* 1. 2016/3/28 vim的学习

1. vi与vim都是文书编辑器（vim是vi的进阶版）
2. vi有三种模式
3. 一般模式：在一般模式中可以使用【删除】【上下左右】【复制.粘贴】
4. 编辑模式：使用【i.I.o.O.a.A.r.R】来进入编辑模式
5. 指令命令模式：在一般模式中输入【：/ ?】三种符号+w，w！，wq，wq！可以实现读取，存盘，取代字符，离开vi。
6. 文档的创建和文档的保存。

（1）vi test.txt可以创建文档或查看文档

（2）输入【i.I.o.O.a.A.r.R】进入编辑模式可以编辑或修改文本中的内容

（3）编辑完成后按Esc按键进入一般模式后输入【：wq】就可保存编辑后的文档然后退出编辑模式，如果有权限即可使用【：wq！】强制保存

4.按键的说明

（1）【Ctrl】+【f】：向下移动一页

（2）【Ctrl】+【b】：向上移动一页

（3）【Ctrl】+【d】：向下移动半页

（4）【Ctrl】+【u】：向上移动半页

（5）+：光标移动到非空格符的下一列

（6）-：光标移动到非空格符的上一列

（7）n+空格：光标向后移动到n个字符的地方

（8）0或【Home】：光标移动到这一行最前面字符处

（9）$或【End】：光标移动到这一行的最后字符处

（10）H:光标移动到这个屏幕的最上方那一行的第一个字符处

（11）M:光标移动到这个屏幕的中央那一行的第一个字符处

（12）L：光标移动到这个屏幕的最下方那一行的第一个字符处

（13）G：移动到这个档案的最后一行

（14）nG：移动到这个档案的第n行

（15）gg：移动到这个档案的第一行 ，也可用1G

（16）n+回车：光标向下移动n行

5.搜索与取代

（1） / word：向光标下寻找一个名称为“word“的字符串

（2）？word：向光标之上寻找一个名称为“word”的字符串

（3）n：是个字母n，表示重复前一个搜寻动作，搜寻下一个目标

（4）N：大写的字母N，表示重复前一个搜寻动作，搜寻上一个目标

（5）:n1,n2s/word1/word2/g：在n1行与n2行之间将word1字符串由word2字符串取代

（6）:1,$s/word1/word2/g：从第一行到最后一行word1都由word2取代

（7）:1,$s/word1/word2/gc：从第一行到最后一行word1都由word2取代，在取代前显示提示字符给用户确认是否需要取代【y】：确定修改【n】：不修改

