

---

**misc<sub>utils</sub>**  
***Release 0.1.0***

**Apr 28, 2022**



---

## Contents:

---

<b>1</b>	<b>Misc_utils</b>	<b>1</b>
<b>2</b>	<b>Decorators</b>	<b>9</b>
<b>3</b>	<b>Changelog</b>	<b>11</b>
<b>4</b>	<b>View all functions</b>	<b>13</b>
<b>5</b>	<b>Indices and tables</b>	<b>15</b>
	<b>Python Module Index</b>	<b>17</b>
	<b>Index</b>	<b>19</b>



# CHAPTER 1

---

## Misc\_utils

---

Misc system & data process utils

### Usage:

```
>>> import misc_utils as utils
>>> utils.func_name()  # to call functions in this file
```

`misc_utils.misc_utils.cmd(shell)`

Run a shell and return results.

Args:

`misc_utils.misc_utils.color_print(text="", color=0, end='\n')`

Print colored text.

### Parameters

- **text** (*str*) – text to print.
- **color** (*int*) –
  - 0 black
  - 1 red
  - 2 green
  - 3 yellow
  - 4 blue
  - 5 cyan (like light red)
  - 6 magenta (like light blue)
  - 7 white
- **end** (*str*) – end string after colored text.

### Example

```
>>> color_print('yellow', 3)
```

`misc_utils.misc_utils.file_lines(filename, prefix="", offset=0, max_num=0)`

Load a text file and parse the content as a list of strings.

#### Parameters

- **filename** (*str*) – Filename.
- **prefix** (*str*) – The prefix to be inserted to the beginning of each item.
- **offset** (*int*) – The offset of lines.
- **max\_num** (*int*) – The maximum number of lines to be read, zeros and negatives mean no limitation.

**Returns** A list of strings.

**Return type** list[str]

`misc_utils.misc_utils.format_num(num: int) → str`

Add comma in every three digits (return a string).

**Parameters** **num** (*int*) – a number.

#### Examples

```
>>> format_num(10000) # 10,000
>>> format_num(123456789) # 123,456,789
```

`misc_utils.misc_utils.format_time(seconds)`

Convert seconds to formatted time string.

**Parameters** **seconds** (*int*) – second number.

#### Examples

```
>>> format_time(10) # 10s
>>> format_time(100) # 1m
>>> format_time(10000) # 2h 47m
>>> format_time(1000000) # 11d 13h 47m
```

`misc_utils.misc_utils.gambling(prob, total=1.0)`

Return True in a given probability :param prob: chance to return True. :type prob: float :param total: total, default 1.0. :type total: float

**Returns** (randomly) True or False.

**Return type** (bool)

`misc_utils.misc_utils.get_dict_value(data, key)`

: 'key1.key2.key3' data[key1][key2][key3]

#### Parameters

- **data** – dict or list
- **key** – str 'key1.key2.key3' format

**Returns** data[key1][key2][key3]

`misc_utils.misc_utils.get_dir_name(path)`

Get parent directory name.

**Args** `path(str)`: file's abs path.

**Returns** `dirname`.

**Example**

```
>>> get_dir_name('root/train/0001.jpg') # mode/train
>>> get_dir_name(get_dir_name('root/train/0001.jpg')) # root
```

`misc_utils.misc_utils.get_file_ext(path)`

**Example**

```
>>> get_file_ext('train/0001.jpg') # .jpg
```

`misc_utils.misc_utils.get_file_name(path)`

Get filename by path (without extension).

**Args** `path(str)`: file's abs path.

**Returns** `filename` (without extension).

**Example**

```
>>> get_file_name('train/0001.jpg') # 0001
```

`misc_utils.misc_utils.get_file_paths_by_pattern(pattern='*', folder=None)`

Get a file path list matched given pattern.

**Parameters**

- **pattern** (`str`) – a pattern to match files.
- **folder** (`str`) – searching folder.

**Returns** (list of `str`): a list of matching paths.

**Examples**

```
>>> get_file_paths_by_pattern('*.png') # get all *.png files in folder
>>> get_file_paths_by_pattern('*rotate*') # get all files with 'rotate' in_
↪ name
```

`misc_utils.misc_utils.get_logger(f='log.txt', mode='w', level='info', print_stream=True)`

Get a logger.

