

Chapter 1: What and Why of Scripting with Bash

```
No LSB modules are available.
Distributor ID: Debian
Description:    Debian GNU/Linux 7.8 (wheezy)
Release:       7.8
Codename:      wheezy
pi@pilabs ~ $ .....
```

```
pi@pilabs ~ $ echo $BASH_VERSION
4.2.37(1)-release
pi@pilabs ~ $ .....
```

```
bash: warning: x: ignoring function definition attempt
bash: error importing function definition for `BASH_FUNC_x'
test
```

```
date
```

```
cat: /tmp/echo: No such file or directory
```

```
pi@pilabs /tmp $ type ls quote pwd do id
ls is aliased to `ls --color=auto'
quote is a function
quote ()
{
    local quoted=${1//\'/\'\\\'\\\'\'};
    printf "%s'" "$quoted"
}
pwd is a shell builtin
do is a shell keyword
id is /usr/bin/id
pi@pilabs /tmp $ _
```

gedit Preferences

View

Editor

Font & Colours

Plugins

Tab Stops

Tab width: 4 - +

Insert spaces instead of tabs

Enable automatic indentation

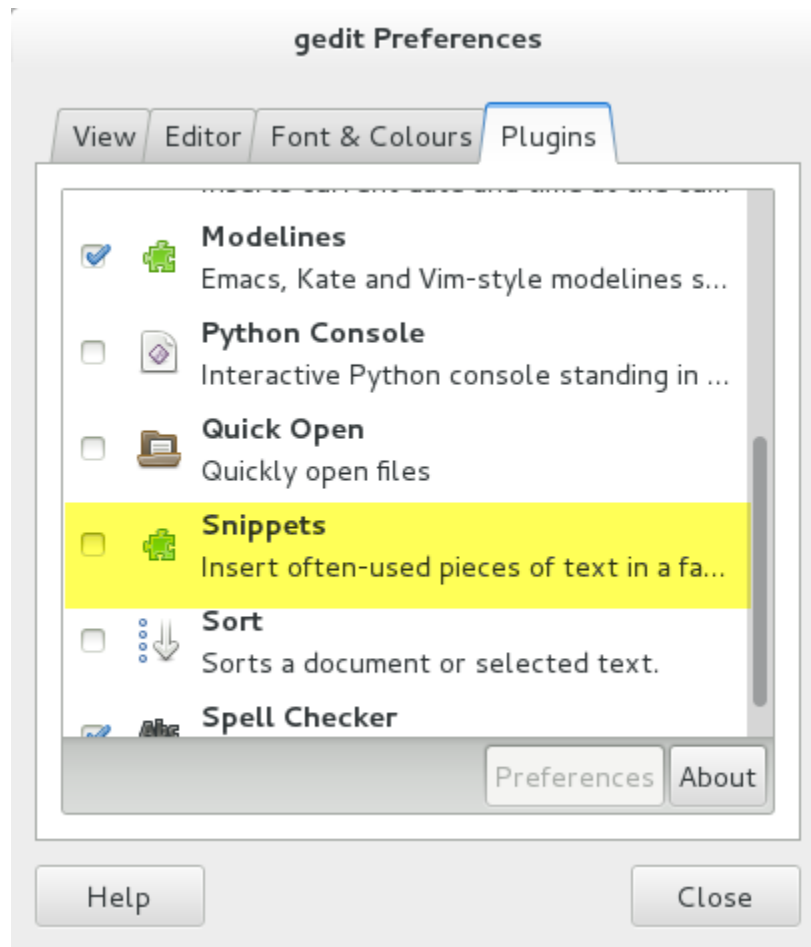
File Saving

Create a backup copy of files before saving

Autosave files every 10 - + minutes

Help

Close



```
#!/bin/bash
echo "Hello World"
exit 0
```

```
pi@pilabs ~ $ chmod +x $HOME/bin/hello1.sh
pi@pilabs ~ $ hello1.sh
Hello World
pi@pilabs ~ $ _
```

```
pi@pilabs ~ $ type hello1.sh
hello1.sh is hashed (/home/pi/bin/hello1.sh)
pi@pilabs ~ $ type -a hello1.sh
hello1.sh is /home/pi/bin/hello1.sh
pi@pilabs ~ $ type -t hello1.sh
file
pi@pilabs ~ $ _
```

```
#!/bin/bash
echo "Hello $1"
exit 0
~
```

```
pi@pilabs ~ $ hello2.sh fred
Hello fred
pi@pilabs ~ $ _
```

```
pi@pilabs ~ $
pi@pilabs ~ $ hello2.sh fred wilma betty barney
Hello fred wilma betty barney
pi@pilabs ~ $ _
```

```
pi@pilabs ~ $ echo "$USER earns $4"
pi earns
pi@pilabs ~ $ echo '$USER earns $4'
$USER earns $4
pi@pilabs ~ $ echo "$USER earns \$4"
pi earns $4
pi@pilabs ~ $ _
```

```
pi@pilabs ~ $ hello2.sh fred
You are using /home/pi/bin/hello2.sh
Hello fred
pi@pilabs ~ $ .....
```

```
pi@pilabs ~ $ hello2.sh fred
You are using hello2.sh
Hello fred
```

```
pi@pilabs ~ $ bash -v $HOME/bin/hello2.sh fred
#!/bin/bash
echo "You are using $(basename $0)"
basename $0)
basename $0)
basename $0
You are using hello2.sh
echo "Hello $*"
Hello fred
exit 0
pi@pilabs ~ $ _
```

```
pi@pilabs ~ $ bash -x $HOME/bin/hello2.sh fred
++ basename /home/pi/bin/hello2.sh
+ echo 'You are using hello2.sh'
You are using hello2.sh
+ echo 'Hello fred'
Hello fred
+ exit 0
pi@pilabs ~ $ _
```

