

---

# utilspy Documentation

*Release 0.0.1*

**AI2**

**Nov 14, 2018**



---

## Contents

---

<b>1</b>	<b>Install</b>	<b>3</b>
<b>2</b>	<b>Utilspy Examples</b>	<b>5</b>
<b>3</b>	<b>Examples:</b>	<b>7</b>
3.1	Utilspy Examples . . . . .	7
<b>4</b>	<b>Utilspy Packages:</b>	<b>9</b>
4.1	Utilspy Packages . . . . .	9
<b>5</b>	<b>Indices and tables</b>	<b>13</b>
	<b>Python Module Index</b>	<b>15</b>



Collection of python utils.



# CHAPTER 1

---

## Install

---

```
pip install git+https://github.com/BkrmDahal/utispy.git
```





## CHAPTER 2

---

### Utilspy Examples

---

```
import time

from utilspy import log, files, decorator

# logger funcation
logs = log.Logger()

logs.log("< started >")
time.sleep(2)
logs.log("running..")
time.sleep(1)
logs.log("</>")

# timeout
@decorator.timeout(5)
def add(x, y):
    return x+y

# other useful utils
files = utils.walk_directory('.')
```



---

Examples:

---

## 3.1 Utilspy Examples

### 3.1.1 Install

`pip install git+https://github.com/BkrmDahal/utilspy.git`

### 3.1.2 Examples

```
import time

from utilspy import log, files, decorator

# logger funcation
logs = log.Logger()

logs.log("< started >")
time.sleep(2)
logs.log("running..")
time.sleep(1)
logs.log("</>")

# timeout
@decorator.timeout(5)
def add(x, y):
    return x+y

# other useful utils
files = utils.walk_directory('.')
```



### 4.1 Utilspy Packages

#### 4.1.1 utilspy.utils

`utilspy.utils.clean_str(string)`

Tokenization/string cleaning for all datasets except for SST. Original taken from [https://github.com/yoonkim/CNN\\_sentence/blob/master/process\\_data.py](https://github.com/yoonkim/CNN_sentence/blob/master/process_data.py)

**Args:**

**string:str** string of sentences

**Returns:** Lowercase cleaned string:str

`utilspy.utils.download_file(url, save_dir='tmp', filename=None)`

download and save the file

**Args:**

**url:str** url of file

**save\_dir:str** directory to save

**filename:str** Name of file to save as

`utilspy.utils.download_files(urls, save_dir='tmp')`

download all files from urls

**Args:**

**urls:list** urls of file

**save\_dir:str** directory to save

`utilspy.utils.format_filename(s)`

Take a string and return a valid filename constructed from the string. Uses a whitelist approach: any characters not present in `valid_chars` are removed. Also spaces are replaced with underscores.

**Args:**

**s:str** string

**Returns:** Valid filename : str

`utilspy.utils.logger(logger_name='utilspy.utils', filename='log.log', level=10)`  
logger for logging

**Args:**

**logger\_name:str** name of logger

**filename:str** path and file name

**level:str** logging level

**Return:** logger funcation for logging:logger

`utilspy.utils.page_soup(url)`  
Make BeautifulSoup of page

**Args:**

**url:str** url of page

**Returns:** BeautifulSoup of page:soup

`utilspy.utils.read_config(filename)`  
Parse config yaml file

**Args:**

**filename:str** path of config file

`utilspy.utils.top_n(lis, n)`  
Get index of top n item from list with value

**Args:**

**lis:list** list

**n:int** number of top value

**Return:** dict of top value and index: dict

`utilspy.utils.walk_directory(dir_path, files_only=True)`  
Get list of all file in directory recursive

**Args:**

**dir\_path:str** parent directory to start

**files\_only:bool** get list of files only other

**Return:** If file only than list of all files otherwise files and list of directories

## 4.1.2 utilspy.decorator

`utilspy.decorator.profile(fnc)`  
A decorator that uses cProfile to profile a function And print the result

`utilspy.decorator.threading_d(func)`  
A decorator to run function in background on thread

**Args:**

**func: function** Function with args

**Return:** background\_thread: Thread

`utilspy.decorator.timeout (seconds=10, error_message='Timer expired')`

Decorator to throw timeout error, if function doesnt complete in certain time

**Args:**

**seconds: int** No of seconds to wait

**error\_message: str** Error message

### 4.1.3 utilspy.log

Logger function that print time required in each session of logger

**class** `utilspy.log.Logger`

Bases: object

Print the log message with time required between each session New session should start with “<” and end with “>” session ends with “</>”

**log** (*message*)

log message new session starts with < and ends with > session end with </>

### 4.1.4 utilspy.files

`utilspy.files.delete_dirs (folder_names)`

delete folder and all its contents

**Args:** ‘

**folder\_names: str** or **list** string or list of folders

`utilspy.files.make_dirs (folder_names)`

make folder if it doesnot exist

**Args:**

**folder\_names: str** or **list** string or list of folders

`utilspy.files.move_file (filename, out_dir)`

move file/dir

**Args:**

**filename: str** filename of file to be moved or name of dir to be moved

**out\_dir: str** output directory

`utilspy.files.parse_path (path, fn_only=False, ext=False, al=True)`

get the directory from filename,

**Args:**

**path: str** path of file

**fn\_only: bool** get the file name only from path

**ext: bool** split file name into name and extension

**al: bool** get list of dir and file name

**Returns:** list of value: list

`utilspy.files.unzip_file(filename, save_dir='data')`  
unzip all file

**Args:**

**filename:str** filename of zip file

**sav\_dir:str** parent directory to zip

**Warning:** for rar file type, install rar and unrar

```
apt install rar && apt install unrar
```

`utilspy.files.zip_folder(dir_path, filename=None, exclude=None, include=None)`  
zip all file in folder,

**Args:**

**dir\_path:str** parent directory to zip

**filename:str** filename of zip file

**exclude:list** list of file to exclude

**include:list** list of file to include



## CHAPTER 5

---

### Indices and tables

---

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



### u

- `utilspy.decorator`, [10](#)
- `utilspy.files`, [11](#)
- `utilspy.log`, [11](#)
- `utilspy.utils`, [9](#)



### C

`clean_str()` (in module `utilspy.utils`), 9

### D

`delete_dirs()` (in module `utilspy.files`), 11

`download_file()` (in module `utilspy.utils`), 9

`download_files()` (in module `utilspy.utils`), 9

### F

`format_filename()` (in module `utilspy.utils`), 9

### L

`log()` (`utilspy.log.Logger` method), 11

`Logger` (class in `utilspy.log`), 11

`logger()` (in module `utilspy.utils`), 10

### M

`make_dirs()` (in module `utilspy.files`), 11

`move_file()` (in module `utilspy.files`), 11

### P

`page_soup()` (in module `utilspy.utils`), 10

`parse_path()` (in module `utilspy.files`), 11

`profile()` (in module `utilspy.decorator`), 10

### R

`read_config()` (in module `utilspy.utils`), 10

### T

`threading_d()` (in module `utilspy.decorator`), 10

`timeout()` (in module `utilspy.decorator`), 11

`top_n()` (in module `utilspy.utils`), 10

### U

`unzip_file()` (in module `utilspy.files`), 12

`utilspy.decorator` (module), 10

`utilspy.files` (module), 11

`utilspy.log` (module), 11

`utilspy.utils` (module), 9

### W

`walk_directory()` (in module `utilspy.utils`), 10

### Z

`zip_folder()` (in module `utilspy.files`), 12