**Parameters**

- **f** (`str`) – log file path.
- **mode** (`str`) – 'w' or 'a'.
- **level** (`str`) – 'debug' or 'info'.
- **print\_stream** (`bool`) – if print to terminal or not.

**Returns** A logger.

**Example**

```
>>> logger = get_logger(level='debug')
>>> logger.info("test")
```

`misc_utils.misc_utils.get_time_stamp(add_offset=0)`

Get time\_zone+0 unix time stamp (seconds)

**Parameters** `add_offset` (*int*) – bias added to time stamp

**Returns** time stamp seconds

**Return type** (*str*)

`misc_utils.misc_utils.get_time_stamp_by_format_str(time_str: str, fmt='%Y/%m/%d %H:%M:%S', timezone=8)`

Get timestamp by formatted time string.

**Parameters**

- `time_str` (*str*) – string in fmt format.
- `fmt` (*str*) – format.
- `timezone` (*int*) – time zone.

**Returns** time stamp

**Return type** (*str*)

### Example

```
>>> get_time_stamp_by_format_str('2020/01/01 15:30:00')
>>> # 1577863800
```

`misc_utils.misc_utils.get_time_str(time_stamp=None, fmt='%Y/%m/%d %H:%M:%S', timezone=8, year_length=4)`

Get formatted time string.

**Parameters**

- `time_stamp` (*str*) – linux time string (seconds).
- `fmt` (*str*) – string format.
- `timezone` (*int*) – time zone.
- `year_length` (*int*) – 2 or 4.

**Returns** formatted time string.

**Return type** (*str*)

### Example

```
>>> get_time_str()
>>> # 2020/01/01 13:30:00
```

`misc_utils.misc_utils.hash(length=8)`

Return a random hash-like string such as *a6b3c47f*. :param length: length of hash :type length: int

**Returns** (randomly) a hash-like string.



**Return type** (bool)

`misc_utils.misc_utils.is_file_image(filename)`

Return if a file's extension is an image's.

**Parameters** `filename` (*str*) – file path.

**Returns** if the file is image or not.

**Return type** (bool)

`misc_utils.misc_utils.is_image_file(filename)`

Return if a file's extension is an image's.

**Parameters** `filename` (*str*) – file path.

**Returns** if the file is image or not.

**Return type** (bool)

`misc_utils.misc_utils.mean(data: list, prec=3)`

Calc mean value of a list.

**Parameters**

- **data** (*list*) – a list.
- **prec** (*int*) – round precision.

**Returns** (float) mean value.

## Example

```
>>> mean([1, 2, 3, 4])
>>> # 2.5
```

`misc_utils.misc_utils.no_need_to_recur(item, list_max_show=5, indents=0)`

: , torch.Tensor([24, 3, 256, 256])

`misc_utils.misc_utils.p(obj)`

Recursively print list, tuple or dict items

**Parameters** `obj` (*list, tuple or dict*) – a list, tuple or dict to print.

`misc_utils.misc_utils.preview(obj, depth=2, dict_max_show=8, text_max_show=20, list_max_show=5, key=None, indents=0)`

Preview large object :param obj: Any type of object, dict, list, set, np.ndarray, torch.Tensor, or anything :param depth: int, :param dict\_max\_show: int, :param text\_max\_show: int, :param list\_max\_show: int, :param key: str, key, key1.key2.key3 :param indents: ,

`misc_utils.misc_utils.print_args(args)`

Print args parsed by argparse.

**Parameters** `args` – args parsed by argparse.

## Example

```
>>> parser = argparse.ArgumentParser()
>>> args = parser.parse_args()
>>> print_args(args)
```

`misc_utils.misc_utils.progress_bar` (*current*, *total*, *pre\_msg=None*, *msg=None*)  
Render a progress\_bar in terminal.

**Preview** Training... Step: [=====>... 26/100 .....] ETA: 0s | loss: 0.45

#### Parameters

- **current** (*int*) – current counter, range in [0, total-1].
- **total** (*int*) – total counts.
- **pre\_msg** (*str*) – message before the progress bar.
- **msg** (*str*) – message after the progress bar.

#### Example

```
>>> for i in range(100):  
>>>     progress_bar(i, 100, 'Training...', 'loss:0.45')
```

`misc_utils.misc_utils.safe_key` (*dic: dict*, *key*, *default=None*)  
Return dict[key] if dict has the key, in case of KeyError.

#### Parameters

- **dic** (*dict*) – a dictionary.
- **key** (*usually str or int*) – key.
- **default** – default return value.

