), *then* Python will descend into any sub-directories. We can see this behaviour in the output above; the parent directory (.) was printed first, then its 2 sub-directories.

Sometimes we want to traverse the directory tree bottom-up (files at the very bottom of the directory tree are processed first), then we work our way up the directories. We can tell `os.walk` to do this via the `topdown` parameter:

GALGOTIAS
UNIVERSITY

Example:

```
# Import the os module, for the os.walk function
import os

# Set the directory you want to start from
rootDir = 'D:\Python directory'
for dirName, subDirList, fileList in os.walk(rootDir, topdown=False):
    print('Found directory: %s' % dirName)
    for fname in fileList:
        print('\t%s' % fname)
```

School of Basic and Applied Sciences

Course Code : BSCM 304

Course Name: Programming Using Python

OUTPUT:

Found directory: D:\Python directory\ test2

abj.txt

fool.txt

Found directory: D:\Python directory\copy1

Found directory: D:\Python directory\Love

abj.txt

above.txt

direcotry tree.py

foo.txt

ram.txt

suraj.txt



GALGOTIAS
UNIVERSITY

School of Basic and Applied Sciences

Course Code : BSCM 304

Course Name: Programming Using Python

OUTPUT Continue.....

Found directory: D:\Python directory

12.py

12345.py

Aarti.txt

abj.txt

copy.py

direcotry tree change.py

direcotry tree.py

file type.py

foo.txt

Love foo.txt

naman.txt

pawan.py

pawan.txt

pawan2.py

print dir and sub dirlist.py

Revrse number.py

rr.py



GALGOTIAS
UNIVERSITY

Name of the Faculty: Dr. O P Verma

Program Name: B.Sc. (Mathematics)

Selectively Recursing Into Sub-Directories

The examples so far have simply walked the entire directory tree, but `os.walk` allows us to selectively skip parts of the tree.

For each directory `os.walk` gives us, it also provides a list of sub-directories (in `subdirList`). If we modify this list, we can control which sub-directories `os.walk` will descend into. Let's tweak our example above so that we skip the first sub-directory.

