

残業時間編・オマケ



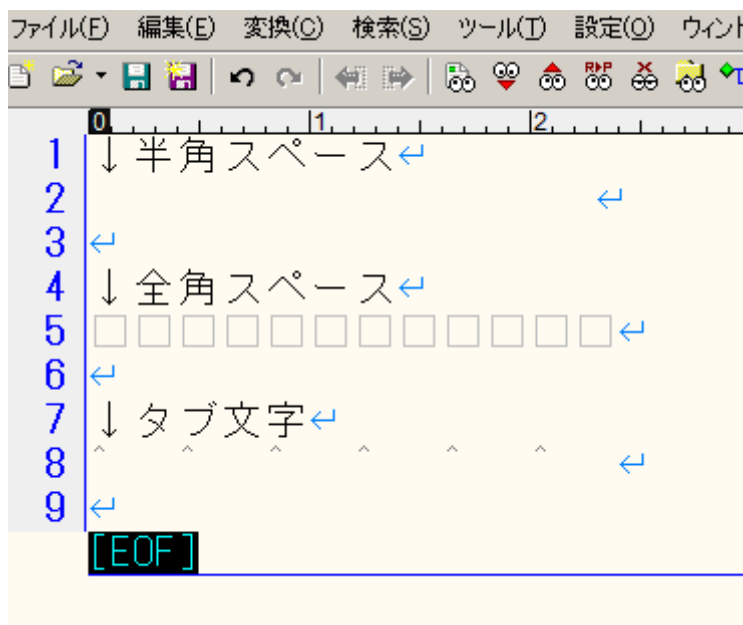
ここまでで、コンマ区切りテキスト(CSV)をpythonで扱うときの方法について、だいたいの見当がつくようになってくれたと思います。

ExcelなんかでもCSVファイルを扱ったことがあるなあ、と思い出したりする人はいますか。ダブルクリックすると普通にExcelの表形式で開けるからあまり深く考えなかったというかたもいらっしゃるでしょうが、あの拡張子が.csvで終わっているファイルは、実態はエディタで開けるテキストファイルなのです。

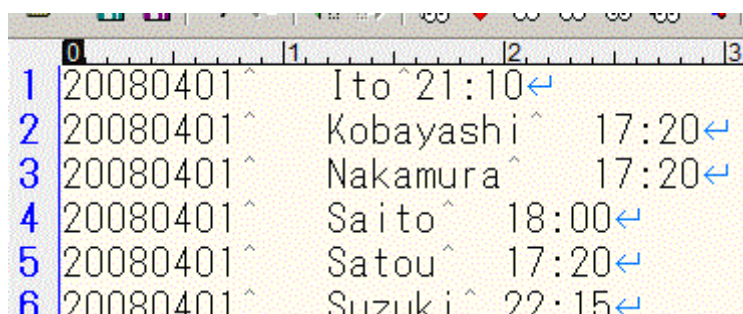
(Excel標準の.xlsとか.xlsxとかいう拡張子がついたやつは、テキストエディタで開いても、グチャグチャな表示にしかありませんよ。こいつは文字の集まりとして読まれることを想定しないものです。バイナリファイルと呼ぶ人もいますね。)

これで、CSV形式のファイルなら、手元にExcelがなくても、pythonがあればなんとか扱えるというわけですね。

もうひとつ、「タブ区切りテキスト」についても一言触れておこうと思います。タブって、[TAB]キーを押したときに、カーソルが横にピョコッと幅とびするアレです。幅は広いですが、あのタブも空白や改行と同じように一文字です。タブ記号の見えるテキストエディタなら、下のように見えるでしょう。「^」に似た記号で代用表示されるものひとつがタブ記号一文字です。



コンマ記号じゃなくてタブ記号で区切られたテキストは、下のようになります。



タブ記号って、「最寄りのキリのいい場所に現在位置を動かす」という性質がありますので、ちょっと見た目には値の並びがズレてしまっているように見えますが、これはこれで大丈夫です。

タブ記号をもとにsplitをするときは、コンマ区切りのときの `line.split(",")` に対応して、`line.split("\t")` と書きます。改行記号のときに `"\n"` と書くのと似ていて、これもエスケープ表記を使って目で理解しやすくしたわけです。まあ、いつかこれを使った練習問題も作りますのでよろしく。

ですが今説明しておきたいのは、タブ区切り文字と、Excelの相性です。Excelがなければ、OpenOfficeでもいいですよ。筆者も自宅にはExcel持ってません。有料だし。