Chapter 2: Creating Interactive Scripts

```
pi@pilabs ~ $ echo
```

```
pi@pilabs ~ $ _
```

```
pi@pilabs ~ $ echo -e "Which directory do you want to use? \c"  
Which directory do you want to use? pi@pilabs ~ $ _
```

```
pi@pilabs ~ $ hello3.sh  
Hello I hello3.sh may I ask your name: fred  
Hello fred  
pi@pilabs ~ $ _
```

`read -p <prompt> <variable name>`

```
pi@pilabs ~ $ hello3.sh  
May I ask your name: fred  
Hello fred  
Press any key to continue  
pi@pilabs ~ $ _
```

We see the entered text

With -s we don't see the entered text

```

#!/bin/bash
# Author: @theurbanpenguin
# Web: www.theurbanpenguin.com
# Script to prompt to back up files and location
# The files will be search on from the user's home
# directory and can only be backed up to a directory
# within $HOME
# Last Edited: July 4 2015
read -p "Which file types do you want to backup " file_suffix
read -p "Which directory do you want to backup to " dir_name
# The next lines creates the directory if it does not exist
test -d $HOME/$dir_name || mkdir -m 700 $HOME/$dir_name
# The find command will copy files the match the
# search criteria ie .sh . The -path, -prune and -o
# options are to exclude the backdirectory from the
# backup.
find $HOME -path $HOME/$dir_name -prune -o \
  -name "$file_suffix" -exec cp {} $HOME/$dir_name/ \;
exit 0

```

```

pi@pilabs ~ $ backup.sh
Which file types do you want to backup .sh
Which directory do you want to backup to backup
pi@pilabs ~ $ ls $HOME/backup
autogunk.sh          ECCcertgen.sh      irix.sh
autoungunk.sh       ECC-RSACertgen.sh launcher.sh
backup.sh            fixNT.sh           mkcerts.sh
bat.sh              FreeBSD.sh         mksmime-certs.sh
c89.sh              hello1.sh          opensslwrap.sh
CA.sh               hello2.sh          point.sh
connect_server.sh   hello3.sh          profile.sh
cygwin.sh            hpux10-cc.sh      pthread2.sh
do_ms.sh             install.sh         pthread.sh
pi@pilabs ~ $ _

```

```
pi@pilabs ~ $  
pi@pilabs ~ $ ping_server.sh  
Which server should be pinged localhost  
pi@pilabs ~ $ ping_server.sh  
Which server should be pinged 1.2.3.4  
Server Dead  
pi@pilabs ~ $ _
```

```
andrew@web:~$ ./run_mysql.sh  
MySQL User: andrew  
MySQL Password:  
MySQL Command: SHOW TABLES;  
MySQL DB: wordpress  
Tables_in_wordpress  
wp_cleanup_optimizer_block_range_ip  
wp_cleanup_optimizer_block_single_ip  
wp_cleanup_optimizer_db_scheduler  
wp_cleanup_optimizer_licensing  
wp_cleanup_optimizer_login_log  
wp_cleanup_optimizer_plugin_settings  
wp_cleanup_optimizer_wp_scheduler
```


Chapter 3: Conditions Attached

```
pi@pilabs: ~  
pi@pilabs ~ $ who | grep pi > /dev/null 2>&1 && write pi < message.txt
```

```
Message from pi@pilabs.theurbanpenguin.com on pts/0 at 10:42 ...  
I see you are logged on then!
```

```
EOF
```

```
pi@pilabs ~ $ _
```

```
#!/bin/bash  
echo "You are using $(basename $0)"  
test -z $1 || echo "Hello $1"  
exit 0  
~
```

```
pi@pilabs: ~/bin
```

```
pi@pilabs ~/bin $ hello4.sh  
You are using hello4.sh  
pi@pilabs ~/bin $ hello4.sh bob  
You are using hello4.sh  
Hello bob  
pi@pilabs ~/bin $ _
```

```
#!/bin/bash  
# Welcome script to display a message to users on login  
# Author: @theurbanpenguin  
# Date: 1/1/1971  
if [ $# -lt 1 ] ; then  
    echo "Usage: $0 <name>"  
    exit 1  
fi  
echo "Hello $1"  
exit 0
```

```
pi@pilabs ~ $ hello5.sh
Usage: /home/pi/bin/hello5.sh <name>
pi@pilabs ~ $ hello5.sh fred
Hello fred
pi@pilabs ~ $ _
```

```
if condition ; then
    statement 1
    statement 2
fi
```

```
if condition ; then
    statement
else
    statement
fi
```

```
pi@pilabs ~ $ backup2.sh
Choose H, M or L compression M_
```

```
pi@pilabs ~/bin $ grade.sh Bob b
Bob is a star pupil
pi@pilabs ~/bin $ grade.sh Bob D
Bob needs to try a little harder!
pi@pilabs ~/bin $ grade.sh Bob e
Bob could do a lot better next year
pi@pilabs ~/bin $ grade.sh Bob 5
Grade could not be evaluated for Bob 5
pi@pilabs ~/bin $ _
```

```
pi@pilabs ~ $ search.sh /etc/ntp.conf ^server c
Counting the matches in /etc/ntp.conf of ^server
4
pi@pilabs ~ $ _
```