GALGOTIAS
UNIVERSITY

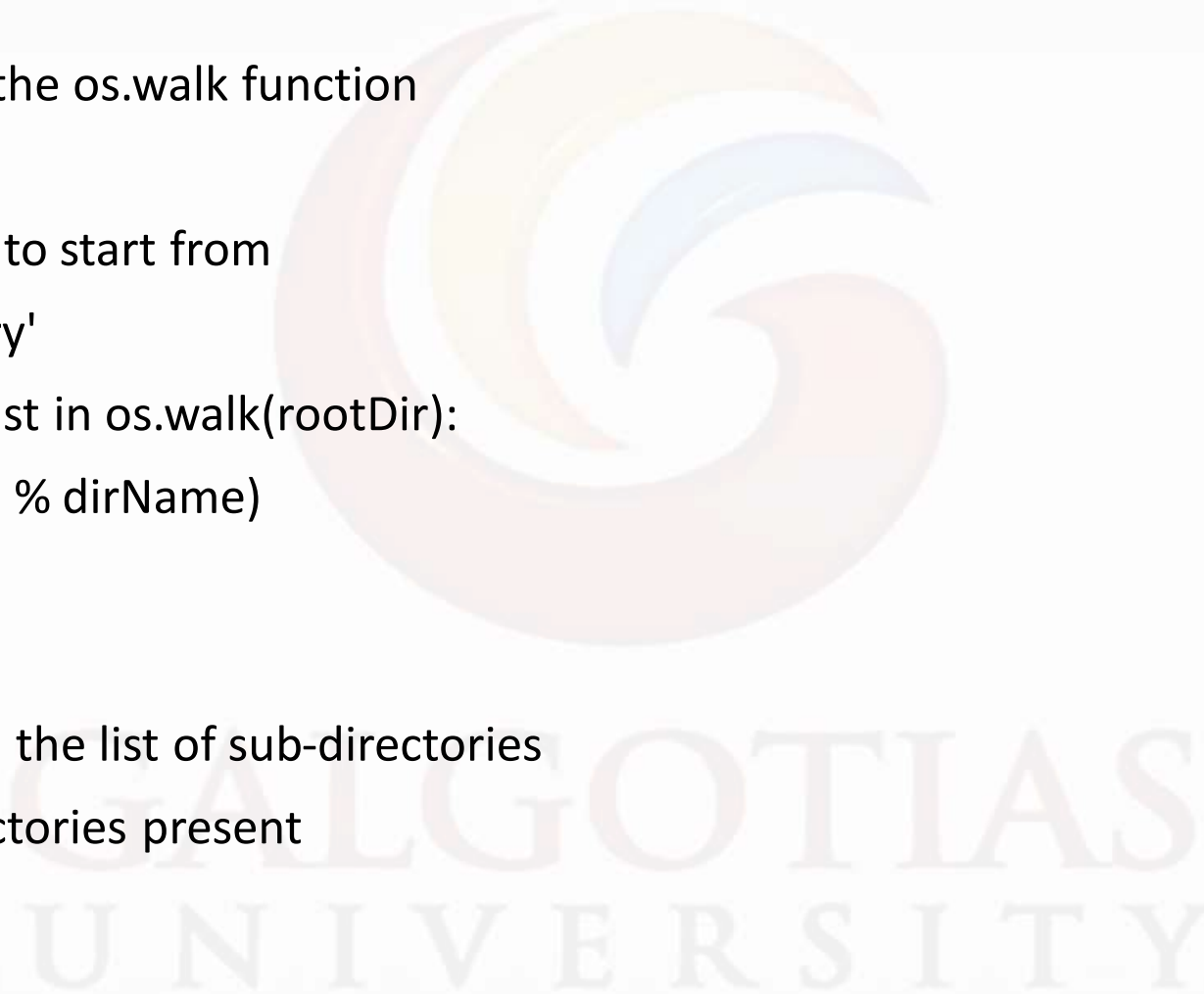
Example1:

```
# Import the os module, for the os.walk function
import os

# Set the directory you want to start from
rootDir = 'D:\Python directory'

for dirName, subdirList, fileList in os.walk(rootDir):
    print('Found directory: %s' % dirName)
    for fname in fileList:
        print('\t%s' % fname)

# Remove the first entry in the list of sub-directories
# if there are any sub-directories present
if len(subdirList) > 0:
    del subdirList[0]
```



School of Basic and Applied Sciences

Course Code : BSCM 304

Course Name: Programming Using Python

OUTPUT:

Found directory: D:\Python directory

12.py

12345.py

Aarti.txt

abj.txt

copy.py

direcotry tree change.py

direcotry tree.py

file type.py

foo.txt

Love foo.txt

naman.txt

pawan.py

pawan.txt

pawan2.py



Name of the Faculty: Dr. O P Verma

Program Name: B.Sc. (Mathematics)

OUTPUT Continue....

```
print dir and sub dirlist.py
```

```
  Revrse number.py
```

```
  rr.py
```

```
Found directory: D:\Python directory\test2
```

```
  abj.txt
```

```
  fool.txt
```

NOTE: We can see that the first sub-directory (*Love*) was indeed skipped

GALGOTIAS
UNIVERSITY

References:

1. Introduction to Computation and Programming using Python, by John Guttag, PHI Publisher
2. Python Programming using problem solving Approach by Reema Thareja, Oxford University, Higher Education Oxford University Press; First edition (10 June 2017), ISBN-10: 0199480173
3. Fundamentals of Python first Programmes by Kenneth A Lambert, Copyrighted material Course Technology Inc. 1 st edition (6th February 2009)
4. <https://www.tutorialspoint.com/python/index.htm>
5. <https://www.geeksforgeeks.org/python-programming-language>

*****END OF THE LECTURE*****

*****THANK YOU*****

GALGOTIAS
UNIVERSITY