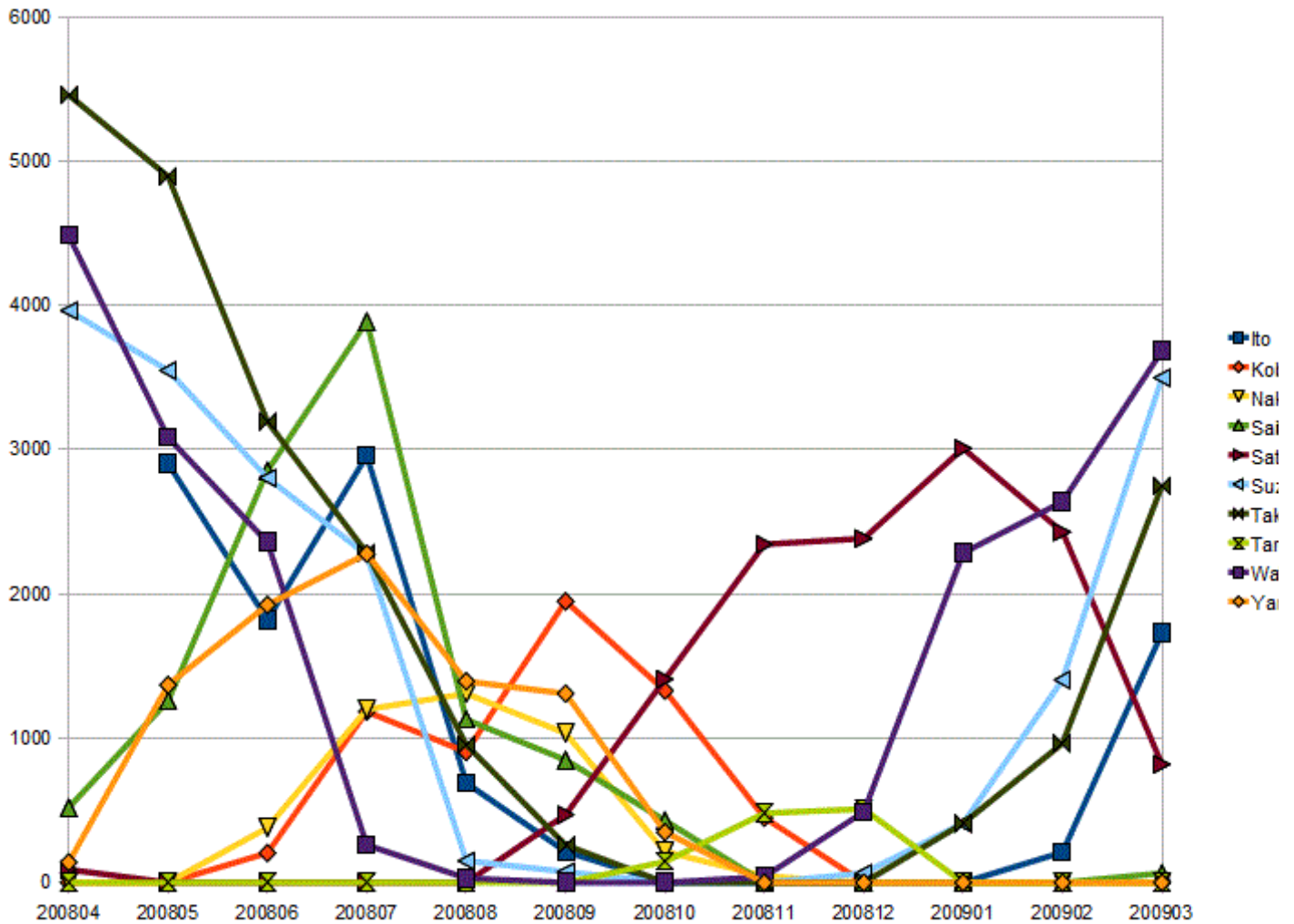
さっきまでの残業時間計算で、下のようなレポートファイルができあがったとします。結果は、今回はコンマや空白じゃなくてすべてタブ記号で区切りました。（まだ、表示結果をファイルに書き出すことについては説明していませんが、すぐできるようになります）

lesson3_report_sample.txt

こいつをエディタで開いて「すべて選択 → コピー」し、ExcelやOpenOffice Calcの新規シートの上で「貼り付け」してみましよう。（Webブラウザ上の表示を直接コピーしてもうまくいかないけないときがありますので注意） どうでしょう。ちゃんとセルごとに値が分かれて入力されましたか。

こうすると何がうれしいかというと、Excelなどでやったほうがやりやすい処理（並べ替えとか、グラフ作成）を、pythonでこしらえた途中経過から引き継ぐことができるのです。

今のデータを手っ取り早く「年月、名前」で並び替えてピボット表をつくれれば、ある年度の残業時間一覧のグラフなんかを作ってしまうことも簡単です。（Excelのテキストではないので説明は省きますが）



本質的な処理をpythonなんかのスクリプト言語でこなしてしまって、途中からビジュアルな表現のためにExcelを援用するってかっこいいでしょ。そうでもない？

コピペの方向を逆にすることもできます。Excel等で計算した値を範囲コピーしてテキストエディタの上にペーストすると、タブ区切りデータができあがります。こいつを保存して、あとでpythonでこれを分析するというのもアリですね。

こういうテクニックが決まって仕事が速くこなせると、ちょっといいものですよ。

ところで、タブ記号の存在価値(?)に疑問とか持ったことはないでしょうか。「コンマ記号で区切り文字は足りるよ。わざわざタブなんていう変な記号を持ってこないでほしいなあ」とかいった具合に。確かにその通りではあるんですが、値自体にコンマを含むデータって、たまにあるんですよ。[YAMAMOTO, Tetsuya] っていう文字列でひとつのデータだったりしたら、ちょっと困るでしょ。CSVではちょっと変わった処理でこの問題を乗り越えなくちゃいけないんですが、タブ記号なら普通はデータそのものに混じらなくて済みますからね。使う場面によってはありがたかったりするのです。