**Returns** dic[key] if key in dic else default.

`misc_utils.misc_utils.save_file_lines` (*filename*, *lines*)  
Load a text file and parse the content as a list of strings.

#### Parameters

- **filename** (*str*) – Filename.
- **prefix** (*str*) – The prefix to be inserted to the begining of each item.
- **offset** (*int*) – The offset of lines.
- **max\_num** (*int*) – The maximum number of lines to be read, zeros and negatives mean no limitation.

**Returns** A list of strings.

**Return type** list[str]

`misc_utils.misc_utils.split_underline` (*str*, *end\_num*, *start\_num=0*, *token='\_'*,  
*keep\_ex=True*)  
split a string by token and return a part of it.

#### Parameters

- **str** (*str*) – string to handle with.
- **end\_num** (*int*) – end of kept parts.
- **start\_num** (*int*) – start of kept parts.
- **token** (*str*) – split by which token.
- **keep\_ex** (*bool*) – whether to keep original extension.

### Example

```
>>> split_underline('abc_123_t134567_cam1.jpg', 2)
>>> # abc_123.jpg
```

`misc_utils.misc_utils.to_string(obj, last_comma=False)`

Convert to string in one line.

#### Parameters

- **obj** (*list, tuple or dict*) – a list, tuple or dict to convert.
- **last\_comma** (*bool*) – add a comma at last.

**Returns** (str) string.

### Example

```
>>> to_string([1, 2, 3, 4], last_comma=True)
>>> # 1, 2, 3, 4,
>>> to_string({'a': 2, 'b': 4})
>>> # a=2, b=4
```

`misc_utils.misc_utils.toggle_list_dict(obj)`

Convert list of dict to dict of list, and vice versa.

**Parameters** **obj** – a list or a dict.

**Returns** converted type of obj.

### Example

```
>>> toggle_list_dict([{'a': 3}, {'a': 5}, {'a': 7}])
>>> # {'a': [3, 5, 7]}
>>> toggle_list_dict({'a': [3, 5, 7]})
>>> # [{'a': 3}, {'a': 5}, {'a': 7}]
>>> k, v = toggle_list_dict({1: 2, 3: 4})
>>> # k=[1, 3], v=[2, 4]
```

`misc_utils.misc_utils.try_make_dir(folder)`

Make a directory when ignoring FileExistsError.

**Parameters** **folder** (*str*) – directory path.



## CHAPTER 2

---

### Decorators

---

Decoration Utilities.

`misc_utils.decorators.deprecated(info="")`  
Decorate a deprecated function.

**Parameters** `info` – info to show.

#### Example

```
>>> from misc_utils import deprecated
>>>
>>> @deprecated('old_func() is deprecated now, use new_func() instead.')
>>> def old_func():
>>>     pass
>>>
>>> old_func()
>>> # DeprecationWarning: old_func() is deprecated now, use new_func() instead.
```

`misc_utils.decorators.timer(show_args=True, logger=None)`  
Decorate a function to log how long the function took to execute.

**Parameters** `logger` – logger to write to, print if None.

#### Example

```
>>> from misc_utils import timer
>>>
>>> @get_timer(logger)
>>> def test(a, **kwargs):
>>>     for i in range(a):
>>>         time.sleep(1)
>>>
```

(continues on next page)

(continued from previous page)

```
>>> test(3, b=2, c=3)
>>> # [INFO] 2020-01-01 15:30:00 Call ttt(3, b=2, c=3), time: 3s.
```

### 3.1 v0.0.6

Added `file_lines()` Modify `get_file_paths_by_pattern()` Added `deprecated()` decorator Added `save_pickle()` Added `load_pickle()` Added `save_json()` Added `load_json()`

### 3.2 v0.0.5

Added `gambling()` Added `hash()` Added `cmd()` Changed `get_timer` to `timer` Added `template`

### 3.3 v0.0.4

Added `toggle_list_dict()` Added `mean()`

### 3.4 v0.0.3

Added `misc_utils.__version__`.

Added `Decorators(get_timer())`.

Added `travis-ci` building test and `codecov` coverage test.

### 3.5 v0.0.2

Update `__init__.py`

Updated docs, function name index now can be found in online docs.

Added `get_time_stamp()`, `get_time_str()`.

## **3.6 v0.0.1**

Added `misc_utils.py`.



