



vim 入門
#061_vimvimvim #0

@virbr0

2019.04.19

発表内容

1. 自分と横田研と勉強会の紹介

2. **Vim** とは

3. **Vim** 操作入門

- モード
- Normal モード
- Command モード
- Visual モード

4. **Vim** の設定



発表内容

1. 自分と横田研と勉強会の紹介

2. Vim とは

3. Vim 操作入門

- モード
- Normal モード
- Command モード
- Visual モード

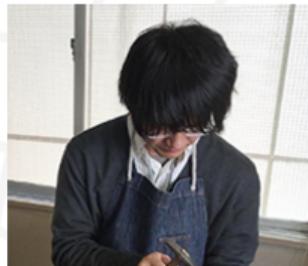
4. Vim の設定



自己紹介 大友広幸/つき

どんな人?

- ▶ 情報理工学院 情報工学系 修士2年
- ▶ Vim 歴5年くらい
- ▶ 昔は VisualStudio, Notepad++ を使っていた
- ▶ 補完プラグインの CUDA 周りをいじったり
- ▶ 普段 Vim で書いているのは C++, L^AT_EX
- ▶ おもしろアーキテクチャ大好き民族見習い



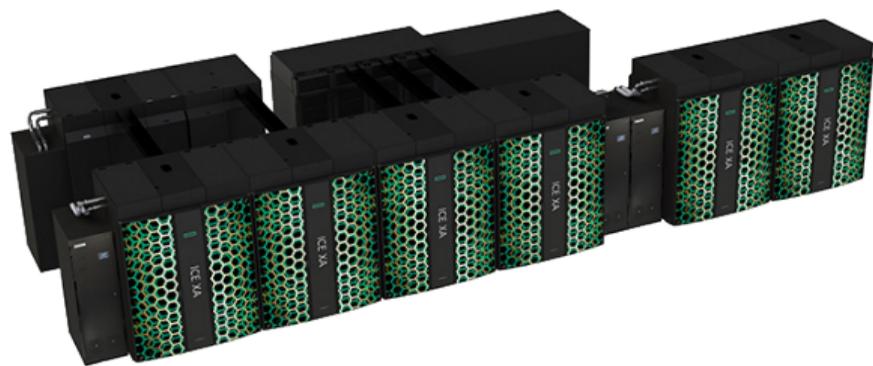
アカウント

- ▶ github : @enp1s0
- ▶ twitter : @virbr0

横田理央研究室

研究内容

GPU などのアクセラレータや大規模なスーパーコンピュータで物理シミュレーションや深層学習などの計算をぶん回す研究



エディタ比率

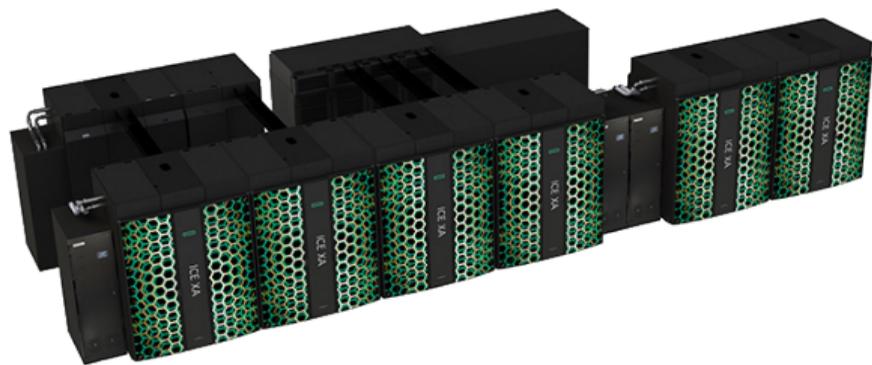


出典 <https://www.titech.ac.jp/news/2017/037500.html>

横田理央研究室

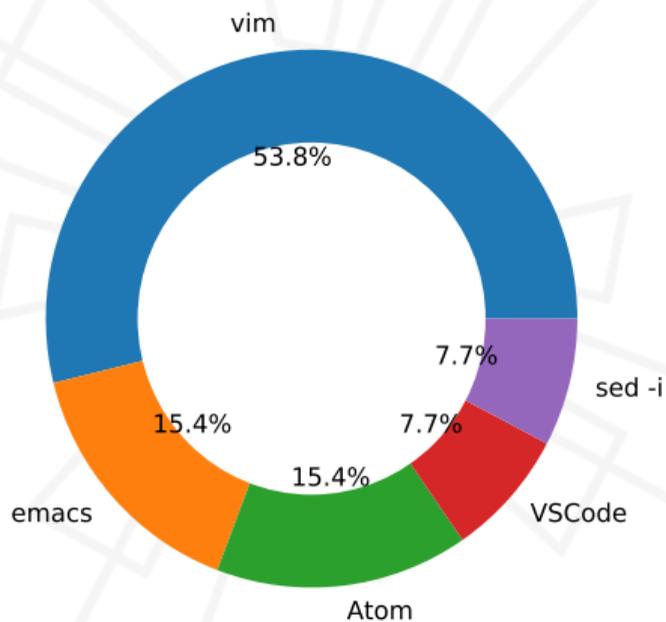
研究内容

