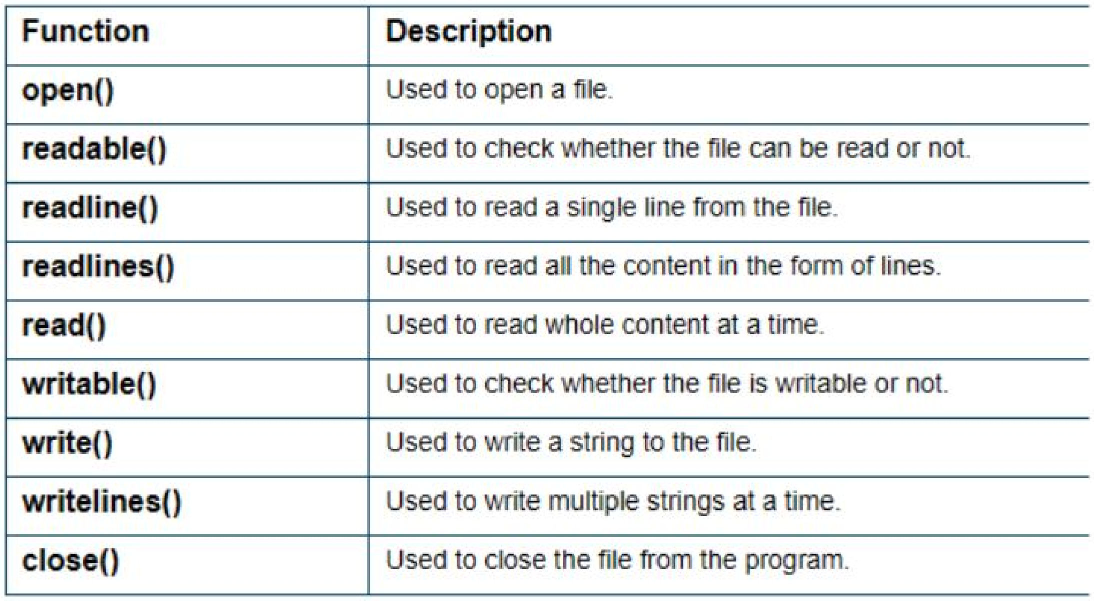
PW SKILLS

Files Handling Basics

Lesson Plan

File handling in Python is a key aspect for dealing with files—whether it's reading from or writing to them. Python provides built-in functions to open, read, write, and manipulate files easily.

Here is a list of some commonly used file-handling functions in Python.



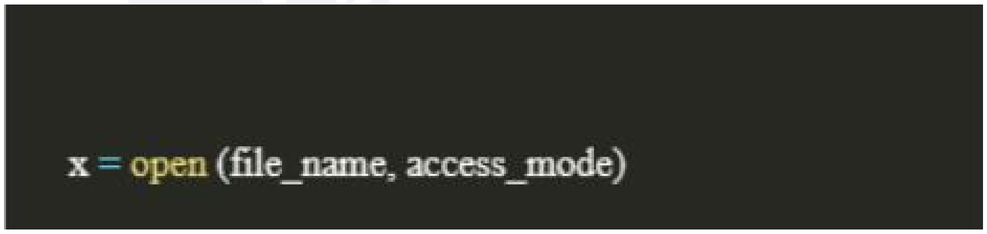
Function	Description
open()	Used to open a file.
readable()	Used to check whether the file can be read or not.
readline()	Used to read a single line from the file.
readlines()	Used to read all the content in the form of lines.
read()	Used to read whole content at a time.
writable()	Used to check whether the file is writable or not.
write()	Used to write a string to the file.
writelines()	Used to write multiple strings at a time.
close()	Used to close the file from the program.

How to open a file in Python

Python has a built-in function open() to open the file.

- open () returns a file object that is called a handle.
- It is used to read or modify the file.

