

本当に便利なマクロの入門の話

怖くない マクロ入門

2014/11/8
deris0126

about me

- ❖ Twitter: @deris0126
- ❖ hatena: deris
- ❖ github: deris



目次

- ❖ マクロとは
- ❖ マクロの使い方
- ❖ マクロ実例
- ❖ マクロを使いこなすために

目次

- ❖ マクロとは
- ❖ マクロの使い方
- ❖ マクロ実例
- ❖ マクロを使いこなすために

マクロとは

- ❖ Vimでの一連の操作を記録し、記録した操作を繰り返し再生できる機能
- ❖ マクロにより複雑な操作を繰り返すことで、Vimでの操作効率を向上できる
- ❖ マクロを使えば手作業で半日かかる作業を1, 2分で終わらせられるケースもある

目次

- ❖ マクロとは
- ❖ マクロの使い方
- ❖ マクロ実例
- ❖ マクロを使いこなすために

マクロの使い方

- ❖ マクロの記録
- ❖ マクロの再生

マクロの使い方

- ❖ マクロの記録
 - ❖ $q\ x\ \{\text{一連の操作}\}\ q$
 - ❖ q : マクロの記録開始
 - ❖ x : マクロを記録するレジスタ (0-9, a-zを1文字指定可)
 - ❖ $\{\text{一連の操作}\}$: 記録したい一連の操作
 - ❖ q : マクロの記録終了

編集・移動を
組み合わせる
ことが一般的

マクロの使い方

- ❖ マクロの再生
 - ❖ **[count] @ x**
 - ❖ **[count]** : 再生したい回数を指定
 - ❖ **@** : レジスタxに記録されている操作を再生
 - ❖ **x** : マクロを記録したレジスタ

目次

- ❖ マクロとは
- ❖ マクロの使い方
- ❖ マクロ実例(3つご紹介)
- ❖ マクロを使いこなすために

マクロ実例(1. 連番入力)

- ❖ ゴール

- ❖ 右のコードがゴール

- ❖ 戦略

- ❖ 起点の行を作った上で、コピー&数値インクリメントという操作を記録し再生します

```
1. hoge
2. hoge
3. hoge
～(中略)～
9999. hoge
10000. hoge
```

マクロ実例(1. 連番入力)

- ❖ おもむろに1. hogeを入力する



1. hoge

マクロ実例(1. 連番入力)

- ❖ おもむろに1. hogeを入力する
- ❖ qqでマクロ記録開始

レジスタには任意の文字が指定可能だが、使うキーはあらかじめ決めておくが良い。おすすめはq, w, eなど。

```
1. hoge
```

マクロ実例(1. 連番入力)

- ❖ おもむろに1. hogeを入力する
- ❖ qqでマクロ記録開始
- ❖ yypで1行コピー

```
1. hoge  
1. hoge
```


マクロ実例(1. 連番入力)

- ❖ おもむろに1. hogeを入力する
- ❖ qqでマクロ記録開始
- ❖ yypで1行コピー
- ❖ <C-a>で数値+1

```
1. hoge
2. hoge
```

マクロ実例(1. 連番入力)

- ❖ おもむろに1. hogeを入力する
- ❖ qqでマクロ記録開始
- ❖ yypで1行コピー
- ❖ <C-a>で数値+1
- ❖ qでマクロ記録停止

```
1. hoge
2. hoge
```

マクロ実例(1. 連番入力)

- ❖ おもむろに1. hogeを入力する
- ❖ qqでマクロ記録開始
- ❖ yypで1行コピー
- ❖ <C-a>で数値+1
- ❖ qでマクロ記録停止
- ❖ @qでマクロ再生(3. hoge挿入)

```
1. hoge  
2. hoge  
3. hoge
```


マクロ実例(1. 連番入力)

- ❖ おもむろに1. hogeを入力する
- ❖ qqでマクロ記録開始
- ❖ yypで1行コピー
- ❖ <C-a>で数値+1
- ❖ qでマクロ記録停止
- ❖ @qでマクロ再生(3. hoge挿入)
- ❖ 9997@qでマクロを9997回再生

```
1. hoge
2. hoge
3. hoge
～(中略)～
9999. hoge
10000. hoge
```

マクロ実例(2. CSV編集)

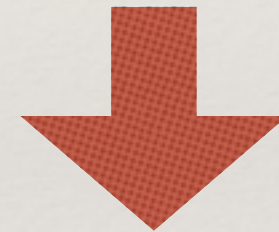
❖ ゴール

❖ 右上のCSVを右下のCSVに変換(2列目と3列目の入れ替え)

❖ 戦略

❖ 2列目をカット、3列目にペースト、次の行へ移動、という操作を記録し再生します

```
hoge1, fuga1, piyo1
hoge2, fuga2, piyo2
hoge3, fuga3, piyo3
~~( 中略 )~~
hoge1000, fuga1000, piyo1000
```



```
hoge1, piyo1, fuga1
hoge2, piyo2, fuga2
hoge3, piyo3, fuga3
~~( 中略 )~~
hoge1000, piyo1000, fuga1000
```

マクロ実例(2. CSV編集)

❖ qqでマクロ記録開始

```
hoge1, fuga1, piyo1  
hoge2, fuga2, piyo2  
hoge3, fuga3, piyo3  
~~( 中略 )~~  
hoge1000, fuga1000, piyo1000
```