Chapter 4: Creating Code Snippets

```
pi@pilabs: ~  
set showmode nohlsearch  
set autoindent tabstop=4  
set expandtab  
syntax on  
abbr _sh #!/bin/bash  
  
pi@pilabs ~ $ echo -e "\033[31mError\033[0m"  
Error
```

```
pi@pilabs: ~/bin  
#!/bin/bash  
# Welcome script to display a message to users on login  
# Author: @theurbanpenguin  
# Date: 1/1/1971  
source $HOME/snippets/color  
if [ $# -lt 1 ] ; then  
    echo -e "${RED}Usage: $0 <name>$RESET"  
    exit 1  
fi  
echo -e "${GREEN>Hello $1$RESET"  
exit 0  
  
pi@pilabs ~/bin $ hello7.sh fred  
Hello fred  
pi@pilabs ~/bin $ hello7.sh  
Usage: /home/pi/bin/hello7.sh <name>  
pi@pilabs ~/bin $ _
```

Chapter 5: Alternative Syntax

```
pi@pilabs ~ $ type -a [  
[ is a shell builtin  
[ is /usr/bin/[  
pi@pilabs ~ $ .....
```

```
pi@pilabs ~ $ FILE=/etc/hosts  
pi@pilabs ~ $ [ -f $FILE -a -r $FILE ] && cat $FILE  
127.0.0.1      localhost  
::1           localhost ip6-localhost ip6-loopback  
fe00::0       ip6-localnet  
ff00::0       ip6-mcastprefix  
ff02::1       ip6-allnodes  
ff02::2       ip6-allrouters
```

```
#127.0.1.1     pilabs.theurbanpenguin.com  
pi@pilabs ~ $ _
```

```
pi@pilabs ~ $ echo "I am using $0 with the options: $-"  
I am using -bash with the options: himBH  
pi@pilabs ~ $ _
```

```
pi@pilabs ~ $ hello8.sh  
Hello Anonymous  
pi@pilabs ~ $ hello8.sh fred  
Hello fred  
pi@pilabs ~ $ cat bin/hello8.sh  
#!/bin/bash  
name=${1-"Anonymous"}  
echo "Hello $name"  
exit 0  
pi@pilabs ~ $ _
```

```
pi@pilabs ~ $ FILE="my file"  
pi@pilabs ~ $ [ -f $FILE -a -r $FILE ] && cat $FILE  
-bash: [: too many arguments  
pi@pilabs ~ $ .....
```

```
pi@pilabs ~ $ FILE="my file"
pi@pilabs ~ $ [ -f "$FILE" -a -r "$FILE" ] && cat "$FILE"
The File Contents
pi@pilabs ~ $ _
```

```
pi@pilabs ~ $ FILE="my.pl"
pi@pilabs ~ $ [[ $FILE =~ \.pl$ ]] && echo "Perl found"
Perl found
pi@pilabs ~ $ FILE="my.apl"
pi@pilabs ~ $ [[ $FILE =~ \.pl$ ]] && echo "Perl found"
pi@pilabs ~ $ _
```

```
#!/bin/bash
# Welcome script to display a message to users on login
# Author: @theurbanpenguin
# Date: 1/1/1971
shopt -s nocasematch #turn off case sensitivity
read -p "Type color or mono for script output: "
if [[ $REPLY =~ colou?r ]] ; then
    source $HOME/snippets/color
fi
#Where parameters are not set the display will be mono
echo -e "${GREEN}This is $0 $RESET"
shopt -u nocasematch #reset case sensitivity
exit 0
```

```
pi@pilabs ~/bin $ COUNT=1
pi@pilabs ~/bin $ (( COUNT++ ))
pi@pilabs ~/bin $ echo $COUNT
2
pi@pilabs ~/bin $ _
```

```
pi@pilabs ~/bin $ COUNT=10
pi@pilabs ~/bin $ (( COUNT-- ))
pi@pilabs ~/bin $ (( COUNT > 1 )) && echo "Count is greater than 1"
Count is greater than 1
pi@pilabs ~/bin $ _
```