Syntax



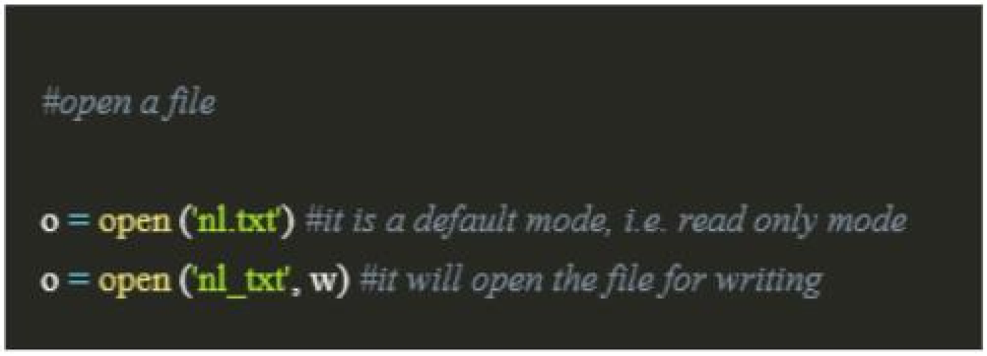
```
x = open (file_name, access_mode)
```

Parameters:
file_name: The name of the file that you want to open
access_mode: read, write, append

Modes:
'r' : Read (default). Open the file for reading, and the file pointer is placed at the beginning of the file.
'w' : Write. Open the file for writing. If the file exists, it is truncated (emptied). If it does not exist, a new file is created.
'a' : Append. Open the file for appending. Data is added at the end of the file.
'b' : Binary mode. Used with other modes for binary files like images or executables (e.g., 'rb' or 'wb').

'x' : Exclusive creation. Open the file for writing only if it doesn’t exist.

Example:



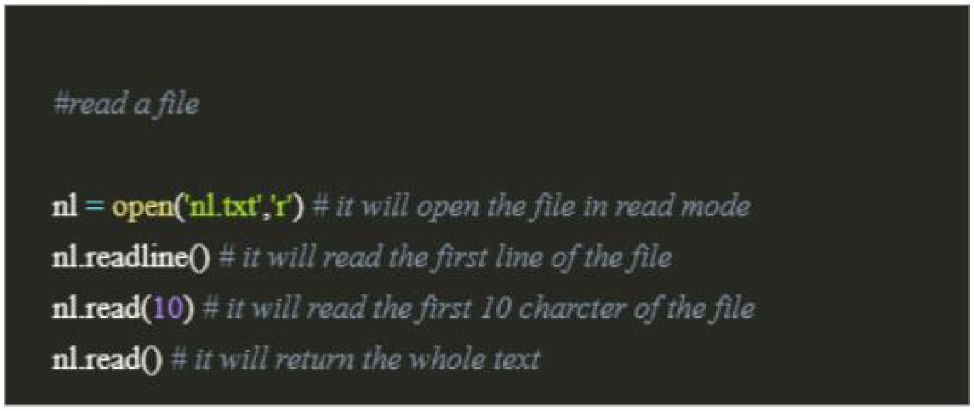
```
#open a file

o = open ('nl.txt') #it is a default mode, i.e. read only mode
o = open ('nl_txt', w) #it will open the file for writing
```

How to read a file in Python

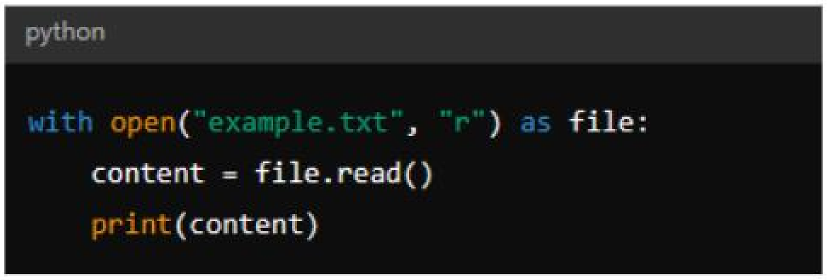
To read a file in Python, the file must first be opened in the read mode.

- Python offers several methods of reading.
- If you want a read-only first line only, use readline (), and if you want to read all the lines in the form of lines, use readlines ()
- If you want to read a few characters of the file, use read (size)
- read () used to read all the content at a time.

