
reuse Documentation

Release 0.0.2

Yunfeng Wang

Nov 25, 2018

Function APIs:

1	Installation	3
2	Examples	5
3	Indices and tables	11
	Python Module Index	13

reuse is a collection of useful python functions in you work.

Here is a demo about how to use reuse: [demo](#).

CHAPTER 1

Installation

```
pip3 install --user py-reuse
```


CHAPTER 2

Examples

```
>>> import reuse as rs
>>> rs.check_exists('/etc/apt/source.lists')
False
>>> rs.check_exists('/etc/apt')
True
>>> rs.concat_path(['home', 'user', 'project'])
'/home/user/project'
>>> rs.create_dir_if_not_exist('./tmp-dir/test')
>>> ll = [[1,2], [3,4], [6,7]]
>>> rs.flat_list(ll)
[1, 2, 3, 4, 6, 7]
>>> rs.full_name('/etc/apt/sources.list')
'sources.list'
>>> rs.pure_name('/etc/apt/sources.list')
'sources'
>>> rs.parent_dir('/etc/apt/sources.list')
'/etc/apt'
>>> rs.run_cmd('ls -l')
1068
-rw-rw-r-- 1 wang wang 15200 9 4 20:21 00218.jpg
-rw-rw-r-- 1 wang wang 15071 9 4 20:21 00219.jpg
drwxrwxr-x 2 wang wang 36864 11 22 16:49 960
-rw-rw-r-- 1 wang wang 5496 11 25 16:07 A.odf
-rw-rw-r-- 1 wang wang 1380 11 22 16:40 a.txt
drwxrwxr-x 2 wang wang 4096 7 14 09:05 bin
-rw-rw-r-- 1 wang wang 5358 11 25 16:08 B.odf
drwxrwxr-x 3 wang wang 4096 11 10 13:19 cpp-test
drwxrwxr-x 2 wang wang 4096 11 19 11:54 diff_masks
drwxr-xr-x 2 wang wang 40960 11 15 18:44 dp
drwxrwxr-x 6 wang wang 61440 11 19 17:32 dy4
drwxrwxr-x 2 wang wang 69632 11 18 22:48 dy4_1G
drwxr-xr-x 2 wang wang 4096 11 20 13:04 epoch-500
drwxrwxr-x 7 wang wang 4096 11 18 09:22 FlameGraph
drwxrwxr-x 4 wang wang 4096 11 23 23:56 go
```

(continues on next page)

(continued from previous page)

```
-rw-rw-r-- 1 wang wang 14 11 23 00:17 hello.txt
-rw-rw-r-- 1 wang wang 309348 11 20 10:34 img1_kp_50.png
drwxrwxr-x 3 wang wang 4096 7 14 09:05 opt
-rw-rw-r-- 1 wang wang 383100 11 20 19:53 ORB.png
-rw-rw-r-- 1 wang wang 337 11 20 21:02 ORB.py
drwxrwxr-x 2 wang wang 4096 11 19 10:48 outputs
-rw-rw-r-- 1 wang wang 35 11 20 10:56 project.py
drwxrwxr-x 3 wang wang 4096 11 1 22:05 PycharmProjects
drwxrwxr-x 11 wang wang 4096 11 18 09:24 pyflame
drwxrwxr-x 4 wang wang 4096 11 14 12:00 results
drwxrwxr-x 2 wang wang 4096 11 16 14:09 S001C001P008R002A060_rgb_out
drwxrwxr-x 2 wang wang 4096 11 16 14:05 S017C003P020R002A060
drwxrwxr-x 2 wang wang 4096 11 12 21:49 tmp
drwxrwxr-x 3 wang wang 4096 11 25 18:01 tmp-dir
drwxrwxr-x 2 wang wang 4096 10 26 13:49 videos
drwxrwxr-x 12 wang wang 4096 11 20 21:40 workspace
drwxrwxr-x 2 wang wang 20480 11 9 17:22 ws_1_step_8
drwxr-xr-x 5 wang wang 4096 11 1 20:08
drwxr-xr-x 11 wang wang 4096 11 24 13:30
drwxr-xr-x 16 wang wang 4096 11 12 23:29
drwxr-xr-x 7 wang wang 12288 11 25 16:23
drwxr-xr-x 2 wang wang 4096 10 2 09:19
0
```

2.1 reuse

Collections of useful python functions.

reuse.**check_exists** (*path*)

Check if the directory exists.

Parameters *path* (*str*) – path to a directory or a file.

Returns if path exist, return True, else return False

Return type *bool*

Example

```
>>> dir_path = '/path/to/not/exist/dir'
>>> check_dir_exists(dir_path)
False
```

reuse.**concat_path** (*parts*)

Concatenate multiple parts to get a complete path.

Parameters *parts* (*list*) – list contains each part of the path.

Returns concatenated path.