Chapter 6: Iterating with Loops

```
pi@pilabs ~ $ sudo -i
[sudo] password for pi:
root@pilabs:~# for u in bob joe ; do
> useradd $u
> echo "$u:Password1" | chpasswd
> passwd -e $u
> done
passwd: password expiry information changed.
passwd: password expiry information changed.
root@pilabs:~#
```

```
pi@pilabs ~/bin $ for f in ba* ; do
> stat "$f"
> done
  File: `backup2.sh'
  Size: 675          Blocks: 8          IO Block: 4096   regular file
Device: b302h/45826d Inode: 270110       Links: 1
Access: (0755/-rwxr-xr-x) Uid: ( 1000/   pi)   Gid: ( 1000/   pi)
Access: 2015-07-17 14:00:04.119477594 +0000
Modify: 2015-07-17 14:00:04.119477594 +0000
Change: 2015-07-17 14:00:04.139477463 +0000
Birth: -
  File: `backup.sh'
  Size: 775          Blocks: 8          IO Block: 4096   regular file
Device: b302h/45826d Inode: 268466       Links: 1
Access: (0755/-rwxr-xr-x) Uid: ( 1000/   pi)   Gid: ( 1000/   pi)
Access: 2015-07-04 19:56:11.481438080 +0000
Modify: 2015-07-04 19:56:11.481438080 +0000
Change: 2015-07-04 19:56:11.491438018 +0000
Birth: -
```

```
pi@pilabs ~/bin $ for user in $(who | cut -f1 -d" "); do
> lsof -u "$user" -a -c bash | grep cwd
> done
bash    14935    pi  cwd    DIR  179,2    4096 268409 /home/pi/bin
bash    15140    pi  cwd    DIR  179,2    4096 268409 /home/pi/bin
pi@pilabs ~/bin $
```

```
pi@pilabs ~/bin $
pi@pilabs ~/bin $ hello9.sh fred bob
You are using hello9.sh
Hello fred
Hello bob
pi@pilabs ~/bin $
```



```
pi@pilabs ~ $ for f in * ; do
> [ -d "$f" ] && break
> done
pi@pilabs ~ $ echo "We have found a directory: $f"
We have found a directory: bin
pi@pilabs ~ $

pi@pilabs ~ $ for f in * ; do [ -d "$f" ] || continue
> dir_list="$dir_list $f"
> done
pi@pilabs ~ $ echo "$dir_list"
bin python_games snippets
pi@pilabs ~ $ _

pi@pilabs ~ $ COUNT=10
pi@pilabs ~ $ while (( COUNT >= 0 )) ; do
> echo -e "$COUNT \c"
> (( COUNT-- ))
> done ; echo
10 9 8 7 6 5 4 3 2 1 0
pi@pilabs ~ $ _
```

```
pi@pilabs ~ $
pi@pilabs ~ $ cat servers.txt
8.8.8.8
8.8.4.4
pi@pilabs ~ $ while read server ; do
> ping -c1 "$server" && servers_up="$servers_up $server"
> done < servers.txt
PING 8.8.8.8 (8.8.8.8) 56(84) bytes of data.
64 bytes from 8.8.8.8: icmp_req=1 ttl=52 time=25.1 ms

--- 8.8.8.8 ping statistics ---
1 packets transmitted, 1 received, 0% packet loss, time 0ms
rtt min/avg/max/mdev = 25.187/25.187/25.187/0.000 ms
PING 8.8.4.4 (8.8.4.4) 56(84) bytes of data.
64 bytes from 8.8.4.4: icmp_req=1 ttl=52 time=24.9 ms

--- 8.8.4.4 ping statistics ---
1 packets transmitted, 1 received, 0% packet loss, time 0ms
rtt min/avg/max/mdev = 24.950/24.950/24.950/0.000 ms
pi@pilabs ~ $ echo "These servers are up $servers_up"
These servers are up 8.8.8.8 8.8.4.4
pi@pilabs ~ $ _
```

```
pi@pilabs ~/bin $ ping_server_from_file.sh servers.txt
PING 8.8.8.8 (8.8.8.8) 56(84) bytes of data.
64 bytes from 8.8.8.8: icmp_req=1 ttl=52 time=24.5 ms

--- 8.8.8.8 ping statistics ---
1 packets transmitted, 1 received, 0% packet loss, time 0ms
rtt min/avg/max/mdev = 24.578/24.578/24.578/0.000 ms
PING 8.8.4.4 (8.8.4.4) 56(84) bytes of data.
64 bytes from 8.8.4.4: icmp_req=1 ttl=52 time=24.5 ms

--- 8.8.4.4 ping statistics ---
1 packets transmitted, 1 received, 0% packet loss, time 0ms
rtt min/avg/max/mdev = 24.521/24.521/24.521/0.000 ms
The following servers are up on 29/08/15
Server up: 8.8.8.8
Server up: 8.8.4.4
pi@pilabs ~/bin $ _
```

```
Choose an item: a,b or c
a: Backup
b: Display Calendar
c: Exit
```

```
#!/bin/bash
# Author: @theurbanpenguin
# Web: www.theurbapenguin.com
# Sample menu
# Last Edited: August 2015

while true
do
    clear
    echo "Choose an item: a,b or c"
    echo "a: Backup"
    echo "b: Display Calendar"
    echo "c: Exit"
    read -sn1
    case "$REPLY" in
        a) tar -czvf $HOME/backup.tgz ${HOME}/bin;;
        b) cal;;
        c) exit 0;;
    esac
    read -n1 -p "Press any key to continue"
done
```

Chapter 7: Creating Building Blocks with Functions

```
pi@pilabs ~ $ declare -F
declare -f __expand_tilde_by_ref
declare -f __get_cword_at_cursor_by_ref
declare -f __git_aliased_command
declare -f __git_aliases
declare -f __git_complete_file
declare -f __git_complete_remote_or_refspec
declare -f __git_complete_revlist
declare -f __git_complete_revlist_file
declare -f __git_complete_strategy

pi@pilabs ~ $ type quote
quote is a function
quote ()
{
    local quoted=${1//\'/\'\\\'\\'};
    printf "'%s'" "$quoted"
}

'pi'pi@pilabs ~ $ quote $USER
'pi'pi@pilabs ~ $ _
```

```
pi@pilabs ~ $ type show_system
show_system is a function
show_system ()
{
    echo "The uptime is:";
    uptime;
    echo;
    echo "CPU Detail";
    lscpu;
    echo;
    echo "User list";
    who
}
```

```
pi@pilabs ~ $ function clean_file { sed -i.bak '/^\s*#/d;/^$/d' "$1"; }
pi@pilabs ~ $ cd
pi@pilabs ~ $ cp /etc/ntp.conf $HOME
pi@pilabs ~ $ wc -l $HOME/ntp.conf
55 /home/pi/ntp.conf
pi@pilabs ~ $ clean_file $HOME/ntp.conf
pi@pilabs ~ $ wc -l $HOME/ntp.conf
13 /home/pi/ntp.conf
pi@pilabs ~ $ wc -l $HOME/ntp.conf.bak
55 /home/pi/ntp.conf.bak
pi@pilabs ~ $ _
```

```
pi@pilabs ~ $ clean.sh
Enter a file to clean: ntp.cff
ntp.cff does not seem to be a file
pi@pilabs ~ $ .....
```

```
pi@pilabs ~ $ clean.sh
Enter a file to clean: ntp.conf
The file ntp.conf starts with 55 ntp.conf
The file ntp.conf is now 13 ntp.conf
pi@pilabs ~ $ _
```

