



ファイルの読み込み

23/05/13

	メソッド	スクリプト1	スクリプト2(1のつづき)	出力結果
テキスト	read()	ファイル全体を「1つの文字列」として読み込む (ひとかたまりの文字列str) with open(text, 'r') as f: s = f.read() 小さいファイルならば、read()やreadlines()で一括読み込みするのが効率的	s 1つの文字列が1つの行にまとめて表示される ex. 文字列全体から特定の文字列を～したい場合 s.strip('Hello') ←lstrip, rstrip s.replace('Hello', 'Morning') print(s) printはデフォルトで末尾に改行*を含む print(s, end='') print(s) end=''で最末尾の改行なし。続くファイルの結合で有効	# Hello,katsu!\nFine?\nI'm fine, and you? # Hello,katsu! # Fine? # I'm fine, and you?*
		// with open(text, 'r') as f: for s in f.read():	print(s) f.read()はファイル全体を1つの文字列で読み込むため forだと文字が1文字ずつ出力される	# H # e # 1...1文字ずつ縦に表示される
	readlines()	ファイル全体を「1つのリスト」で読み込む(len: 1) with open(text, 'r') as f: l = f.readlines()	1 改行で区切られた要素がまとめて1つのリストに存在 print(l) 改行を除外したい(改行を含んでいる場合) new_l = list(map(lambda s:s.rstrip('\n'), l)) 以下でもよい new_l = [s.strip() for s in l] (引数なしのstripは文字列の先頭末尾から空白文字(スペース、タブ、改行等)を削除する(rstripだと末尾から削除))	# ['Hello,katsu!\n', 'Fine?\n', 'I'm fine, and you?\n'] # ['Hello,katsu!\n', 'Fine?\n', 'I'm fine, and you?\n'] # ['Hello,katsu!', 'Fine?', 'I'm fine, and you?']
			以下でもよい new_l = [s.strip() for s in l] (引数なしのstripは文字列の先頭末尾から空白文字(スペース、タブ、改行等)を削除する(rstripだと末尾から削除))	
			改行を付与したい(改行を含まない場合) new_l = list(map(lambda s:s + '\n', l)) or [s + '\n' for s in l]	# ['Hello,katsu!\n', 'Fine?\n', 'I'm fine, and you?\n']
			for s in l: print(l) printの改行が間に入る(print自体が改行する かつ 文字列末尾にも改行があるため)	# Hello,katsu! # Fine? # I'm fine, and you?
			for s in l: print(s, end='') (リスト内の文字列の末尾の改行を削除)	# Hello,katsu! # Fine? # I'm fine, and you?
	readline()	ファイルを「1行だけ」読み込む ファイルが大きい場合、一括読み込みするとメモリを圧迫 readline()で1行ずつ読み込み with open(text, 'r') as f: s = f.readline() readlinesと異なり、リストではない	print(type(s)) # <class 'str'> with open(text, 'r') as f: s = f.readline() print(s) 上記を何回やっても「Hello,World!」しか表示されない 例) seekを使えば、5文字目からの1行だけを表示 f.seek(5) s = f.readline() print(s)	# Hello,katsu! # # ,World!
		readline()はwhileとセットで利用してもよい f = open(text, 'r') while True: s = f.readline() if s == '': ←行が無くなれば終わり break s = s.rstrip('\n') print(s) f.close()	print(s) rstripを未使用の場合は下行と同様 最後(EOF)には空文字''が返される	# Hello,katsu! # Fine? # I'm fine, and you? rstripを未使用の場合は下行と同様 最後(EOF)には空文字''が返される
	イテレート	ファイルを1行ずつ(文字列で)読み込む (readline(s)メソッドを利用せずとも、 ファイルオブジェクト をfor文で直接イテレートすれば1行ずつ読み込み可能) with open(text, 'r') as f: ~(右記参照)	for s in f: print(s) 改行を除外(改行を含んでいる場合) for s in f: s = s.rstrip('\n') print(s) 改行なしリストを作成 new_list = [] for s in f: s = s.rstrip('\n') new_list.append(s) new_list 内包表記だと new_list = [s.rstrip('\n') for s in f]	# Hello,katsu! # Fine? # I'm fine, and you? # (ここは空行) # Hello,katsu! # Fine? # I'm fine, and you? # Good! # ['Hello,katsu!', 'Fine?', 'I'm fine, and you?']
CSV		CSVファイルを読み込む ファイルを1行ずつリスト、要素はカンマ区切りで読み込む csv_fp = r'~/../.csv' import csv with open(csv_fp) as f: csv_reader_obj = csv.reader(f)	csv_reader_obj for row in csv_reader_obj: print(row) for row in csv_reader_obj: for v in row: print(v) csv_list = list(csv_reader_obj) csv_list	<_csv.reader at 0x7*****> # ['Hello', 'World!'] 文字列にカンマを含むと変な文章で切られる # ['Fine?'] # ["I'm fine", ' and you'] ←同様 これは元のファイルの各文字列をwクォート等でquoteがない場合に起こる # Hello これも同様 # World! # Fine? # I'm fine # and you [['Hello', 'World!'], ['Fine?'], ['I'm fine', ' and you']]