

random

<code>random.randint(a,b)</code>	generate random number N $a \leq N \leq b$
<code>random.shuffle(list)</code>	shuffles elements of a list

sys

<code>sys.argv</code>	argument list
<code>sys.exit()</code>	terminate program

File manipulation

<code>obj=open('path','mode')</code>	opens a file and returns a file object
<code>with open('file',mode) as obj</code>	same as open but closes file automatically
<code>obj.readlines()</code>	read lines of a text file
<code>obj.write('string')</code>	writes string to text file
<code>obj.close()</code>	closes obj file object
<code>obj.shelve.open('filename')</code>	open a persistent dictionary

modes: 'r','w','a','rb','wb'

copy

<code>copy.copy(obj)</code>	makes copy of mutable obj (list or dict)
<code>copy.deepcopy(obj)</code>	makes a copy of objs inside obj recursively

pprint (pretty print)

<code>pprint.pprint(obj)</code>	prints the formatted object on stream
<code>pprint.pformat(obj)</code>	Returns the formatted object as a string

send2trash

<code>send2trash.send2trash('file')</code>	sends file to recycle bin
--	---------------------------

zipfile

<code>obj=zipfile.ZipFile('file.zip')</code>	opens file obj to compress file.zip
<code>obj.write('file.txt',compress_type=zipfile.ZIP_DEFLATED)</code>	
<code>obj.extractall('file.zip')</code>	extracts files from zip file
<code>obj.close()</code>	closes compress and extract file obj
<code>obj.namelist()</code>	return a list of archive members by name
<code>info=obj.getinfo('file.zip')</code>	Return info about the archived file

zipfile (cont)

<code>info.file_size</code>	returns original file size in bytes
<code>info.compress_size</code>	returns the compressed file size

os

<code>os.chdir('path')</code>	change firectory
<code>os.getcwd()</code>	get current working dir
<code>os.listdir('path')</code>	returns list of filename strings in 'path'
<code>os.makedirs('path')</code>	recursive directory creation
<code>os.rmdir('path')</code>	deletes the directory path if empty
<code>os.path.abspath('p-path')</code>	returns string w/absolute path
<code>os.path.basename('-path')</code>	returns the filename of the path
<code>os.path.dirname('p-path')</code>	returns the directory name of pathname path
<code>os.path.exists('path')</code>	returns True if path refers to existing path
<code>os.path.getsize('path')</code>	returns filesize in bytes of the file in path
<code>os.path.isabs('path')</code>	returns True if path is absolute
<code>os.path.isdir('path')</code>	returns True if path is an existing directory
<code>os.path.isfile('path')</code>	returns True if path is an existing file
<code>os.path.join()</code>	join strings into a file path
<code>os.path.relpath('path')</code>	returns relative path string
<code>os.path.split('path')</code>	returns a tuple w/path separated from basename
<code>os.unlink('path')</code>	delete file at path (irreversible)
<code>os.walk('path')</code>	returns list of strings for path,subdirs,files vars

Path format in windows - 'Drive:\\Dir'

pyperclip (copies data to clipboard)

<code>pyperclip.copy(obj)</code>	copies obj to clipboard
<code>pyperclip.paste()</code>	pastes clipboard content



requests

<code>r=requests.get('url')</code>	downloads url contents
<code>r.raise_for_status()</code>	raise exception if download fails
<code>r.status_code()</code>	returns request status code
<code>r.headers['content-type']</code>	returns headers
<code>r.encoding</code>	returns encoding format
<code>r.content</code>	response content in a raw string format
<code>r.text()</code>	decodes server response
<code>r.json()</code>	returns a decoded json obj
<code>r.history</code>	tracks redirection 302 if redirected
<code>r.url</code>	returns requested url
<code>r.iter_content(N)</code>	returns "chunks of N bytes" per iteration
<code>requests.post('url',data={'key':'-value'})</code>	pos request



By **Zoltan**
cheatography.com/zoltan/

Not published yet.
Last updated 16th January, 2018.
Page 2 of 2.

Sponsored by **CrosswordCheats.com**
Learn to solve cryptic crosswords!
<http://crosswordcheats.com>