GPU などのアクセラレータや大規模なスーパーコンピュータで物理シミュレーションや深層学習などの計算をぶん回す研究



出典 <https://www.titech.ac.jp/news/2017/037500.html>

エディタ比率



この勉強会について

勉強テーマ

- ▶ vim の使い方やプラグインの管理方法, 編集や自作
- ▶ vim に限らず tmux や zsh などの周辺環境, linux など

進め方

- ▶ 1~2ヶ月に1度くらいの頻度で開きたい
- ▶ 聞くだけでも良いけど, 発表するほうが勉強になるよ
- ▶ 基本は自分の好きな内容について語れば良い

#061_vimvimvim の名前の由来

- ▶ 横田研の Slack の vim チャンネルの名前
- ▶ #062_emacs などもある
- ▶ ハッシュタグにもなる, 便利

この勉強会について

勉強テーマ

- ▶ vim の使い方やプラグインの管理方法, 編集や自作
- ▶ vim に限らず tmux や zsh などの周辺環境, linux など

進め方

- ▶ 1~2ヶ月に1度くらいの頻度で開きたい
- ▶ 聞くだけでも良いけど, 発表するほうが勉強になるよ
- ▶ 基本は自分の好きな内容について語れば良い

#061_vimvimvim の名前の由来

- ▶ 横田研の Slack の vim チャンネルの名前
- ▶ #062_emacs などもある
- ▶ ハッシュタグにもなる, 便利

この勉強会について

勉強テーマ

- ▶ vim の使い方やプラグインの管理方法, 編集や自作
- ▶ vim に限らず tmux や zsh などの周辺環境, linux など

進め方

- ▶ 1~2ヶ月に1度くらいの頻度で開きたい
- ▶ 聞くだけでも良いけど, 発表するほうが勉強になるよ
- ▶ 基本は自分の好きな内容について語れば良い

#061_vimvimvim の名前の由来