Choose an item: a,b or c

a: Backup

b: Display Calendar

c: Exit

September 2015

Mo 7 14 21 28

Tu 1 8 15 22 29

We 2 9 16 23 30

Th 3 10 17 24

Fr **4** 11 18 25

Sa 5 12 19 26

Su 6 13 20 27

 36 37 38 39 40

Press any key to continue_

Chapter 8: Introducing sed

```
pi@pilabs ~ $ ifconfig eth0 | grep "RX packets"
RX packets:689830 errors:0 dropped:5 overruns:0 frame:0
pi@pilabs ~ $ _
```

```
pi@pilabs ~ $ grep "$USER" /etc/passwd
pi:x:1000:1000:,,,:/home/pi:/bin/bash
pi@pilabs ~ $ .....
```

```
pi@pilabs ~ $ read -p "Enter a user name: "
Enter a user name: pi
pi@pilabs ~ $ if (grep "$REPLY" /etc/passwd > /dev/null ); then
> echo "The user $REPLY exists"
> exit 1
> fi
The user pi exists
exit
pi@pilabs ~ $ _
```

```
pi@pilabs ~ $ grep -c name /proc/cpuinfo
4
```

```
[pi@black-pearl ~ ]$ grep -c name /proc/cpuinfo
1
[pi@black-pearl ~ ]$ _
```

```
[pi@black-pearl ~ ]$ bash
[pi@black-pearl ~ ]$ CPU_CORES=$(grep -c name /proc/cpuinfo)
[pi@black-pearl ~ ]$ if (( CPU_CORES < 4 )); then
> echo "A mininum a 4 cores are required"
> exit 1
> fi
A mininum a 4 cores are required
exit
[pi@black-pearl ~ ]$ .....
```

```
[pi@black-pearl ~ ]$ type check_cores
check_cores is a function
check_cores ()
{
    [ -z $1 ] && REQ_CORES=2;
    CPU_CORES=$(grep -c name /proc/cpuinfo);
    if (( CPU_CORES < REQ_CORES )); then
        echo "A minimum a $REQ_CORES cores are required";
        exit 1;
    fi
}
[pi@black-pearl ~ ]$ ..
```

```
[pi@black-pearl ~ ]$ check_cores 1
[pi@black-pearl ~ ]$ check_cores
A minimum a 2 cores are required
exit
[pi@black-pearl ~ ]$ _
```

```
#!/bin/bash
OLDIFS="$IFS"
IFS=","
while read product price quantity
do
    echo -e "\033[1;34m$product \
===== \033[0m\n\
Price : \t $price \n\
Quantity : \t $quantity \n"
done < "$1"
IFS=$OLDIFS
```



```
pi@pilabs ~/bin $ parsecsv.sh tools
drill =====
      Price :          99
      Quantity :        5

hammer =====
      Price :          10
      Quantity :       50

brush =====
      Price :           5
```

```
pi@pilabs ~/bin $ parsecsv.sh tools | grep -A2 hammer
hammer =====
      Price :          10
      Quantity :       50
pi@pilabs ~/bin $ ..
```

```
[pi@black-pearl ~ ]$ echo -e "color\ncolour" | grep -E 'colou?r'
color
colour
[pi@black-pearl ~ ]$ ..
```

```
pi@pilabs ~/bin $ grep -E '[aeoiu]{4}' /usr/share/dict/words
gooier
gooiest
onomatopoeia
onomatopoeia's
plateaued
plateauing
queue
queue's
queued
queues
queuing
pi@pilabs ~/bin $ ..
```

```
pi@pilabs ~ $ sed -n '/^root/ p' /etc/passwd
root:x:0:0:root:/root:/bin/bash
pi@pilabs ~ $ _
```

```
pi@pilabs ~ $ sed -n '/^pi/ s/bash/sh/p' /etc/passwd
pi:x:1000:1000:,,,:/home/pi:/bin/sh
pi@pilabs ~ $ _
```

```
pi@pilabs ~ $ sed -i '/^pi/ s@bash@sh@pg' "$HOME/passwd"
```

Chapter 9: Automating Apache Virtual Hosts

```
#<VirtualHost *:80>
#   ServerAdmin webmaster@dummy-host.example.com
#   DocumentRoot /www/docs/dummy-host.example.com
#   ServerName dummy-host.example.com
#   ErrorLog logs/dummy-host.example.com-error_log
#   CustomLog logs/dummy-host.example.com-access_log common
#</VirtualHost>
```

