

 <b>ファイルの読み込み</b>				
				23/05/13
<b>テキスト</b>	メソッド	スクリプト1	スクリプト2(1つのづき)	出力結果
	read()	<p>ファイル全体を「<b>1つの文字列</b>」として読み込む (ひとたまりの文字列str)</p> <pre>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</pre>	<p>s</p> <p>1つの文字列が1つの行にまとめて表示される</p> <p>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')</p> <pre>print(s)  printはデフォルトで末尾に改行*を含む</pre>	# Hello,katsu!\nFine?\nI'm fine, and you?
			<pre>print(s, end='') print(s)</pre> <p>end=''で最末尾の改行なし。続くファイルの結合で有効</p>	# Hello,katsu! # Fine? # I'm fine, and you?~
		<pre>"" with open(text, 'r') as f:     for s in f.read()</pre>	<pre>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</pre>	# H # e # l...1文字ずつ縦に表示される
	readlines()	<p>ファイル全体を「<b>1つのリスト</b>」で読み込む(len:1)</p> <pre>with open(text, 'r') as f:     l = f.readlines()</pre>	<p>l</p> <p>改行で区切られた要素がまとめて1つのリストに存在</p> <pre>print(l)</pre> <p>改行を除外したい(改行を含んでいる場合)</p> <pre>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rstrip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)</pre>	[ 'Hello,katsu!\n', 'Fine?\n', "I'm fine, and you?\n"]
			<p>改行を付与したい(改行を含まない場合)</p> <pre>new_l = list(map(lambda s:s + '\n', l)) or [s + '\n' for s in l]</pre>	[ 'Hello,katsu!\n', 'Fine?\n', "I'm fine, and you?\n"]
			<p>for s in l:     print(s)</p> <p>printの改行が間にに入る(print 자체が改行する かつ 文字列末尾にも改行があるため)</p> <pre>for s in l:     print(s, end='') #リスト内の文字列の末尾の改行を削除</pre>	# Hello,katsu! # Fine? # I'm fine, and you?
	readline()	<p>ファイルを「<b>1行'だけ'</b>読み込む</p> <p>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</p> <pre>with open(text, 'r') as f:     s = f.readline()</pre> <p>readlines()と異なり、リストではない</p>	<pre>print(type(s)) # &lt;class 'str'&gt;  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)</pre>	# Hello,katsu!
		<pre>f = open(text, 'r') while True:     s = f.readline()     if s == '':         break     s = s.rstrip('\n')     print(s) f.close()</pre>	<pre>print(s)  rstrip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</pre>	# Hello,katsu! # Fine? # I'm fine, and you?
<b>イテレート</b>	ファイルを「 <b>1行ずつ(文字列)</b> 」読み込む	<p>(readline()メソッドを利用せずとも、<b>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</b>)</p> <pre>with open(text, 'r') as f:     ~(右記参照)</pre>	<p>for s in f:</p> <pre>print(s)</pre>	# Hello,katsu! # Fine? # I'm fine, and you?
			<p>改行を除外(改行を含んでいる場合)</p> <pre>for s in f:     s = s.rstrip('\n')     print(s)</pre>	# Hello,katsu! # Fine? # I'm fine, and you? # Good!
			<p>改行なし「リスト」を作成</p> <pre>new_list = [] for s in f:     s = s.rstrip('\n')     new_list.append(s)  new_list</pre> <p>内包表記だと</p> <pre>new_list = [s.rstrip('\n') for s in f]</pre>	[ 'Hello,katsu!', 'Fine?', "I'm fine, and you?" ]
<b>CSV</b>	CSVファイルを読み込む ファイルを「 <b>1行ずつリスト、要素はカンマ区切り</b> で読み込む	<pre>csv_fp = r'~/~,csv' import csv with open(csv_fp) as f:     csv_reader_obj = csv.reader(f)</pre>	<p>csv_reader_obj</p>	<_csv.reader at 0x7*****>
			<pre>for row in csv_reader_obj:     print(row)</pre>	# ['Hello', 'World!'] 文字列にカンマを含むと複数の文字で切られる # ['Fine?'] # ['I'm fine', ' and you'] ←同様
			<pre>for row in csv_reader_obj:     for v in row:         print(v)</pre>	# Hello これも同様 # World! # Fine? # I'm fine # and you
		<pre>csv_list = list(csv_reader_obj) csv_list</pre>		[["Hello", "World!"], ["Fine?"], ["I'm fine", ' and you']]