- ▶ 横田研の Slack の vim チャンネルの名前
- ▶ #062_emacs などもある
- ▶ ハッシュタグにもなる, 便利

発表内容

1. 自分と横田研と勉強会の紹介

2. **Vim** とは

3. **Vim** 操作入門

- モード
- Normal モード
- Command モード
- Visual モード

4. **Vim** の設定



Vim とは

何であるか

- ▶ 効率的にファイルを編集するエディタ
- ▶ メールや各種設定ファイルの編集など全てに精通

何でないか

- ▶ 初見で直感で操作できるデザイン方針ではない
- ▶ 文章を整形しながら書ける「ワープロ」ではない

Vim とは

何であるか

- ▶ 効率的にファイルを編集するエディタ
- ▶ メールや各種設定ファイルの編集など全てに精通

何でないか

- ▶ 初見で直感で操作できるデザイン方針ではない
- ▶ 文章を整形しながら書ける「ワープロ」ではない

発表内容

1. 自分と横田研と勉強会の紹介

2. **Vim** とは

3. **Vim** 操作入門

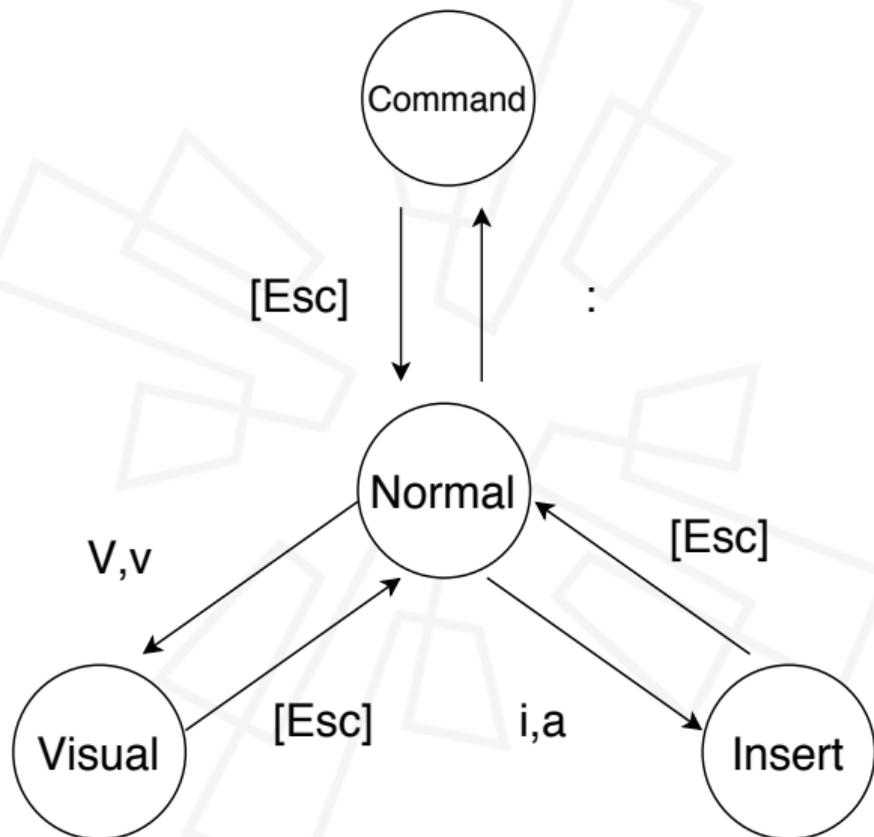
- モード
- Normal モード
- Command モード
- Visual モード

4. **Vim** の設定



モード

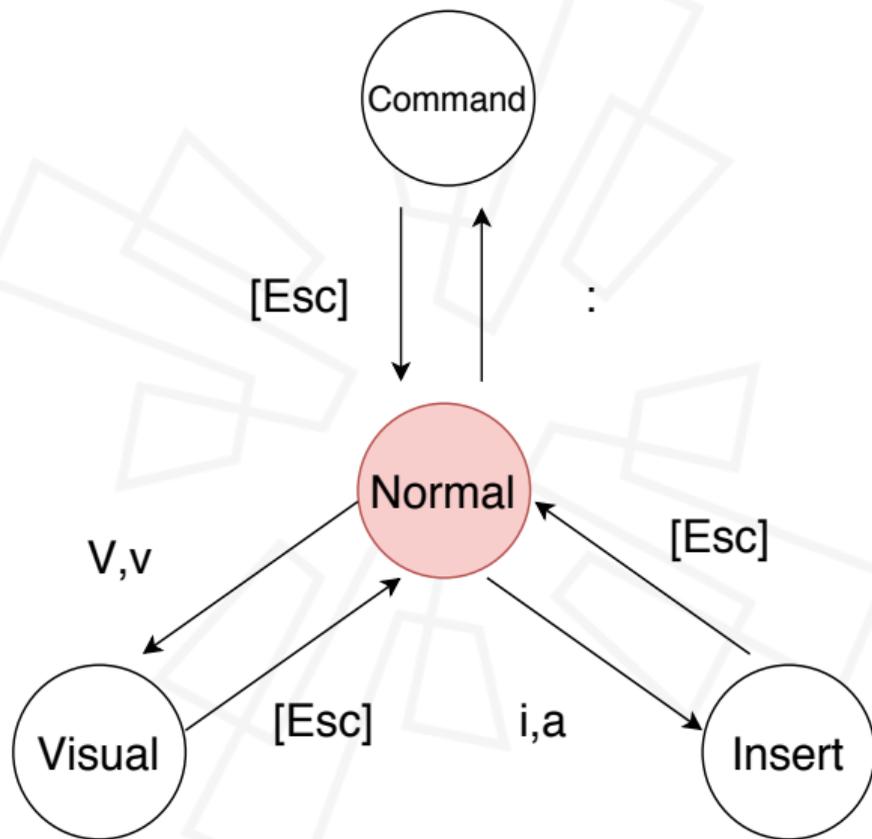
- ▶ Normal モード
Vim を開いたときのモード
- ▶ Insert モード
文字を入力するときのモード
- ▶ Command モード
:で始まるモード (:wq など)
- ▶ Visual モード
範囲選択をするときのモード



Normal モード

何ができるの？

- ▶ 各モードへの遷移
- ▶ カーソル移動
- ▶ 文字列削除
- ▶ 文字列検索
- ▶ etc.



