



Luciano Pacheco

pacheco@simplesconsultoria.com.br

PythonBrasil[5]

DDD.py

Develop
Debug
Distribute

```
>>> me = Palestrante()
```



me.apresentando()

Luciano Pacheco - lucmult

Python e Plone

Simplex Consultoria

```
>>> dev = Develop()
```



dev.addTool(vim)

```
>>> pprint(vim.modes)
```

- Modo Normal - execute comandos - manipule o texto
- Modo Insert - Edite o texto
- Modo Visual - Seleccione o texto

```
>>> help(vim.mode['insert'])
```



Entrar no modo Normal

- **i** - *insert* na posição do cursor
- **a** - *append* adicione após a sua posição
- **A** - *Append* ao final da linha
- **I** - *Insert* no começo da linha
- **R** - *replace*
- **c** - *change* substituir (**cw**, **c\$**, **cb**)
- **C** - Substitui a linha
- **o** - inserindo na próxima linha
- **O** - inserindo na linha anterior



NÃO USE CAPS LOCK



Navegando no texto

- Evite remover a mão do teclado
 - Inclusive evite as “setinhas”
- h – esquerda
- l – direita
- k – cima
- j – baixo

- Sim é difícil nos primeiros dias, mas vale a pena. :-)



Navegando no Texto

- Ainda em modo normal
- `w` - avança palavra
- `b` - volta palavra
- `gg` - começo do arquivo
- `G` - fim do arquivo
- `Ctrl + f` - Avança pág. (= Page Down)
- `Ctrl + b` - Volta pág. (= Page Up)
- `0` - começo da linha (coluna zero)
- `$` - fim da linha (igual regex)



Navegando no Texto

- Ainda em modo normal
- `w` - avança palavra
- `b` - volta palavra
- `gg` - começo do arquivo
- `G` - fim do arquivo
- `Ctrl + f` - Avança pág. (= Page Down)
- `Ctrl + b` - Volta pág. (= Page Up)
- `0` - começo da linha (coluna zero)
- `$` - fim da linha (igual regex)



Manipular texto

- Ainda no modo normal
- `y` - `yank` - copiar
- `yw` - copiar palavra
- `yy` - copiar linha - ou `Y`
- `p` - `paste` - colar



Manipular texto

- d - delete - recortar
- dw - deletar palavra
- dd - deletar linha
- D - deletar até o final da linha
- ddp - delete a linha e cola, na prática inverte as linhas
- x - deleta caracter
- xp - deleta e cola, na prática troca letras de lugar



Outros comandos

- u - undo - desfazer ou g-
- Ctrl + r - redo - refazer ou g+
- Ctrl + n - autocomplete Ctrl + n - Ctrl + o
- Ctrl + o - executar 1 comando do modo normal
- . - repete a última edição



Editando várias linhas

- Ctrl+v - selecione o começo da linha
- I - Entre no modo edição para todas as linhas
- faça a edição necessária
- ESC - irá aplicar em todas as linhas



Trabalhe com Abas

- Abrindo uma nova aba - tab edit
 - `:tabe nome_do_arquivo.txt`
- `vim -p arquivo.txt arquivo2.txt`
- Próxima Aba:
 - `Ctrl + PgDn` ou `gt`
- Aba anterior:
 - `Ctrl + PgUp` ou `gT`



Plugins Legais

- NERD_tree - árvore de arquivos
- EnhancedCommentify - comentar linhas em qualquer linguagem
- snipMate - snippets fácil de customizar
- taglist - navegar classes, métodos, funções
- surround - () [] {} <tags> etc...
- tasklist - TODO list



dev.addTool(pdb)

- Python Debugger

```
>>> import pdb; pdb.set_trace()
```

```
>>> <stdin>(1)?() ->None
```

```
(Pdb)
```

```
(Pdb) help
```



Principais comandos

- **n** - **next** - vai para a próxima linha
- **s** - **step in** - executa entradas nas funções, métodos e classes (muito bom)
- **j** - **jump** - pule para a linha
- **l** - **list** - listar o código - **l** ou **l 10** ou **l 10, 20**
- **c** - **continue** - continue até o fim ou até o próximo break point
- **q** - **quit** - sai fora, pare tudo



Principais comandos

- **w** - **where** - exibe a pilha das chamadas até o break point
- **u** - **up** - sobe na pilha chamadas
- **down** - desce na pilha chamadas
- **!** - escape nas variáveis, usado qdo temos variáveis com os nomes dos comando
 - **!w** - imprime a variável, não será chamado o comando **w** (**where**)



Outros Debugger

- ipdb - baseado no IPython
- `>>> import ipdb; ipdb.set_trace()`
- Highlight Syntax e Autocomplete
- pdb - janelinhas no terminal
- `>>> import pdb; pdb.set_trace()`
- Highlight Syntax
- Pode-se utilizar o IPython dentro dele



Principais comandos

- w - where - mostre a pilha das chamadas até o ponto onde vc está
- up - sobe na pilha das chamadas
- down - desce na pilha chamadas
- ! - dá um escape nas variáveis, usado qdo temos variáveis com os nomes dos comando
- !w - imprime a variável, não será chamado o comando w (where)



Distruindo como eggs

- `setuptools`
- Hospede-o no pypi
- `easy_install` para o seu produto
- Dê visibilidade ao seu produto



Eggs

- `collective.dist`
 - plugins para o `setuptools`
- `paster`
 - template para gerar um pacote
- `Distribute` – sucessor do `setuptools`