マクロ実例(2. CSV編集)

- ❖ qqでマクロ記録開始
- ❖ f,で1つめの,まで移動

```
hoge1, fuga1, piyo1  
hoge2, fuga2, piyo2  
hoge3, fuga3, piyo3  
~~( 中略 )~~  
hoge1000, fuga1000, piyo1000
```

マクロ実例(2. CSV編集)

- ❖ qqでマクロ記録開始
- ❖ f,で1つめの,まで移動
- ❖ dt,で, fuga1をカット

```
hoge1, piyo1  
hoge2, fuga2, piyo2  
hoge3, fuga3, piyo3  
~~( 中略 )~~  
hoge1000, fuga1000, piyo1000
```

マクロ実例(2. CSV編集)

- ❖ qqでマクロ記録開始
- ❖ f,で1つめの,まで移動
- ❖ dt,で, fuga1をカット
- ❖ \$で終端pで, fuga1をペースト

```
hoge1, piyo1, fuga1
hoge2, fuga2, piyo2
hoge3, fuga3, piyo3
~~( 中略 )~~
hoge1000, fuga1000, piyo1000
```


マクロ実例(2. CSV編集)

- ❖ qqでマクロ記録開始
- ❖ ^f,で1つめの,まで移動
- ❖ dt,で, fuga1をカット
- ❖ \$で終端pで, fuga1をペースト
- ❖ <CR>で次の行の先頭に移動

```
hoge1, piyo1, fuga1
hoge2, fuga2, piyo2
hoge3, fuga3, piyo3
~~( 中略 )~~
hoge1000, fuga1000, piyo1000
```

マクロ実例(2. CSV編集)

- ❖ qqでマクロ記録開始
- ❖ ^f,で1つめの,まで移動
- ❖ dt,で, fuga1をカット
- ❖ \$で終端pで, fuga1をペースト
- ❖ <CR>で次の行の先頭に移動
- ❖ qでマクロ記録停止

```
hoge1, piyo1, fuga1
hoge2, fuga2, piyo2
hoge3, fuga3, piyo3
~~( 中略 )~~
hoge1000, fuga1000, piyo1000
```

マクロ実例(2. CSV編集)

- ❖ qqでマクロ記録開始
- ❖ ^f,で1つめの,まで移動
- ❖ dt,で, fuga1をカット
- ❖ \$で終端pで, fuga1をペースト
- ❖ <CR>で次の行の先頭に移動
- ❖ qでマクロ記録停止
- ❖ 999@qでマクロ再生

```
hoge1, piyo1, fuga1
hoge2, piyo2, fuga2
hoge3, piyo3, fuga3
~~( 中略 )~~
hoge1000, piyo1000, fuga1000
```

マクロ実例(3. 複数ファイル)

- ❖ 前述の例のCSVファイルが100個あったらどうする？
- ❖ **:args** コマンドと **:argdo** コマンドを使う
- ❖ **:args** で対象としたいファイルを(複数)指定し、**:argdo** で任意のコマンドを指定することで、指定したファイルすべてにコマンドを適用できる

マクロ実例(3. 複数ファイル)

- ❖ 実行例(カレントディレクトリ配下にファイルが配置されていると仮定)
 - ❖ **:args *.csv**
 - ❖ カレントディレクトリ配下にある.csvを対象とする
 - ❖ **:argdo normal! 1000@q**
 - ❖ normal! コマンドはnormalモードでのコマンドを実行する
 - ❖ 1000@qを.csvファイルすべてに対して適用できる
 - ❖ 合わせて保存したければ:argdo execute('normal! 1000@q') | update

目次

- ❖ マクロとは
- ❖ マクロの使い方
- ❖ マクロ実例
- ❖ マクロを使いこなすために

マクロを使いこなすために

- ❖ マクロを使うこと自体は難しいことではないが、マクロを使いこなすのは難しい
- ❖ マクロはVim操作の集大成。Vimの操作を使いこなせていなければ、効果的にマクロを使うことはできない

マクロを使いこなすために

- ❖ 例えば今回上げた例でいうと
 - ❖ 連番入力の例では、<C-a>が必要
 - ❖ CSV編集の例では、移動のf, t、オペレータのdが必要

マクロを使いこなすために

- ❖ マクロを使う上で知っておくべきコマンドは無数にある
 - ❖ 単語移動(w, b, e, ge)
 - ❖ 検索(/, ?, *, #)
 - ❖ テキストオブジェクト(iw, ib, ipなどなど)
 - ❖ オペレーター(d, c, yなどなど)
 - ❖ 行移動(gg, G, {, })
 - ❖ などなど

マクロを使いこなすために

- ❖ マクロ習得には訓練が必要
- ❖ マクロ訓練に適したサービス
 - ❖ VimGolf(<http://www.vimgolf.com/>)
 - ❖ 厳選マクロ練習用(<https://gist.github.com/deris/db66ddfd8ad5d4877f0>)
- ❖ マクロ習得に必要なVim基礎力を上げるための本
 - ❖ 実践Vim
- ❖ あと重要なのは、日々のVim鍛錬(マクロも意識して)

マクロを使いこなすために

- ❖ マクロは本当に便利なのであまり使いこなせていない方は是非使ってみてください
- ❖ 面白いマクロの活用方法をご存じの方は、懇親会などでお話いただけるとうれしいです

ご静聴ありがとうございました