Insert モードへの移り方

- ▶ カーソル前に挿入
i
- ▶ カーソル後に挿入
a
- ▶ 行頭に挿入
|
- ▶ 行末に挿入
A
- ▶ 次の行に挿入
o
- ▶ 前の行に挿入
O



カーソル移動

- ▶ h(左), j(下), k(上), l(右)
- ▶ 0(行頭), \$(行末)
- ▶ gg(先頭行), G(最終行)
- ▶ 行内文字検索移動
f[char]
- ▶ 単語飛ばしで次へ移動
w
- ▶ 単語飛ばしで前へ移動
b



削除と貼り付け

削除

- ▶ カーソル下 1 文字削除
x
- ▶ 1 行削除
dd
- ▶ 行のカーソル以降削除
D

削除すると"コピー"した状態になる

貼り付け

- ▶ カーソル後に貼り付け
p
- ▶ カーソル前に貼り付け
P

コピー

Vim では"ヤンク"と呼ばれる

- ▶ 1 行ヤンク
yy

その他のヤンクについては Visual モードで紹介

削除と貼り付け

削除

- ▶ カーソル下 1 文字削除
x
- ▶ 1 行削除
dd
- ▶ 行のカーソル以降削除
D

削除すると"コピー"した状態になる

貼り付け

- ▶ カーソル後に貼り付け
p
- ▶ カーソル前に貼り付け
P

コピー

Vim では"ヤンク"と呼ばれる

- ▶ 1 行ヤンク
yy

その他のヤンクについては Visual モードで紹介

削除と貼り付け

削除

- ▶ カーソル下 1 文字削除
x
- ▶ 1 行削除
dd
- ▶ 行のカーソル以降削除
D

削除すると"コピー"した状態になる

貼り付け

- ▶ カーソル後に貼り付け
p
- ▶ カーソル前に貼り付け
P

コピー

Vim では"ヤンク"と呼ばれる

- ▶ 1 行ヤンク
yy

その他のヤンクについては Visual モードで紹介

戻る/進む

- ▶ 直前の操作を取り消す (Undo)
u
- ▶ 取り消しを取り消す (Redo)
Ctrl-r



コマンドの繰り返しと組み合わせ

コマンドの繰り返し

- ▶ 同じコマンドを指定回数繰り返すことが可能
- ▶ 10行下へ移動
10j
- ▶ 6行削除
6dd

コマンドの組み合わせ

- ▶ 一部のコマンドは他のコマンドと組み合わせることが可能
- ▶ 以下の行全削除
dG
