

| B1A_random.py  |   |
|--|---|
| <pre> # ===== """RANDOM : demo for some functions from the 'random' module""" # ===== __author__ = "Christophe Schlick" __version__ = "1.0" __date__ = "2015-09-01" __usage__ = "" Simply press &lt;ENTER&gt; at each pause""" # ===== from ezCLI import * from random import shuffle, choice, sample, randrange, random, gauss # ----- # Shuffle a list of letters # ----- letters = "ABCDEFGHIJKLMNOPQRSTUVWXYZ"; pause("letters : %s" % letters) test = list(letters); shuffle(test); pause("shuffle : %s" % ''.join(test)) # ----- # Take random elements (with push back) from a list of letters # ----- pause("choice : %s" % ' '.join(choice(letters) for n in range(20))) # ----- # Take random elements (without push back) from a list of letters # ----- pause("sample : %s" % ' '.join(sample(letters, 20))) # ----- # Take random values from an integer range (0,100,5) # ----- pause("randrange : %s" % [randrange(0, 100, 5) for n in range(15)]) # ----- # Histogram of the integer range selection for 100000 random values # ----- histo = [0] * 10 for n in range(100000): val = randrange(0, 10); histo[val] += 1 pause("histogram of randrange(0,10) for 100000 values :\n%s" % histo) # ----- # Take random values with uniform distribution on [0,1) # ----- pause("random : %s" % [random() for n in range(10)]) # ----- # Histogram of the uniform distribution for 100000 random values # ----- histo = [0] * 10 for n in range(100000): val = int(10*random()); histo[val] += 1 pause("histogram of random() for 100000 values :\n%s" % histo) # ----- # Take random values with a gaussian distribution on R # ----- pause("gauss(0,1) : %s" % [gauss(0,1) for n in range(10)]) # ----- # Histogram of the gaussian distribution for 1000000 random values # ----- def clamp(val,a,b): return a if val &lt; a else b if val &gt; b else int(val) histo = [0] * 11 for n in range(100000): val = clamp(gauss(0,1),-5,5); histo[val+5] += 1 pause("histogram of gauss(0,1) for 100000 values :\n%s" % histo) # ----- # Take random values with a gaussian distribution of mean=5 and variance=0.01 # ----- pause("gauss(5,0.01) : %s" % [gauss(5,.01) for n in range(10)]) </pre> | <pre> # ----- # Histogram of the gaussian distribution for 1000000 random values # ----- def clamp(val,a,b): return a if val &lt; a else b if val &gt; b else int(val) histo = [0] * 10 for n in range(100000): val = clamp(gauss(5,0.01),0,10); histo[val] += 1 pause("histogram gauss(5,0.01) for 100000 values :\n%s" % histo) # ===== B2A_txtfile.py # ===== """TXTFILE : demo for 'read_txt/write_txt' functions from the 'ezCLI' toolbox""" # ===== __author__ = "Christophe Schlick" __version__ = "1.0" __date__ = "2015-07-01" __usage__ = "" Simply press &lt;ENTER&gt; at each pause""" # ===== from ezCLI import * # ----- # Sample use cases for 'read_txt' # ----- # read the whole content from file txt = read_txt('test-txt.txt') pause("&gt;&gt;&gt; read all lines :\n%s" % txt)  # return line at index 4 from file txt = read_txt('test-txt.txt', 4) pause("&gt;&gt;&gt; read line at index 4 :\n%s" % txt)  # return lines in range(4,7) from file txt = read_txt('test-txt.txt', 4, 7) pause("&gt;&gt;&gt; read lines in range(4,7) :\n%s" % txt)  # return the first 3 lines from file txt = read_txt('test-txt.txt', None, 3) pause("&gt;&gt;&gt; read the first 3 lines :\n%s" % txt)  # return the last 5 lines from file txt = read_txt('test-txt.txt', -5, None) pause("&gt;&gt;&gt; read the last 5 lines :\n%s" % txt)  # ----- # Sample use cases for 'write_txt' # ----- # replace the whole file and return the new file content txt = write_txt('test.txt', 'ccc\nccc\nccc') pause("&gt;&gt;&gt; replace whole content :\n%s" % txt)  # insert line at index 2 and return new file content txt = write_txt('test.txt', 'ddd', 2) pause("&gt;&gt;&gt; insert line at index 2 :\n%s" % txt)  # insert line at end of file and return new file content txt = write_txt('test.txt', 'eee', -1) pause("&gt;&gt;&gt; insert line at end of file :\n%s" % txt)  # replace first line and return new file content txt = write_txt('test.txt', 'aaa\nbbb', None, 1) pause("&gt;&gt;&gt; replace the first line :\n%s" % txt)  # replace the last 2 lines and return new file content </pre> |

```

txt = write_txt('test.txt', 'eee\nfff\nggg', -2, None)
pause(">>> replace the last 2 lines :\n%s" % txt)
# =====
B3A_replace.py
# =====
"""REPLACE : replace all occurrences of a string in a given text file"""
# =====
__author__ = "Christophe Schlick"
__version__ = "1.0" # process a single file
__date__ = "2015-09-01"
__usage__ = """
User input : <oldstring> <newstring> <filename>
App output : modify 'filename' by replacing each 'oldstring' by 'newstring'
Note: use quotes if 'oldstring' or 'newstring' contains space characters"""
# =====
from ezCLI import *
# -----
def replace(oldstring, newstring, name):
    """replace each occurrence of 'oldstr' by 'newstr' in file 'name'"""
    #text = read_txt(name) # read content of file (use default encoding = 'utf8')
    text = read_txt(name, encoding='utf8') # (use 'utf8' encoding)
    #text = read_txt(name, encoding='latin1') # (use 'latin1' encoding)
    #text = read_txt(name, encoding='utf16') # (use 'utf16' encoding)
    count = text.count(oldstring); inspect()
    text = text.replace(oldstring, newstring); #inspect()
    write_txt(name.replace('.', '+'), text) # write modified text in new file
    return "%s : %s strings have been replaced" % (name, count)
# -----
def parser(command):
    """extract arguments from 'command' before calling 'replace()'"""
    oldstring, newstring, name = parse(command); #inspect()
    return replace(oldstring, newstring, name)
# =====
if __name__ == '__main__':
    userloop(parser, "Enter <oldstr> <newstr> <name>")
# =====
B4A_csvfile.py
# =====
"""CSVFILE : demo for 'read_csv/write_csv' functions from the 'ezCLI' toolbox"""
# =====
__author__ = "Christophe Schlick"
__version__ = "1.0"
__date__ = "2015-07-01"
__usage__ = """
Simply press <ENTER> at each pause"""
# =====
from ezCLI import *
# -----
# Sample use cases for 'read_csv'
# -----
# first read the file as a text file to show its raw content
txt = read_txt('test-csv.txt')
pause(">>> read CSV file as a TXT file :\n%s" % txt)

# return the matrix stored in file and apply 'convert' to all cells
csv = read_csv('test-csv.txt')
pause(">>> read CSV file as a matrix :\n%s" % csv)

# return the matrix stored in file but keep all cells as strings
csv = read_csv('test-csv.txt', raw=True)
pause(">>> read CSV file as a matrix of strings :\n%s" % csv)

