Python Standard Library - https://docs.python.org/2/library/

Distributed with every copy of Python, the Standard Library contains hundreds of modules that provide tools for interacting with the operating system, interpreter, and Internet—all of them tested and ready to be used to jump-start the development of your applications.

To import a module, we use the `import` command. Check out the full list of built-in modules in the Python standard library here.

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.

Exploring built-in modules

Two very important functions come in handy when exploring modules in Python - the **`dir`** and **`help`** functions.

We can look for which functions are implemented in each module by using the `dir` function:

```
>>> import urllib
>>> dir(urllib)
['ContentTooShortError', 'FancyURLopener', 'MAXFTPCACHE', 'URLopener',
'__all__', '__builtins__', '__doc__', '__file__', '__name__', '__package__',
'__version__', '_ftperrors', '_get_proxies', '_get_proxy_settings',
'_have_ssl', '_hexdig', '_hextochr', '_hostprog', '_is_unicode',
'_localhost', '_noheaders', '_nportprog', '_passwdprog', '_portprog',
'_queryprog', '_safe_map', '_safe_quoters', '_tagprog', '_thishost',
'_typeprog', '_urlopener', '_userprog', '_valueprog', 'addbase',
'addclosehook', 'addinfo', 'addinfourl', 'always_safe', 'basejoin', 'c',
'ftpcache', 'ftperrors', 'ftpwrapper', 'getproxies',
'getproxies_environment', 'getproxies_macosx_sysconf', 'i', 'localhost',
'main', 'noheaders', 'os', 'pathname2url', 'proxy_bypass',
'proxy_bypass_environment', 'proxy_bypass_macosx_sysconf', 'quote',
'quote_plus', 'reporthook', 'socket', 'splitattr', 'splithost', 'splitnport',
'splitpasswd', 'splitport', 'splitquery', 'splittag', 'splittype',
'splituser', 'splitvalue', 'ssl', 'string', 'sys', 'test', 'test1',
'thishost', 'time', 'toBytes', 'unquote', 'unquote_plus', 'unwrap',
'url2pathname', 'urlcleanup', 'urlencode', 'urlopen', 'urlretrieve']
```

****

import math

>>> print ('The value of PI is approximately {}' .format(math.pi))

****

```
Import random
for I in range(5):
        x = random.randint(1,6)
        print x

**********

## create a text file on the c drive where you can find it
## file open and print each line

file = open('c:/test/states.txt')
>>> for line in file:
        state = line.strip()
        print(state)
file.close()

**********

## if file doe not exist it creates a new file but if exists the old data is cleared out and file is reused

file = open('c:/test/new_file.txt', 'w')

file.write("First line in the new file \n")
file.write("and another line.")
file.close()

**********

Make a directory
import os

# Create a directory "test"
os.mkdir("c:/test2")
```