- ▶ 3単語削除
3dw

検索

検索文字列の指定

- ▶ 文字列で指定
/[target string]
- ▶ カーソル直下の文字列を指定
*

検索結果の移動

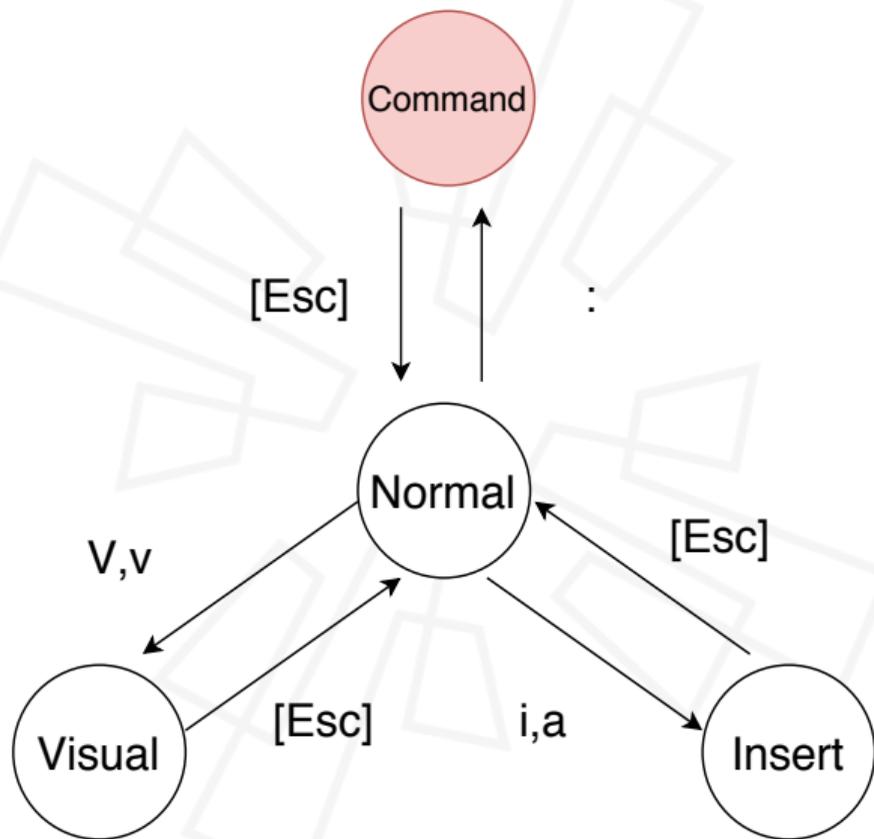
- ▶ 次へ
n
- ▶ 前へ
N



Command モード

何ができるの？

- ▶ 移動
- ▶ 置換
- ▶ ファイルに保存
- ▶ ファイルを開く
- ▶ vim の終了
- ▶ 外部プログラムの呼び出し
- ▶ etc.



置換

置換の基本

ファイル全体から hoge を foo に置換

```
:%s/hoge/foo/g
```

各行ではじめに見つかった **hoge** のみ **foo** に置換

ファイル全体から hoge を foo に置換

```
:%s/hoge/foo
```

他にも...

正規表現で置換もできます。

確認しながら置換

hoge を foo に置換するかを確認しながら置換

```
:%s/hoge/foo/gc
```

行の範囲を指定して置換

20 行目から 67 行目の hoge を foo に置換

```
:20,67s/hoge/foo/g
```

置換

置換の基本

ファイル全体から hoge を foo に置換

```
:%s/hoge/foo/g
```

各行ではじめに見つかった **hoge** のみ **foo** に置換

ファイル全体から hoge を foo に置換

```
:%s/hoge/foo
```

他にも...

正規表現で置換もできます。

確認しながら置換

hoge を foo に置換するかを確認しながら置換

```
:%s/hoge/foo/gc
```

行の範囲を指定して置換

20 行目から 67 行目の hoge を foo に置換

```
:%s/hoge/foo/g
```

置換

置換の基本

ファイル全体から hoge を foo に置換

```
:%s/hoge/foo/g
```

各行ではじめに見つかった **hoge** のみ **foo** に置換

ファイル全体から hoge を foo に置換

```
:%s/hoge/foo
```

他にも...