1003

```
#<VirtualHost *:80>
```

1004

```
#   ServerAdmin webmaster@dummy-host.example.com
```

1005

```
#   DocumentRoot /www/docs/dummy-host.example.com
```

1006

```
#   ServerName dummy-host.example.com
```

1007

```
#   ErrorLog logs/dummy-host.example.com-error_log
```

1008

```
#   CustomLog logs/dummy-host.example.com-access_log common
```

1009

```
#</VirtualHost> _
```

```
[root@apache ~]# sed -n '1003,1009 p' httpd.conf
```

```
#<VirtualHost *:80>
```

```
#   ServerAdmin webmaster@dummy-host.example.com
```

```
#   DocumentRoot /www/docs/dummy-host.example.com
```

```
#   ServerName dummy-host.example.com
```

```
#   ErrorLog logs/dummy-host.example.com-error_log
```

```
#   CustomLog logs/dummy-host.example.com-access_log common
```

```
#</VirtualHost>
```

```
[root@apache ~]# █
```

```
[root@apache ~]# sed -n '/^#<VirtualHost/,/^#\</VirtualHost/p' httpd.conf
```

```
#<VirtualHost *:80>
```

```
#   ServerAdmin webmaster@dummy-host.example.com
```

```
#   DocumentRoot /www/docs/dummy-host.example.com
```

```
#   ServerName dummy-host.example.com
```

```
#   ErrorLog logs/dummy-host.example.com-error_log
```

```
#   CustomLog logs/dummy-host.example.com-access_log common
```

```
#</VirtualHost>
```

```
[root@apache ~]# █
```

```
/^#<VirtualHost/,/^#<\/VirtualHost/ {  
s/^#//  
w template.txt  
}
```

```
[root@apache ~]# cat template.txt  
<VirtualHost *:80>  
    ServerAdmin webmaster@dummy-host.example.com  
    DocumentRoot /www/docs/dummy-host.example.com  
    ServerName dummy-host.example.com  
    ErrorLog logs/dummy-host.example.com-error_log  
    CustomLog logs/dummy-host.example.com-access_log common  
</VirtualHost>  
[root@apache ~]#
```

```
[root@apache bin]# cat vhost.sh  
#!/bin/bash  
WEBDIR=/www/docs  
CONFDIR=/etc/httpd/conf.d  
TEMPLATE=$HOME/template.txt  
[ -d $CONFDIR ] || mkdir -p $CONFDIR  
sed s/dummy-host.example.com/$1/ $TEMPLATE > $CONFDIR/$1.conf  
mkdir -p $WEBDIR/$1  
echo "New site for $1" > $WEBDIR/$1/index.html
```

```
[root@apache bin]#
```

```
[root@apache bin]# cat /etc/httpd/conf.d/sales.example.com.conf  
<VirtualHost *:80>  
    ServerAdmin webmaster@sales.example.com  
    DocumentRoot /www/docs/sales.example.com  
    ServerName sales.example.com  
    ErrorLog logs/sales.example.com-error_log  
    CustomLog logs/sales.example.com-access_log common  
</VirtualHost>  
[root@apache bin]#
```

```
[root@apache bin]# ./vhost2.sh marketing.example.com  
Do you want to restrict access to this site? y/n y  
Which network should we restrict access to: 192.168.0.0/24  
[root@apache bin]#
```

```
[root@apache bin]# cat /etc/httpd/conf.d/marketing.example.com.conf
<VirtualHost *:80>
    ServerAdmin webmaster@marketing.example.com
    DocumentRoot /www/docs/marketing.example.com
    ServerName marketing.example.com
    ErrorLog logs/marketing.example.com-error_log
    CustomLog logs/marketing.example.com-access_log common
<Directory /www/docs/marketing.example.com >
    Order allow,deny
    Allow from 127.0.0.1
    Allow from 192.168.0.0/24
</Directory>
</VirtualHost>
```

```
[root@apache bin]# cat vhost2.sh
#!/bin/bash
WEBDIR=/www/docs/$1
CONFDIR=/etc/httpd/conf.d
CONFFILE=$CONFDIR/$1.conf
TEMPLATE=$HOME/template.txt
[ -d $CONFDIR ] || mkdir -p $CONFDIR
sed s/dummy-host.example.com/$1/ $TEMPLATE > $CONFFILE
mkdir -p $WEBDIR
echo "New site for $1" > $WEBDIR/index.html
read -p "Do you want to restrict access to this site? y/n "
[ $REPLY = 'n' ] && exit 0
read -p "Which network should we restrict access to: " NETWORK
sed -i "/<\VirtualHost>/i <Directory $WEBDIR >\
    \n Order allow,deny\
    \n Allow from 127.0.0.1\
    \n Allow from $NETWORK\
\n</Directory>" $CONFFILE
```

Chapter 10: Awk Fundamentals

```
pi@pilabs ~ $ awk ' BEGIN { print "Hello World!" } '  
Hello World!  
pi@pilabs ~ $ _
```

```
pi@pilabs ~ $ awk ' BEGIN { FS=":" }  
{ print NR,$1 }  
END { print "Total:",NR } ' /etc/passwd  
1 root  
2 daemon  
3 bin  
4 sys  
5 sync
```

```
pi@pilabs ~ $ awk ' END { print NR } ' /etc/passwd  
28
```

```
pi@pilabs ~ $ wc -l /etc/passwd  
28 /etc/passwd
```

```
pi@pilabs ~ $ awk '/bash$/ ' /etc/passwd  
root:x:0:0:root:/root:/bin/bash  
pi:x:1000:1000:,,,:/home/pi:/bin/bash  
bob:x:1001:1004::/home/bob:/bin/bash  
joe:x:1002:1005::/home/joe:/bin/bash  
pi@pilabs ~ $ _
```

```
pi@pilabs ~ $ awk ' BEGIN { FS=":" } { print $1,$3,$7 } ' /etc/passwd  
root 0 /bin/bash  
daemon 1 /bin/sh  
bin 2 /bin/sh  
sys 3 /bin/sh  
sync 4 /bin/sync
```

```

pi@pilabs ~ $ awk ' BEGIN { FS=":" }
{ printf "%10s %4d %17s\n", $1, $3, $7 } ' /etc/passwd
    root      0          /bin/bash
  daemon     1          /bin/sh
    bin       2          /bin/sh
    sys       3          /bin/sh
  sync       4          /bin/sync
  games      5          /bin/sh
    man       6          /bin/sh