6.删除，复制与粘贴

（1）x,X：x向光标后删除一个字符，X向前删除一个字符

（2）nx：向后删除n个字符

（3）dd：删除游标所在的那一整行

（4）ndd：删除光标所在向下n行

（5）d1G：删除光标所在到第一行的所有数据

（6）dG：删除光标所在到最后一行的所有数据

（7）d$：删除游标所在，到该行最后一个字符

（8）d0：删除游标所在到该行第一个字符

（9）yy：复制游标所在的一整行

（10）nyy：复制光标所在的向下n列

（11）y1G：复制光标所在列到第一列的所有数据

（12）yG：复制光标所在列到最后一列的所有数据

（13）y0：复制光标所在的那个字符到该行行首的所有数据

（14）y$：复制光标所在的那个字符到该行行尾的所有数据

（15）p,P：小p粘贴到下一行，P粘贴到上一行

（16）J：将光标所在行与下一行的数据结合成一列

（17）c：重复删除多个数据，例如向下删除10行【10cj】

（18）u：复原前一个动作

（19）【Ctrl】+r：重做上一个动作

* 1. Vim的学习

1.vim环境的变化

（1）:set nu：显示行号，设定之后，在每一行的前面显示该行的行号

（2）:set nonu：取消每一行的行号

2.vim的区块选择的按键意义

（1）v：字符选择，会将光标经过的地方反白选择

（2）V：行选择，会将光标经过的行反白选择

（3）【Ctrl】+v：区块选择，可以用长方形的方式选择资料

（4）y：将反白的地方复制起来

（5）d：将反白的地方删除掉

3.多档案编辑的按键

（1）:n：编辑下一个档案

（2）:N：编辑上一个档案

（3）:files：列出目前这个vim的开启的所有档案

4.多窗口功能

（1）:sp{filename}：启动另一个档案加入档名

（2）【Ctrl】+w+下键：先按下【Ctrl】不放，再按下w后放开所有按键，然后再按下“下键”，则光标可移动到下方的窗口

（3）【Ctrl】+w+上键：方法和（2）一样，光标移动到上面的窗口

（4）【Ctrl】+w+q：离开当前窗口

5.环境设定与记录

（1）:set nu：设定行号

（2）:set nonu：取消行号

（3）:set hlsearch：设定搜寻的字符串反白的设定值，默认值是hlsearch

（4）:set onhlsearch：不设定搜寻的字符串反白的设定值

（5）:set autoindent：自动缩排

（6）:set noautoindent：不自动缩排

（7）:set backup：是否储存备份档，一般是nobackup（不储存），如果设定，更新任何一个，源文件会被另存一个档名为filename~的档案

（8）:set ruler：设置状态栏说明是否显示

（9）:set showmode：是否显示—INSERT—之类的字样在左下角的状态栏内

（10）:set all：显示目前所有的环境参数设定值

（11）:set：显示与系统默认值不同的设定参数，一般来说就是你有自行变动过的设定参数

（12）:syntax on和syntax off：是否依据程序相关语法显示不同颜色

（13）:set backspace=(012)：当backspace为2时可以删除任意值，为0或1时，仅可删除刚输入的字符。

（14）:set bg=dark和:set bg=light：设置显示的颜色色调light为深 蓝色

* 1. 2016/3/29 Ubuntu中Linux的命令

1. exit ：注销Linux
2. ls –al~：列出所有文件目录
3. date：显示日期与时间
4. echo $LANG：编码格式的查询
5. LANG=编码格式：编码格式的设定
6. cal：日历的显示
7. bc：进入计算器功能
8. quit：结束计算器功能
9. 【Tab】键：如（ca【Tab】【Tab】）命令补全和档案补全的功能
10. 【Ctrl】+c：中断指令
11. 【Ctrl】+d：输入结束
12. Man page：查看与page相关的字符串【q】是退出查看
13. 【ctrl】+G：取得联机帮助
14. 【ctrl】+X：离开naon软件，若有修改过档案会提示是否需要储存
15. 【ctrl】+O：储存档案，若你有权限的话就能够储存档案
16. 【ctrl】+R：从其他档案读入资料，可以将某个档案的内容贴在本档案中
17. 【ctrl】+W：搜寻字符串，这个也是很有帮助的指令
18. 【ctrl】+C：说明目前光标所在处的行数与列数等信息
19. 【ctrl】+\_：可以直接输入行号，让光标快速移动到该行
20. 【alt】+Y：校正语法功能开启或关闭（单击开，再单击关）
21. 【alt】+M：可以支持鼠标来移动光标的功能
22. who：查看目前谁在线
23. netstat –a：查看网络连接状态
24. ps –aux：查看背景执行的程序
25. shutdown：结束当前所有工作
26. sync：将数据同步写入硬盘中的指令
27. reboot或halt：重启计算机
28. poweroff：关机
    1. 变量的设定与修改
29. echo $变量名：变量名的设定
30. 变量名=值：将变量名中的内容赋一个值
31. 变量与变量之间可以拼接字符串。如：version=”$P ATH”+/home/bin
32. unset 变量名：取消变量的设定
33. export PATH：将变量PATH变为环境变量（在父程序中定义的变量能在子程序中找到）
    1. 双引号与单引号的区别
34. 双引号能保持原变量的内容
35. 单引号将失去原有的变量内容
36. ``：在``内的指令都将被先执行。如：ls –l`locate crontab`
    1. 环境变量的功能
37. 使用env观察环境变量与常见环境变量说明：declare - i number=$RANDOM\*10/32768;echo $number：产生一个随机数
38. 用set观察所有变量（环境变量和自定义变量）
39. PS1：提示字符的设定

