dandan Documentation

Release latest

ccyg

Oct 10, 2020

Contents:

CHAPTER 1

show in pypi

CHAPTER 2

show in github

CHAPTER 3

dandan package

dandan package

Several convenient tools for python programming

Events

- 2020-10-10 [0.7.3] remove psutil in dependencies
- 2018-10-13 [0.7.2] fix bug for AttrDict call dict.update
- 2018-10-11 [0.7.1] optimize performance for AttrDict and fix bug
- 2018-10-11 [0.7.0] optimize performance for AttrDict
- 2018-05-21 [0.6.0] add interrupt decorator
- 2018-05-16 [0.5.8] fix bug for AttrDict member functions
- 2018-05-12 [0.5.7] improve code robustness
- 2018-05-11 [0.5.6] modify logger roll suffix is "%Y-%m-%d.log"
- 2018-05-07 [0.5.5] fix bug for md5 and sha1
- 2018-05-03 [0.5.4] fix bug for logger file utf8 encoding
- 2018-04-28 [0.5.3] fix bug for logger file
- 2018-03-23 [0.5.2] add default logger name as 'dandan'
- 2017-12-13 [0.5.1] fix bug
- 2017-12-13 [0.5.0] add system.kill and execute timeout
- 2017-12-12 [0.4.2] fix bug for system.execute in callback mode
- 2017-12-11 [0.4.1] fix bug for system.execute return result with str
- 2017-12-10 [0.4.0] add getLogger method in logger
- 2017-11-29 [0.3.3] fix bug for put_json in python3
- 2017-11-29 [0.3.2] add indent for dandan.value.put_json

- 2017-11-19 [0.3.1] fix bug in dandan.value.length when given string length is zero
- 2017-11-19 [0.3.0] add function dandan.value.length
- 2017-11-17 [0.2.3] update document for project enhance AttrDict class
- 2017-11-15 [0.2.2] update document for project
- 2017-10-14 move to another github project
- 2017-06-25 add function system.clear
- 2017-06-23 add function system.getch
- 2017-06-23 Support python3
- 2017-06-23 Add TestCase

3.1 Submodules

3.2 dandan.value module

class dandan.value.**AttrDict**(*dic={}*, *json_string=None*, **args*, ***kwargs*)

Bases: *dict*

Use dict key as attr

examples

```
data = Attrdict()

# after two lines are equal statement
data.key1 = 1
data["key1"] = 1


# defaut value also is Attrdict after line is allowed
data.key2.key.key.key = 5

# if plus or minus a number the default value is 0
data.key3 += 5
assest(data.key2 == 5)
```

enjoys!!!

DICT_ATTRS = set(['__class__', '__cmp__', '__contains__', '__delattr__', '__delitem__'

dict()

return dict object of current instance

Returns:

- dict: convert from current instance

dandan.value.**get_json**(*filename*)

Load file as pickle object

Args:

- filename (string): local system filename

Returns:

- object: if file is json data else None

dandan.value.**get_pickle**(*filename*)
Load file as pickle object

Args:

- filename (string): local system filename

Returns:

- object: if file is pickled data else None

dandan.value.**is_number**(*number*)
Test parameter is number True or False

Args:

- number (TYPE): any object

Returns:

- bool: True if number is float value or False

dandan.value.**length**(*string*)
Get true size of string or char, char might be 1 or 2, string accumulate all of char

example

```
dandan.value.length("test string") == 11
dandan.value.length("") == 10
dandan.value.length(" test string") == 22
```

Args: string (string or char): requested

Returns: int: size of string or char

dandan.value.**md5**(*data=None*, *filename=None*, *encoding='utf8'*)
get data or file md5 checksum

Args:

- data (string, optional): string
- filename (string, optional): local system filename

Returns:

- string: md5 checksum

dandan.value.**number**(*num*)
convert parameter to float or None

Args:

- num (TYPE): any object

Returns:

- float: num float value or None

dandan.value.**put_json**(*data*, *filename*, *indent=None*)
Save object as json string to filename

Args:

- data (object): any can jsoned object

- filename (string): local system filename
- indent (None, optional): json indent width default None

dandan.value.**put_pickle**(*data*, *filename*)
Save object as pickle to filename

Args:

- data (object): any can pickled object
- filename (string): local system filename

dandan.value.**sha1**(*data=None*, *filename=None*, *encoding='utf8'*)
get data or file sha1 checksum