```

```

# -----
# Sample use cases for 'write_csv'
# -----
# create a sample non-rectangular 3D matrix containing 16 different cells
mat = [[list('abcde'), 'aa,bb,cc', ('dd','ee')], [1,(2,3),[4,5,6]]]
pause(">>> create a sample 3D matrix :\n%s" % mat)

# replace the whole file and return the new file content
csv = write_csv('test.txt', mat)
pause(">>> write matrix as a CSV file:\n%s" % csv)

# insert new block at head and return new file content
csv = write_csv('test.txt', '1,2,3\n4,5,6', 0)
pause(">>> insert new block at head:\n%s" % csv)

# insert new block at tail and return new file content
csv = write_csv('test.txt', [7,8,9], -1)
pause(">>> insert new block at tail:\n%s" % csv)

# replace the first two blocks and return new file content
csv = write_csv('test.txt', mat[1], None, 2)
pause(">>> replace the first two blocks:\n%s" % csv)

# replace the last two blocks and return new file content
csv = write_csv('test.txt', mat[0], -2, None)
pause(">>> replace the last two blocks:\n%s" % csv)
# =====
B5A_inifile.py
# =====
"""INIFILE : demo for 'read_ini/write_ini' functions from the 'ezCLI' toolbox"""
# =====
__author__ = "Christophe Schlick"
__version__ = "1.0"
__date__ = "2015-07-01"
__usage__ = """
Simply press <ENTER> at each pause"""
# =====
from ezCLI import *
# -----
# Sample use cases for 'read_ini'
# -----
# first read the file as a text file to show its raw content
txt = read_txt('test-ini.txt')
pause(">>> read INI file as a TXT file :\n%s" % txt)

# return the data stored in file and apply 'convert' to all items
ini = read_ini('test-ini.txt')
pause(">>> read INI file and convert all data type :\n%s" % ini)

# return the data stored in file but keep all items as strings
ini = read_ini('test-ini.txt', raw=True)
pause(">>> read INI file as keep all data as strings :\n%s" % ini)

# -----
# Sample use cases for 'write_ini'
# -----
# sample data containing 5 properties spread in 3 sections
items = {'': {'a':1, 'b':2}, 'AAA': {'aa':1.2, 'bb':''}, 'BBB': {'cc':'#'}}
pause(">>> sample structured data : \n%s" % items)

# replace the whole file and return the new file content

```

```
ini = write_ini('test.txt', items)
pause(">>> write structured data as an INI file : \n%s" % ini)

# insert new section at tail and return new file content
ini = write_ini('test.txt', '\n[CCC]\nzz = 0', -1)
pause(">>> insert new section at tail : \n%s" % ini)

# replace property stored at line 4 and return new file content
ini = write_ini('test.txt', {'aa':2.5}, 4, 5)
pause(">>> replace property stored at line 4 : \n%s" % ini)
# =====
```