Example:



```
#read a file

nl = open('nl.txt','r') # it will open the file in read mode
nl.readline() # it will read the first line of the file
nl.read(10) # it will read the first 10 charcter of the file
nl.read() # it will return the whole text
```



python

```python
with open("example.txt", "r") as file:
    content = file.read()
    print(content)
```

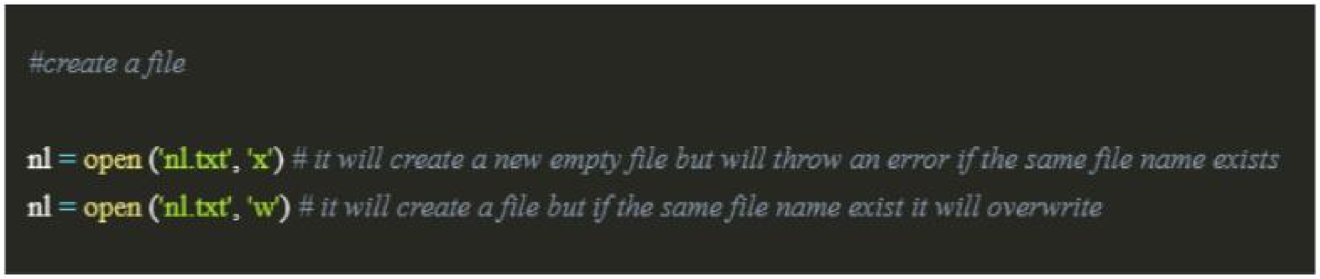
How to Create a new file in Python

To create a new file, we have to open a file using one of the two parameters:

x: it will create a new empty text file iff there does not exist any file with the same name; otherwise, it will throw an error

w: it will create a file, whether any file exists with the same or not, i.e., it will not return an error if the same file name exists.

Example:



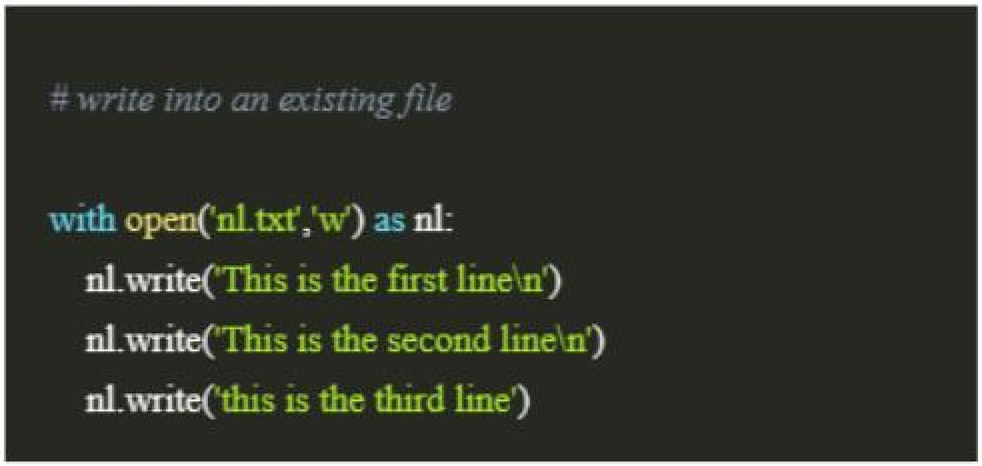
```python
#create a file

nl = open ('nl.txt', 'x') # it will create a new empty file but will throw an error if the same file name exists
nl = open ('nl.txt', 'w') # it will create a file but if the same file name exist it will overwrite
```

How to Write in an existing file in Python

Python offers two methods to write in an existing file:

