gum Documentation

Release 0

Ilia Kurenkov

June 16, 2014

CONTENTS

1 Gum for Python 2.X 3

2 Indices and tables 5

Python Module Index 7

Index 9

Contents:

CHAPTER

ONE

GUM FOR PYTHON 2.X

test docstring

gum.**add_newlines**(*input_iter*, *newline='\n'*, *item_type=<type 'str'>*)
Takes a iterable, adds a new line character to the end of each of its members and then returns a generator of the newly created items. The idea is to convert some sequence that was created with no concern for spliting it into lines into something that will produce a text file. It is assumed that the only input types will be sequences of lists or strings, because these are the only practically reasonable types to be written to files. It is also assumed that by default the sequence will consist of strings and that the lines will be separated by a Unix newline character. This behavior can be changed by passing different newline and/or itemType arguments.

gum.**create_debug_log**(*base='error'*, *ext='.log'*, *separator='_'*, *app='DEFAULT'*)
wrapper for creating a logger.

Parameters **fileNameBase** – base for the log file name to which date, time,

and the extension are later attached. :type ext: string :param ext: string for an extension :type separator: string :param separator: character used to separate different parts of the filename :type app: string :param app: name for the application that generates the error

gum.**create_row_dicts**(*fields*, *data*, *fill_val='NA'*)
Helper generator function for the write_to_table(). Collecting data is often much more efficient and clear when this data is stored in tuples or lists, not dictionaries. Python's csv DictWriter class requires that it be passed a sequence of dictionaries, however. This function takes a header list of column names as well as some data in the form of a sequence of rows (which can be tuples or lists) and converts every row in the data to a dictionary usable by DictWriter.

gum.**find_something**(*smthng*, *string*, *All=False*)
I'm not sure I should keep this

gum.**gen_file_paths**(*dir_name*, *filter_func=None*)
A function for wrapping all the os.path commands involved in listing files in a directory, then turning file names into file paths by concatenating them with the directory name. This also optionally supports filtering file names using filter_func.

gum.**pickle_data**(*data*, *file_name*, *ext='.picl'*)
wrapper for picling any data. :type data: any :param data: python object to be pickled :type fileName: string :param fileName: specifies the name of the pickled file :type ext: string :param ext: adds and extension to the file name

gum.**read_table**(*file_name*, *function=None*, ***fmtparams*)
Function that simplifies reading it table files of any kind.

gum.**subset_dict**(*src_dict*, *relevants*, *replace=False*, *exclude=False*)
Given some keys and a dictionary returns a dictionary with only specified keys. Assumes the keys are in fact present and will raise an error if this is not the case

gum.**write_to_table** (*file_name*, *data*, *header=None*, ***kwargs*)
Writes data to file specified by filename.

Parameters

- **file_name** (*string*) – name of the file to be created
- **data** (*iterable*) – some iterable of dictionaries each of which

must not contain keys absent in the 'header' argument :type header: list :param header: list of columns to appear in the output :type **kwargs: dict :param **kwargs: parameters to be passed to DictWriter. For instance, restvals specifies what to set empty cells to by default or 'dialect' loads a whole host of parameters associated with a certain csv dialect (eg. "excel").

gum.**write_to_txt** (*file_name*, *data*, *mode='w'*, *AddNewLines=False*, ***kwargs*)
Writes data to a text file.

Parameters

- **fName** (*string*) – name of the file to be created
- **data** (*iterable*) – some iterable of strings or lists of strings (not a string)
- **addNewLines** (*bool*) – determines if it's necessary to add newline chars to

members of list :type kwargs: dict :param kwargs: key word args to be passed to list_to_plain_text, if needed

CHAPTER

TWO

INDICES AND TABLES

- *genindex*
- *modindex*
- *search*

g

gum, 3

A

add_newlines() (in module gum), 3

C

create_debug_log() (in module gum), 3
create_row_dicts() (in module gum), 3

F

find_something() (in module gum), 3

G

gen_file_paths() (in module gum), 3
gum (module), 3

P

pickle_data() (in module gum), 3

R

read_table() (in module gum), 3

S

subset_dict() (in module gum), 3

W

write_to_table() (in module gum), 3
write_to_txt() (in module gum), 4