正規表現で置換もできます。

確認しながら置換

hoge を foo に置換するかを確認しながら置換

```
:%s/hoge/foo/gc
```

行の範囲を指定して置換

20 行目から 67 行目の hoge を foo に置換

```
:20,67s/hoge/foo/g
```

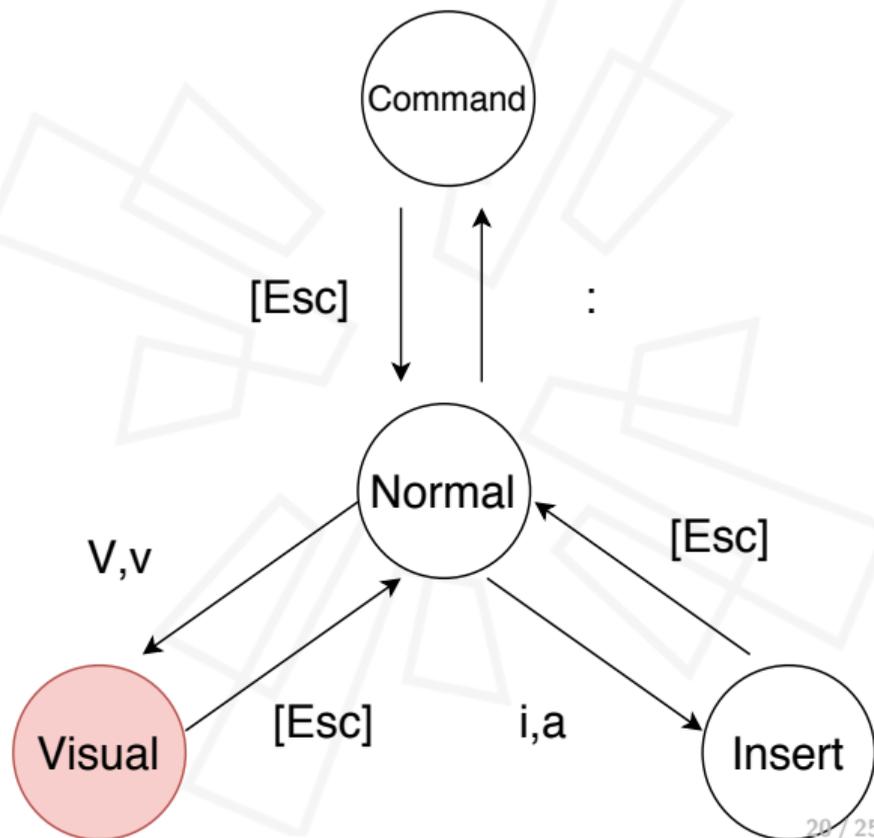
Visual モード

何ができるの？

- ▶ 文字列選択

選択すると何ができるの？

- ▶ 削除
- ▶ 置換
- ▶ 整形
- ▶ etc.



Visual モード

選択の仕方

- ▶ カーソル下から選択
v
- ▶ 行選択
V
- ▶ 矩形選択
Ctrl - v

選択範囲を削除

d

選択範囲を置換

1. : を入力
2. : '<'> s/hoge/foo

矩形選択前後に文字列を挿入

矩形選択後に

1. I/A を入力
2. 挿入したい文字列を入力
3. ESC を入力

することで矩形選択範囲前/後に文字列を挿入

Visual モード

選択の仕方

- ▶ カーソル下から選択
v
- ▶ 行選択
V
- ▶ 矩形選択
Ctrl - v

選択範囲を削除

d

選択範囲を置換

1. : を入力
2. : '<'> s/hoge/foo

矩形選択前後に文字列を挿入

矩形選択後に

1. I/A を入力
2. 挿入したい文字列を入力
3. ESC を入力

することで矩形選択範囲前/後に文字列を挿入

Visual モード

選択の仕方

- ▶ カーソル下から選択
v
- ▶ 行選択
V
- ▶ 矩形選択
Ctrl - v

選択範囲を削除

d

選択範囲を置換

1. : を入力
2. :'<,'> s/hoge/foo

矩形選択前後に文字列を挿入

矩形選択後に

1. I/A を入力
2. 挿入したい文字列を入力
3. ESC を入力

することで矩形選択範囲前/後に文字列を挿入

Visual モード

選択の仕方

- ▶ カーソル下から選択
v
- ▶ 行選択
V
- ▶ 矩形選択
Ctrl - v

選択範囲を削除

d

選択範囲を置換

1. : を入力
2. :'<,'> s/hoge/foo

矩形選択前後に文字列を挿入

矩形選択後に

1. I/A を入力
2. 挿入したい文字列を入力
3. ESC を入力

することで矩形選択範囲前/後に文字列を挿入

発表内容

1. 自分と横田研と勉強会の紹介

2. **Vim** とは

3. **Vim** 操作入門

- モード
- Normal モード
- Command モード
- Visual モード

4. **Vim** の設定



Vim の設定

設定ファイル

Vim の設定ファイルは
`$HOME/.vimrc`
です。

自分の使う最低限の最低限の `vimrc`

```
set number  
set relativenumber  
set shiftwidth=4  
set tabstop=4  
set expandtab  
syntax on
```



Vim の設定

設定ファイル

Vim の設定ファイルは
`$HOME/.vimrc`
です。

自分の使う最低限の最低限の **vimrc**

```
set number  
set relativenumber  
set shiftwidth=4  
set tabstop=4  
set expandtab  
syntax on
```



ウガンダへの寄付

Vim は Freeware ではなく **Charityware** です。
ウガンダの子どもたちに寄付することが奨励されています。

寄付するには
:help uganda で寄付方法を確認できます。

Bram Moolenaar <Bram@moolenaar.net>

To 自分 ▾

Ootomo Hiroyuki -

Thank you very much for your donation!

- Bram

終わり