Return type *str*

Examples

```
>>> parts = ['home', 'myname', 'project']
>>> path = concat_path(parts)
>>> print(path)
home/myname/project
>>> parts = ['/home', 'myname', 'project']
>>> path = concat_path(parts)
>>> print(path)
/home/myname/project
```

reuse.**create_dir_if_not_exist** (*dir_path*)

Create the directory if it didn't exist.

Parameters **dir_path** (*str*) – path to directory that to create.

Example

```
>>> dir_path = '/path/to/dir1'
>>> create_dir_if_not_exist(dir_path)
```

reuse.**flat_list** (*ll*)

Flatten a list of list to a list.

Parameters **ll** (*list*) – a list has elements of type list.

Returns one dimension list.

Return type list

Examples

```
>>> lol = [[1, 2], [3, 4]]
>> l = flat_list(lol)
>> print(l)
[1, 2, 3, 4]
```

reuse.**full_name** (*path*)

Get the full name (name + '.' + extension) of a path.

Parameters **path** (*str*) – a str contain a POSIX path.

Returns name of the file or dir in path.

Return type str

Example

```
>>> path = '/path/to/a.txt'
>>> full_name(path)
'a.txt'
>>> path = '/path/to/mydir'
>>> full_name(path)
'mydir'
```

`reuse.list_dir` (*path*, *sort=True*)

List directories in given directory.

Parameters

- **path** (*str*) – path of target directory.
- **sort** (*bool*) – if set to True, return sorted list of dirs.

Returns a list contains subdirectories in the directory.

Return type `list`

Example

```
>>> path = '/path/to/my/dir'
>>> list_dir(path)
['/path/to/my/dir/dir1', '/path/to/my/dir/dir2']
```

`reuse.list_file` (*path*, *ptn=None*, *sort=True*)

List files in given directory.

Parameters

- **path** (*str*) – path of target directory.
- **ptn** (*str*) – pattern to selected certain type of file. Default: None.
- **sort** (*bool*) – if set to True, return sorted list of files.

Returns files in the directory that match the pattern.

Return type `list`

Example

```
>>> path = '/path/to/my/dir'
>>> ptn = "*.py" # only choose *.py file
>>> list_file(path, ptn=ptn, sort=True)
['/path/to/my/dir/1.py', '/path/to/my/dir/2.py']
```

`reuse.parent_dir` (*path*)

Return the parent directory of a path.

Parameters **path** (*str*) – a str contain a POSIX path.

Returns the parent directory of target path.

Return type `str`

Example

```
>>> path = '/path/to/my/a.txt'
>>> parent_dir(path)
'/path/to/my'
```

`reuse.pure_name` (*path*)

Get the name with extension of a path.

Parameters `path` (*str*) – a str contain a POSIX path.

Returns name of the file or dir in path.

Return type `str`

Example

```
>>> path = '/path/to/a.txt'
>>> pure_name(path)
a
```

`reuse.read_from_file` (*file_name*)

Read content in a text file.

Parameters `file_name` (*str*) – name of a text file.

Returns all content in the text file.

Return type `str`

Example

```
>>> content = read_from_file('/path/to/a.txt')
>>> print(content)
print('hello')
```

`reuse.run_assert` (*cond, out*)

Run assert and output out if failed.

Parameters

- **cond** (*func*) – condition to judge True or False.
- **out** (*str*) – output when assert fails.

Example

```
>>> i = 20
>>> run_assert(isinstance(i, str), 'i must be a str')
AssertionError: i must be a str
```

`reuse.run_cmd` (*cmd*)

Run a Bash command in python.

Parameters `cmd` (*str or list*) – the Shell command to execute.

Returns 0 if runs successfully, other value if fails.

Return type `int`

Example

```
>>> cmd = 'ls -l'
>>> run_cmd(cmd)
-rw-rw-r-- 1 user user 4651 Nov 10 20:19 reuse.py
>>> cmd = ['ls', '-l']
>>> run_cmd(cmd)
-rw-rw-r-- 1 user user 4651 Nov 10 20:19 reuse.py
```

reuse.**write_to_file**(*content*, *file_name*)

Write content to a text file.

Parameters

- **content** (*str*) – content wanted to write to file.
- **file_name** (*str*) – name of file to write.

Example

```
>>> content = 'line1   line2   line3'
>>> write_to_file(content, 'a.txt')
```

CHAPTER 3

Indices and tables

- `genindex`
- `modindex`
- `search`

r

reuse, 6

C

check_exists() (in module reuse), 6
concat_path() (in module reuse), 6
create_dir_if_not_exist() (in module reuse), 7

F

flat_list() (in module reuse), 7
full_name() (in module reuse), 7

L

list_dir() (in module reuse), 7
list_file() (in module reuse), 8

P

parent_dir() (in module reuse), 8
pure_name() (in module reuse), 8

R

read_from_file() (in module reuse), 9
reuse (module), 6
run_assert() (in module reuse), 9
run_cmd() (in module reuse), 9

W

write_to_file() (in module reuse), 10