Nits Documentation

Release 0.19

Scott Howard James

Sep 26, 2021

API

1 Classes **3**
- 1.1 Type Casting . . . 3
- 1.2 File Handling . . . 3
- 1.3 Reporting . . . 4

2 Functions **5**
- 2.1 Time . . . 5

3 Command Line Scripts **7**
- 3.1 Repeating Commands . . . 7
- 3.2 Executing Commands in Folders . . . 7

4 Index **9**

Python Module Index **11**

Index **13**

Nits is a collection of convenience functions and classes, primarily for simplifying python syntax and smoothing some (at least in the opinon of the author) rough edges. Currently, it includes the following:

- Wrappers for basic text and CSV file handling
- Date and time functions for converting to/from UNIX time
- A factory class for general purpose type casting

CHAPTER 1

Classes

1.1 Type Casting

1.2 File Handling

class nits.file.**File**
An abstract class simplifying file access through the use of only two functions:

- read (file)
- write (data, file):

classmethod read(*filename*)
return file elements in a generator

classmethod write(*data*, *filename*)
write data to filename

class nits.file.**Text**
Instantiate the File class for a simple text file

classmethod read(*filename*, *comment=None*, *blanklines=False*, *strip=True*)

- comment: ignore comments
- blanklines: ignore blank lines
- strip: strip write space

classmethod write(*data*, *filename*, *eol='\n'*)
write data to filename

class nits.file.**CSV**
Instantiate the File class for Comma Separated Values (CSV)

classmethod read(*filename*, *header=True*, *comment=None*, *fields=None*)

- header: is first line the header?

- fields: optional list of field values

classmethod write(*data*, *filename=None*, *fields=None*, *header=True*, *append=False*, *delimiter=','*)

- fields: optional list of field values
- header: display header on first line?
- append: add to existing file?
- delimiter: what character to use for separating elements

1.3 Reporting

class nits.reporter.**Reporter**(*name=None*, *verbose=False*)

A simple timestamping logging class

abort(*message*, *target=None*)

report error and exit

say(*message*, *target=None*)

report message

warn(*message*, *target=None*)

report warning

CHAPTER 2

Functions

2.1 Time

Description: Convenience functions and constants to deal with python's eclectic date-time packaging conventions

class nits.time.**Test_Time**(*methodName='runTest'*)
Regression tests for time

nits.time.**date2unix**(*d*)
convert python datetime to UNIX time format

nits.time.**file2time**(*f*)
get last modification time of file

nits.time.**now2str**(*format='%Y%m%d %H:%M:%S'*)
current time in string format

nits.time.**now2time**()
current time in *datetime* format

nits.time.**now2unix**()
current time in UNIX format

nits.time.**str2unix**(*s*, *format='%Y%m%d %H:%M:%S'*)
parse string to UNIX time format

nits.time.**time2file**(*f*, *time*)
stamp modification time on file

nits.time.**unix2date**()
Construct a naive UTC datetime from a POSIX timestamp.

nits.time.**unix2str**(*u*, *format='%Y%m%d %H:%M:%S'*, *zone_offset=0*)
create string from UNIX time forfmat

CHAPTER 3

Command Line Scripts

3.1 Repeating Commands

Description: Run the same command on a bunch of files

Usage: repeatit -c <command> <files>... [-d]

Text replacement options within (-c) command string:

- %f gets replaced with the file name
- %n gets replaced with the file name up to the last extension

Options: -h –help show this screen -c, –command <command> command -f, –files <files> file list -d, –dontwait do not wait for them to complete

Example(s): Print names of text files:

*repeatit -c "echo %f" *.txt*

Unzip files in subfolders in place:

*repeatit -c "cd %f;unzip *.zip" **

3.2 Executing Commands in Folders

Description: Run another program from this folder

Environment variables:

- DOITPATH: assign program folder
- PYTHON: the python executable name (e.g. python3)

Usage: doit [<command>] [<parameter>...]

Example(s): *doit something.py else.py -v –folder stuff*

CHAPTER 4

Index

- genindex

Python Module Index

n

nits.doit, 7
nits.repeatit, 7
nits.time, 5

Index

A

abort() (*nits.reporter.Reporter method*), 4

C

CSV (*class in nits.file*), 3

D

date2unix() (*in module nits.time*), 5

F

File (*class in nits.file*), 3
file2time() (*in module nits.time*), 5

N

nits.doit (*module*), 7
nits.repeatit (*module*), 7
nits.time (*module*), 5
now2str() (*in module nits.time*), 5
now2time() (*in module nits.time*), 5
now2unix() (*in module nits.time*), 5

R

read() (*nits.file.CSV class method*), 3
read() (*nits.file.File class method*), 3
read() (*nits.file.Text class method*), 3
Reporter (*class in nits.reporter*), 4

S

say() (*nits.reporter.Reporter method*), 4
str2unix() (*in module nits.time*), 5

T

Test_Time (*class in nits.time*), 5
Text (*class in nits.file*), 3
time2file() (*in module nits.time*), 5

U

unix2date() (*in module nits.time*), 5
unix2str() (*in module nits.time*), 5

W

warn() (*nits.reporter.Reporter method*), 4
write() (*nits.file.CSV class method*), 4
write() (*nits.file.File class method*), 3
write() (*nits.file.Text class method*), 3