## CHAPTER 4

---

View all functions

---

misc\_uils/\_\_init\_\_.py

```
from .version import __version__

#####
#         io
#####
from .misc_utils import p, preview
from .misc_utils import color_print
from .misc_utils import print_args

from .misc_utils import get_logger
from .misc_utils import cmd
bash = cmd

#####
#         math
#####
from .misc_utils import hash
from .misc_utils import gambling
from .misc_utils import mean

#####
#         safe load
#####
from .misc_utils import safe_key

#####
#         str stuff
#####
from .misc_utils import to_string
from .misc_utils import split_underline
```

(continues on next page)

(continued from previous page)

```

#####
#     file system
#####
from .misc_utils import try_make_dir
from .misc_utils import get_file_name, get_file_ext
from .misc_utils import get_dir_name
from .misc_utils import get_file_paths_by_pattern

from .misc_utils import save_file_lines, file_lines
from .misc_utils import save_pickle, load_pickle
from .misc_utils import save_json, load_json

#####
#     time stamp & str
#####
from .misc_utils import get_time_stamp
from .misc_utils import get_time_str
from .misc_utils import get_time_stamp_by_format_str

#####
#         format
#####
from .misc_utils import format_time
from .misc_utils import format_num

#####
#     dict&list helpers
#####
from .misc_utils import toggle_list_dict
from .misc_utils import get_dict_value

#####
#         misc utils
#####
from .misc_utils import progress_bar

from .misc_utils import is_image_file, is_file_image

#####
#         decorators
#####
from .decorators import timer
from .decorators import deprecated

#####
#         classes
#####
from .classes import EasyDict, TypeHandler, ThreadPool

```

## CHAPTER 5

---

### Indices and tables

---

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



### m

`misc_utils.decorators`, 9  
`misc_utils.misc_utils`, 1



## C

`cmd()` (in module *misc\_utils.misc\_utils*), 1  
`color_print()` (in module *misc\_utils.misc\_utils*), 1

## D

`deprecated()` (in module *misc\_utils.decorators*), 9

## F

`file_lines()` (in module *misc\_utils.misc\_utils*), 2  
`format_num()` (in module *misc\_utils.misc\_utils*), 2  
`format_time()` (in module *misc\_utils.misc\_utils*), 2

## G

`gambling()` (in module *misc\_utils.misc\_utils*), 2  
`get_dict_value()` (in module *misc\_utils.misc\_utils*), 2  
`get_dir_name()` (in module *misc\_utils.misc\_utils*), 2  
`get_file_ext()` (in module *misc\_utils.misc\_utils*), 3  
`get_file_name()` (in module *misc\_utils.misc\_utils*), 3  
`get_file_paths_by_pattern()` (in module *misc\_utils.misc\_utils*), 3  
`get_logger()` (in module *misc\_utils.misc\_utils*), 3  
`get_time_stamp()` (in module *misc\_utils.misc\_utils*), 4  
`get_time_stamp_by_format_str()` (in module *misc\_utils.misc\_utils*), 4  
`get_time_str()` (in module *misc\_utils.misc\_utils*), 4

## H

`hash()` (in module *misc\_utils.misc\_utils*), 4

## I

`is_file_image()` (in module *misc\_utils.misc\_utils*), 5  
`is_image_file()` (in module *misc\_utils.misc\_utils*), 5

## M

`mean()` (in module *misc\_utils.misc\_utils*), 5

`misc_utils.decorators` (module), 9  
`misc_utils.misc_utils` (module), 1

## N

`no_need_to_recur()` (in module *misc\_utils.misc\_utils*), 5

## P

`p()` (in module *misc\_utils.misc\_utils*), 5  
`preview()` (in module *misc\_utils.misc\_utils*), 5  
`print_args()` (in module *misc\_utils.misc\_utils*), 5  
`progress_bar()` (in module *misc\_utils.misc\_utils*), 5

## S

`safe_key()` (in module *misc\_utils.misc\_utils*), 6  
`save_file_lines()` (in module *misc\_utils.misc\_utils*), 6  
`split_underline()` (in module *misc\_utils.misc\_utils*), 6

## T

`timer()` (in module *misc\_utils.decorators*), 9  
`to_string()` (in module *misc\_utils.misc\_utils*), 7  
`toggle_list_dict()` (in module *misc\_utils.misc\_utils*), 7  
`try_make_dir()` (in module *misc\_utils.misc\_utils*), 7