- write () method
- It will add a single line at a time

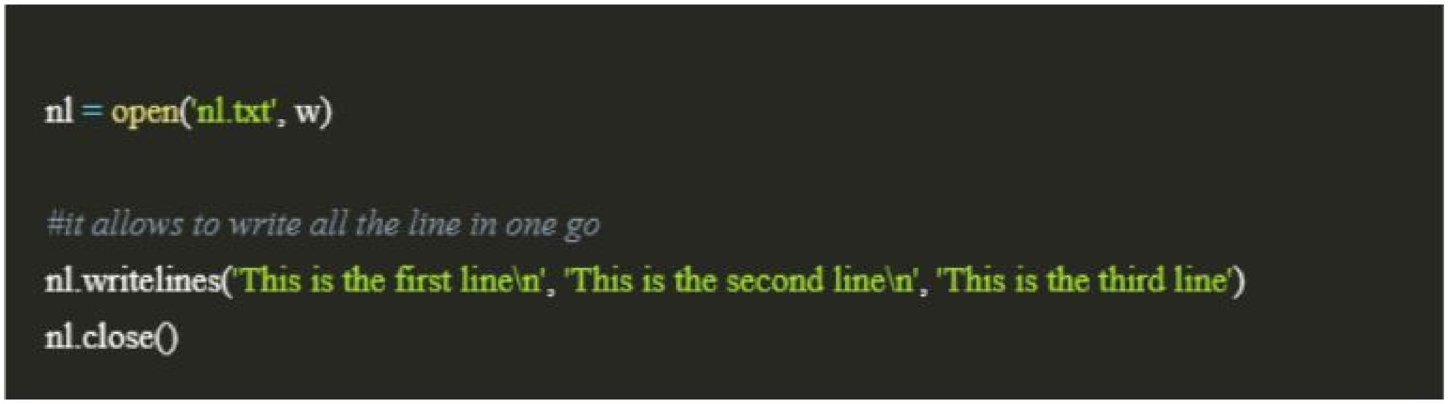
Example:



```python
# write into an existing file

with open('nl.txt','w') as nl:
    nl.write('This is the first line\n')
    nl.write('This is the second line\n')
    nl.write('this is the third line')
```

- writelines () method
- It allows to insert multiple string at a time

Example:



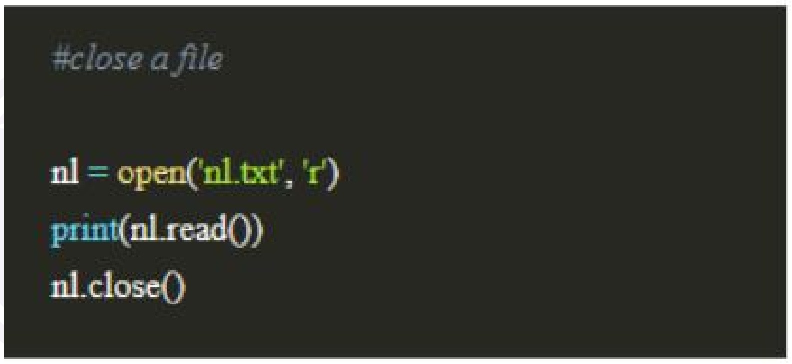
```
nl = open('nl.txt', w)

#it allows to write all the line in one go
nl.writelines('This is the first line\n', 'This is the second line\n', 'This is the third line')
nl.close()
```

How to close a file in Python

Once you are done with all the operations, you need to close the file correctly.

- The close () commands will terminate all the resources and will free the system
- It is a good practice to close the file.

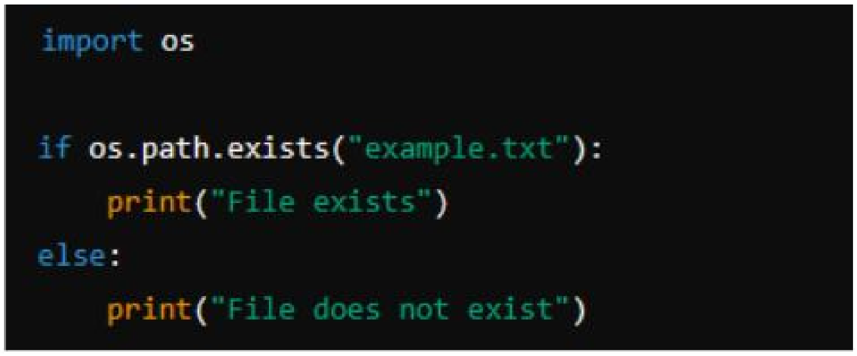
Example:



```
#close a file

nl = open('nl.txt', 'r')
print(nl.read())
nl.close()
```

Checking if a file exists:

Sometimes, you may want to check if a file exists before performing operations on it. You can do this using the ‘os’ module.



```
import os

if os.path.exists("example.txt"):
    print("File exists")
else:
    print("File does not exist")
```