一．\d：可显示的日期格式，如：“Mon Feb 2”

二．\H：完整的主机名。

三．\h：仅取主机名在第一个小数点之前的名字

四．\t：显示时间，为24小时格式

五．\T：显示时间，为12小时格式

六．\A：显示时间，为24小时格式（没有秒）

七．\@：显示时间，为12小时格式（没有秒）

八．\u：目前使用者的账号

九．\v：BASH的版本信息

十．\w：完整的工作目录名称，由根目录写起的目录名称

十一.\W：利用basename函数取得工作目录名称只会列出最后一个目录名

十二.\#：下达的第几个指令

十三.\$：提示字符

* 1. 2016/4/7GIT(版本控制系统)学习

1.Git的设定被存放在用户本地目录的.gitconfig档案里

$ git config - -global user.name -----------<用户名>

$ git config - -global user.email -----------<电子邮箱>

2.以下命令能让Git以彩色显示

$ git config - -global color.ui auto

1. 以Git命名设定别名。如【checkout】缩略为【co】，然后就使用【co】来执行命令

$ git config - - global alias.co checkout

4．首先在任意一个地方创建tutorial目录。然后使用init命令把该tutorial目录移动到本地Git数据库

$ git init

1. 按照以下步骤把新创建的tutorial目录设置到Git数据库

$ mkdir tutorial

$ cd tutorial

$ git init

5．使用status命令确认工作树和索引的状态

$ git status

6．执行status命令以确认tutorial目录的状态

$ git status

7．将文件加入索引，要使用add命令。在<file>指定加入索引的文件，用空格分割可以指定多个文件。

$ git add <file>..

Eg: $ git add sample.txt

$ git status

8．执行commit命令 之后确认状态

Eg: $git commit -m “first commit”

$git status

9．使用log命令，我们可以看到新提交的信息

$ git log

10．使用gitk工具可以在GUI下确认提交信息

$ gitk

11．使用remote 指令添加远程数据库

$ git remote add <name> <url> name--------远程数据库的名称。url------

远程数据库的URL

Eg： $ git remote add origin https://[your\_space\_id].backlogtool.com/git/[your\_project\_key]/tutorial.git

12．使用push命令向数据库推送更改内容

$ git push <repository> <refspec> … repository-----------目标地址。refspec--------指定推送的分支

Eg： $ git push -u origin master

Username:<用户名>

Password：<密码>

13．使用clone指令可以复制数据库

$ git clone <repository> <directory> repository----------指定远程数据库的URL。directory------------指定新目录的名称

Eg： $ git clone <https://nulab.bacllog.jp/git/BLG/tutorial.git> tutorial2

14．修改后提交

$ git add sample.txt

$ git commit -m “添加add说明”

15．推送变更后的添加到远程数据库

$ git push ------- ----在克隆的数据库目录执行推送时，可以省略数据库和分支名称

16. 使用pull指令进行拉取，省略数据库名，默认在origin数据库中pull

$ git pull <repository> <refspes>…

Eg:$ git pull origin master

17．解决冲突：当两个人同时修改了远程数据库的同一个地方时需要手动修改冲突部分后才可以提交

18. “！ Rejected 表示推送失败 ”