```

```

pi@pilabs ~ $ awk 'BEGIN {FS=":";printf "%10s %4s %17s\n","Name","UID","Shell" }
{ printf "%10s %4d %17s\n", $1, $3, $7 } ' /etc/passwd
    Name  UID      Shell
    root   0        /bin/bash
  daemon  1        /bin/sh
    bin    2        /bin/sh
    sys    3        /bin/sh

```

```

pi@pilabs ~ $ awk 'function green(s) {
printf "\033[1;32m" s "\033[0m\n"
}
BEGIN {FS=":";
green("    Name:    UID:    Shell:") }
{ printf "%10s %4d %17s\n", $1, $3, $7 } ' /etc/passwd
    Name:    UID:    Shell:
    root     0        /bin/bash
  daemon    1        /bin/sh
    bin      2        /bin/sh
    sys      3        /bin/sh

```

Chapter 11: Summarizing Logs with Awk

```
root@andrew-15-10:~# tail /var/log/apache2/access.log
127.0.0.1 - - [12/Oct/2015:09:48:42 +0100] "GET / HTTP/1.1" 200 3525 "-" "Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:41.0) Gecko/20100101 Firefox/41.0"
127.0.0.1 - - [12/Oct/2015:09:48:43 +0100] "GET /icons/ubuntu-logo.png HTTP/1.1" 200 3689 "http://localhost/" "Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:41.0) Gecko/20100101 Firefox/41.0"
127.0.0.1 - - [12/Oct/2015:09:48:43 +0100] "GET /favicon.ico HTTP/1.1" 404 500 "-" "Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:41.0) Gecko/20100101 Firefox/41.0"
127.0.0.1 - - [12/Oct/2015:09:48:43 +0100] "GET /favicon.ico HTTP/1.1" 404 500 "-" "Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:41.0) Gecko/20100101 Firefox/41.0"
```

```
root@andrew-15-10:~# awk ' { print $4, $5 } ' /var/log/apache2/access.log
[12/Oct/2015:09:48:42 +0100]
[12/Oct/2015:09:48:43 +0100]
[12/Oct/2015:09:48:43 +0100]
[12/Oct/2015:09:48:43 +0100]
```

```
pi@pilabs ~/bin $ awk '( $4 ~ /10\/Sep\/2014/ )' access.log | less
128.252.139.84 - - [10/Sep/2014:00:00:03 +0100] "GET /wp/?cat=281 HTTP/1.1" 200
51860 "http://theurbanpenguin.com/wp/?cat=281" "Mozilla/5.0 (Macintosh; Intel Mac OS X 10_8_5) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/36.0.1985.125 Safari/537.36"
41.150.168.184 - - [10/Sep/2014:00:00:23 +0100] "GET /scripting/java.html HTTP/1
```

```
$ awk '( $4 ~ /10\/Sep\/2014/ ) { print $1; COUNT++ } END { print COUNT } ' access.log
```

```
pi@pilabs ~/bin $ awk '( $4 ~ /10\/Sep\/2014/ ) { COUNT++ } END { print COUNT } ' access.log
16205
pi@pilabs ~/bin $ _
```

```
pi@pilabs ~/bin $ awk ' ( $9 ~ /404/ ) { print $9 } ' access.log _
```

```
pi@pilabs ~/bin $ awk -f status.awk access.log
200 has occurred 23825 times.
206 has occurred 48 times.
301 has occurred 60 times.
302 has occurred 21 times.
304 has occurred 2273 times.
403 has occurred 133 times.
404 has occurred 4382 times.
501 has occurred 63 times.
pi@pilabs ~/bin $ _
```



```
pi@pilabs ~/bin $ awk -f 404.awk access.log
404/old/wp-admin/ has occurred 2 times.
404/monitor.html has occurred 1 times.
404/windows.html has occurred 1 times.
404/novell.html has occurred 1 times.
404/user/ has occurred 2 times.
404/linux.html has occurred 1 times.
404/zcm10.html has occurred 1 times.
```

```
pi@pilabs ~/bin $
pi@pilabs ~/bin $ awk -f ip.awk access.log
121. .52.100 has accessed 12 times.
pi@pilabs ~/bin $ _
```

```
pi@pilabs ~/bin $ awk -f browser.awk access.log
"DoCoMo/2.0 has accessed 7 times.
"com.apple.WebKit.WebContent/10600.1.15 has accessed 4 times.
"Xenu has accessed 3 times.
"-" has accessed 90 times.
"PHP/5.3.14" has accessed 1 times.
"FeedBot" has accessed 8 times.
"OpenOffice/4.1.0" has accessed 91 times.
"facebookexternalhit/1.1 has accessed 11 times.
"Mozilla/5.0 has accessed 7 times.
"Feed has accessed 48 times.
"msnbot-UDiscovery/2.0b has accessed 9 times.
"Twitterbot/1.0" has accessed 94 times.
"AdsBot-Google has accessed 10 times.
"Python-urllib/1.17" has accessed 1 times.
"HTTP_Request2/2.1.1 has accessed 4 times.
"Mozilla/4.0 has accessed 1713 times.
```

```
root@andrew-15-10:~# awk '( $7 ~ /^to/ )' /var/log/mail.log
Oct 12 17:00:47 andrew-15-10 postfix/local[10109]: 80346680E8: to=<root@andrew-15-10>, relay=local, delay=0.14, delays=0.09/0.05/0/0.01, dsn=2.0.0, status=sent (delivered to mailbox)
root@andrew-15-10:~# █
```