Args:

- data (string, optional): string
- filename (string, optional): local system filename

Returns:

- string: sha1 checksum

3.3 dandan.query module

dandan.query.**html**(*url*, ***kwargs*)
get html string object by url

User-Agent : default "Mozilla/5.0 (Windows NT 6.1; Win64; x64; rv:52.0) Gecko/20100101 Firefox/52.0" modify User-Agent put to headers in kwargs

Args:

- url (string): requested http url
- kwargs
- timeout : set get timeout default 60
- headers : set http headers have User-Agent
- params : set http parameters
- retry : set retry count default 0
- encoding : page encoding default utf8
- method : set http method default get
- json : if True return json else return string

Returns: string: if request correct else None

dandan.query.**json**(*url*, ***kwargs*)
get json object by url familiar html if http return json or None

Args:

- url (string): requested http url
- **kwargs: same as *html()*

Returns: json: if request correct else None

dandan.query.**local_ip**()
get local ip address for IP V4

Returns:

- string: current machine ip

TODO: ip v6 not implement

dandan.query.**soup**(*url*, ***kwargs*)
get BeautifulSoup object by url

Args:

- url (string): requested http url
- **kwargs: same as *html()*

Returns: BeautifulSoup: if request correct else None

dandan.query.**whois**(*ip*)
Get whois infommation (developing)

Args:

- ip (string): request for whois

Returns:

- AttrDict: if get whois currect else None

3.4 dandan.system module

dandan.system.**clear**()
Clear console screen

dandan.system.**execute**(*command*, *callback=None*, *timeout=0*)
Execute a system command return status and output

Args: command (TYPE): execute command command must not bash command callback (None, optional): callback function per output line

Returns: string: console output if callback is None int: execute exit code

dandan.system.**getch**()
Get a char pressed on keyboard without press Enter

Returns: char(s): return pressed key

dandan.system.**is_linux**()
Check os platform is linux

Returns: bool: True if is linux else False

dandan.system.**is_python2**()
Check current interpreter is python2

Returns: bool: True if is python2 else False

dandan.system.**is_python3**()
Check current interpreter is python3

Returns: bool: True if is python3 else False

dandan.system.**is_win32**()
Check os platform is win32

Returns: bool: True if is win32 else False

dandan.system.**kill**(*pid*)

Kill progress with pid

Args: pid (int): progress id

dandan.system.**kill_command**(*command*)
Kill progress with command

Args: command (string): run command

dandan.system.**readable**(*path*)
Check if a given path is readable by the current user.

Args: path (string): local system path

Returns: bool: True if readable else False

dandan.system.**set_unicode**()
Set default encoding to utf8

dandan.system.**writeable**(*path*, *check_parent=True*)
Check if a given path is writeable by the current user.

Args:

- path (string): local system path
- check_parent (bool, optional): If the path to check does not exist, check for the ability to write to the parent directory instead

Returns:

- bool: True if writeable else False

3.5 dandan.traffic module

dandan.traffic.**download**(*url*, *filename*, *callback=None*, *force=False*, *headers={}*, *check_length=False*)
Download http file to filename

Args:

- url (string): http url for download
- filename (string): local system filename to save file
- callback (function, optional): callback function when file downloading
- force (bool, optional): redownload if filename exists default False
- headers (dict, optional): http headers
- check_length (bool, optional): Check length for downloaded file and http headers content-length

Returns:

- bool: success status

- string: file length if correct else error information
- int : spend time for download period

dandan.traffic.**upload**(*filename*, *url*, *callback=None*, ***kwargs*)
Upload filename to url

Args:

- filename (string): local system filename to upload
- url (string): http url
- callback (function, optional): callback function when file downloading
- kwargs: parameters for http

Returns:

- bool: success status
- string: url return content if correct else error information
- int : spend time for download period

3.6 dandan.logger module

dandan.logger.**getLogger**(*name='dandan'*, *level=10*, *filename=None*, *backup_count=10*)
Get logger for convenient method

Args:

- name (string): logger name, default as 'dandan'
- level (logger level, optional): level of this logger such as DEBUG, INFO, WARNING, ERROR, FATAL
- filename (string, optional): filename for timerotedlogger
- backup_count (int, optional): file backup count, if file count larger than count, then oldest file will be deleted.

Returns:

- logger: the logger named name

CHAPTER 4

Indices and tables

- genindex
- modindex
- search

Python Module Index

d

Index