* 1. 2016/4/8 GIT高级篇

1. 什么是分支

（1）分支是为了将修改记录的整体流程分叉保存

（2）分叉后的分支不受其它分支的影响，所以在同一个数据库里可以同时修改。

2．分支的运用

（1）在Git中可以自由建立分支

（2）Git分支分为Merge分支和Topic分支

Merge分支：为了可以随时发布release而创建的分支

Topic分支：为了开发新功能或修改Bug等任务而建立的分支

3．分支的切换

1. 若要切换分支，就要进行checkout操作。
2. HEAD指向的是现在使用中的分支的最后一次更新
3. stash是临时保存文件修改内容的区域

4．分支的合并

（1）完成后的topic分支，合并回merge分支。有merge和rebase两种方式

Merge：保持修改内容的历史记录，但是历史记录会很复杂。

Rebase：历史记录简单，是在原有提交的基础上将差异内容反映进去，可能导致原本的提交内容无法正常运行

5．topic分支和merge分支的运用实例

（1）通过branch命令来创建分支：$ git branch <branchname>

Eg： $ git branch issue1

6．切换分支

（1）若要在新建的issue1分支进行提交，需要切换到issue1分支，要执行checkout命令以退出分支：$ git checkout <branch>

Eg：$ git checkout issue1

（2）在checkout命令指定 –b 选项执行，可以创建分支并切换

Eg：$ git checkout -b issue1

7．合并分支

（1）向master分支合并issue1分支的修改，执行merge命令以合并分支：$ git merge <commit>。在合并之前先切换master分支，然后把issue1分支导入到master分支：$ git checkout master

Eg：$ git checkout master

$ git merge issue1

8．删除分支

（1）删除issue1分支：$ git branch -d <branchname>

Eg：$ git branch -d issue1

（2）用branch命令确认是否被删除

Eg：$ git branch

9．解决合并的冲突

（1）创建分支issue2，issue3

（2）切换master分支后，与issue2分支合并

Eg：$ git checkout master

$ git merge issue2

$ git merge issue3

$ git add sample.txt

$ git commit -m “合并issue3分支”

10．用rebase合并

（1）取消刚才的合并

Eg：$ git reset --hard HEAD~

（2）切换到issue3分支后，对master执行rebase

Eg：$ git checkout issue3

$ git rebase master

（3）修改sample.txt中的合并冲突后切换到master分支

Eg：$ git checkout master

$ git merge issue3

11．添加标签

（1）使用tag命令添加标签：$ git tag <tagname>

Eg：$ git tag apple

（2）如果没有使用参数而执行tag，可以显示标签列表

Eg：$ git tag

1. 如果log命令添加 –decorate选项执行，可以显示包含标签资料的历史记录

Eg：$ git log --decorate

（5）添加注解标签：$ git tag -a <tagname>

Eg：$ git tag -am “连猴子都懂的Git” banana

（6）如果tag命令指定 –n 选项执行，可以显示标签的列表和注解

Eg：$ git tag -n

12．删除标签：$ git tag -d <tagname>

Eg：$ git tag -d banana

13．commit --amend

Eg：$ git add sample.txt

$ git commit --amend

$ git log

14．用revert命令取消提交：$ git revert HEAD

15．用reset来删除master分支最前面的两个提交：$ git reset --hard HEAD ~~

16．将其他分支的修改导入到master分支

Eg：$ git checkout master

$ git cherry-pick 99daed2

如果有冲突先修改后提交

$ git add sample.txt

$ git commit

17．用rebase –i汇合提交

$ git rebase -i HEAD~~

18．用rebase –i 修改提交

$ git rebase -i HEAD~~

19．merge –squash

$ git checkout master

$ git merge --squash issue1

* 1. git-am和git-patch的使用

1. git-patch里包含了提交的日志，作者，日期等信息
2. 打patch的方式：

Eg：patch -p1 < 001 –JFFS2…………缺点是：会把一些有用的信息丢失

1. git -am和patch -p1用法相同。（注意：在使用git -am前需要git am -abort）
2. 撤销整个am的东西：git -am -abort
3. 撤销一个patch：git am -skip