Chapter 12: A Better lastlog with Awk

```
gdm                **Never logged in**
sshd               **Never logged in**
tcpdump           **Never logged in**
tux                pts/1    localhost        Tue Oct 20 13:02:35 +0100 2015
bob               **Never logged in**
u1                **Never logged in**
vboxadd           **Never logged in**
```

```
centos6 ~ $ lastlog -u 500-5000
Username      Port    From          Latest
tux           pts/1   localhost     Tue Oct 20 13:02:35 +0100 2015
bob           **Never logged in**
u1           **Never logged in**
centos6 ~ $ █
```

```
centos6 ~ $ lastlog | awk -f ll.awk
tux                pts/1    localhost        Tue Oct 20 13:02:35 +0100 2015
=====
Total Number of Users Processed:  1
centos6 ~ $ █
```

```
centos6 ~ $ lastlog | awk -f ll.awk
tux                pts/1    192.168.0.3     Thu Oct 22 13:31:04 +0100 2015
bob                tty1                    Thu Oct 22 13:34:48 +0100 2015
=====
Total Number of Users Processed:  2
centos6 ~ $ █
```

```
centos6 ~ $ lastlog | awk -f ll.awk
Username  Login date
=====
          tux 22 Oct 2015
          bob 22 Oct 2015
=====
Total Number of Users Processed:  2
centos6 ~ $ █
```

```
centos6 ~ $ awk -f vh.awk search=packt virtualhost.conf
<VirtualHost *:80>
DocumentRoot /www/packt
ServerName www.packtpub.com
# Other directives here
</VirtualHost>
centos6 ~ $ █
```

```
centos6 ~ $ awk -f catalog.awk search=drill catalog.xml
name: drill price: 99 stock: 5
centos6 ~ $ █
```

Chapter 13: Using Perl as a Bash Scripting Alternative

```
pi@pilabs ~ $ perl -v
```

```
This is perl 5, version 14, subversion 2 (v5.14.2) built  
-thread-multi-64int  
(with 89 registered patches, see perl -V for more detail)
```

```
pi@pilabs ~/bin $ ./args.pl fred  
You are using ./args.pl  
Hello fred  
pi@pilabs ~/bin $ _
```

```
pi@pilabs ~/bin $ ./forargs.pl fred bob  
You are using ./forargs.pl  
You have supplied 2 arguments  
Hello fred  
Hello bob  
pi@pilabs ~/bin $ _
```

```
pi@pilabs ~/bin $ ./array.pl  
You are using ./array.pl  
Fred Bloggs is 24  
pi@pilabs ~/bin $ _
```

```
pi@pilabs ~/bin $ ./ifargs.pl  
You are using ./ifargs.pl  
pi@pilabs ~/bin $ ./ifargs.pl fred  
You are using ./ifargs.pl  
You have supplied 1 arguments  
Hello fred  
pi@pilabs ~/bin $ _
```

Chapter 14: Using Python as a Bash Scripting Alternative

```
pi@pilabs ~/bin $ python3
Python 3.2.3 (default, Mar  1 2013, 11:53:50)
[GCC 4.6.3] on linux2
Type "help", "copyright", "credits" or "license" for more information.
>>> ..
```

```
>>> import sys
>>> sys.version
'3.2.3 (default, Mar  1 2013, 11:53:50) \n[GCC 4.6.3]'
>>> _
```

```
>>> import this
The Zen of Python, by Tim Peters
```

```
Beautiful is better than ugly.
Explicit is better than implicit.
Simple is better than complex.
Complex is better than complicated.
Flat is better than nested.
Sparse is better than dense.
Readability counts.
Special cases aren't special enough to break the rules.
Although practicality beats purity.
Errors should never pass silently.
Unless explicitly silenced.
In the face of ambiguity, refuse the temptation to guess.
There should be one-- and preferably only one --obvious way to do it.
Although that way may not be obvious at first unless you're Dutch.
Now is better than never.
Although never is often better than *right* now.
If the implementation is hard to explain, it's a bad idea.
If the implementation is easy to explain, it may be a good idea.
Namespaces are one honking great idea -- let's do more of those!
>>> _
```

```
pi@pilabs ~/bin $ ./args.py fred
Hello fred
pi@pilabs ~/bin $ _
```

```
pi@pilabs ~/bin $ ./args.py fred
Hello fred
2
pi@pilabs ~/bin $ .....
```

```
pi@pilabs ~/bin $ ./args.py
Traceback (most recent call last):
  File "./args.py", line 3, in <module>
    print("Hello " + sys.argv[1] + " " + str(len(sys.argv)))
IndexError: list index out of range
pi@pilabs ~/bin $ _
```

```
pi@pilabs ~/bin $ ./args.py
Exiting ./args.py
pi@pilabs ~/bin $ ./args.py fred
Arguments supplied: 2
Hello fred
Exiting ./args.py
```

```
#!/usr/bin/python3
import sys
count = len(sys.argv)
name = ''

if ( count == 1 ):
    name = input("Enter a name: ")
else:
    name = sys.argv[1]

print("Hello " + name)
print("Exiting " + sys.argv[0])
```

```
#!/usr/bin/python3
import sys
count = len(sys.argv)
name = ''

if ( count == 1 ):
    name = input("Enter a name: ")
else:
    name = sys.argv[1]

log = open("/tmp/script.log", "a")
log.write("Hello " + name + "\n")
log.close()
```