Using Aliases in Raspbian

Aliases are a way of assinging a complex command to an easy to remember word. Looking at the command below:

```
     ip addr |grep '[0-9]\{1,3\}\.[0-9]\{1,3\}\.[0-9]\{1,3\}\.[0-9]\{1,3\}'
```

This returns just the IP part of the output from an ***ip addr*** command, making the output much easier to read, however remembering and typing the command requires more effort than it saves. For this then we can create an alias.

I want an alias called **showip** that acutally does the same as typing ip addr |grep '[0-9]\{1,3\}\.[0-9]\{1,3\}\.[0-9]\{1,3\}\.[0-9]\{1,3\}'

You need to edit the .bash file for the logged in user who will use this alias. I am logged in as root, so I will edit that bash file.\
Navigate to root:

```
  cd /root
```

Show the list of files:

```
  ls -al to show file list, you will see a list of files, including one called .bashrc\\
```

Edit the .bashrc file by entering:

```
   sudo nano /root/.bashrc
```

My default .bashrc file looks like this:

```
# ~/.bashrc: executed by bash(1) for non-login shells.
# see /usr/share/doc/bash/examples/startup-files (in the package bash-doc)
# for examples
# If not running interactively, don't do anything
case $- in
    *i*) ;;
      *) return;;
esac
# don't put duplicate lines or lines starting with space in the history.
# See bash(1) for more options
HISTCONTROL=ignoreboth
# append to the history file, don't overwrite it
shopt -s histappend
# for setting history length see HISTSIZE and HISTFILESIZE in bash(1)
HISTSIZE=1000
HISTFILESIZE=2000
# check the window size after each command and, if necessary,
# update the values of LINES and COLUMNS.
shopt -s checkwinsize
# If set, the pattern "**" used in a pathname expansion context will
# match all files and zero or more directories and subdirectories.
#shopt -s globstar
# make less more friendly for non-text input files, see lesspipe(1)
#[ -x /usr/bin/lesspipe ] && eval "$(SHELL=/bin/sh lesspipe)"
# set variable identifying the chroot you work in (used in the prompt below)
if [ -z "${debian_chroot:-}" ] && [ -r /etc/debian_chroot ]; then
    debian_chroot=$(cat /etc/debian_chroot)
fi
# set a fancy prompt (non-color, unless we know we "want" color)
case "$TERM" in
    xterm-color) color_prompt=yes;;
esac
# uncomment for a colored prompt, if the terminal has the capability; turned
# off by default to not distract the user: the focus in a terminal window
# should be on the output of commands, not on the prompt
force_color_prompt=yes
if [ -n "$force_color_prompt" ]; then
```

```
      if [ -x /usr/bin/tput ] && tput setaf 1 >&/dev/null; then
          # We have color support; assume it's compliant with Ecma-48
          # (ISO/IEC-6429). (Lack of such support is extremely rare, and such
          # a case would tend to support setf rather than setaf.)
          color_prompt=yes
      else
          color_prompt=
      fi
  fi
  if [ "$color_prompt" = yes ]; then
      PS1='${debian_chroot:+($debian_chroot)}\[\033[01;32m\]\u@\h\[\033[00m\]:\[\033[01;34m\]\w
\$\[\033[00m\] '
  else
      PS1='${debian_chroot:+($debian_chroot)}\u@\h:\w\$ '
  fi
  unset color_prompt force_color_prompt
  # If this is an xterm set the title to user@host:dir
  case "$TERM" in
  xterm*|rxvt*)
      PS1="\[\e]0;${debian_chroot:+($debian_chroot)}\u@\h: \w\a\]$PS1"
      ;;
  *)
      ;;
  esac
  # enable color support of ls and also add handy aliases
  if [ -x /usr/bin/dircolors ]; then
      test -r ~/.dircolors && eval "$(dircolors -b ~/.dircolors)" || eval "$(dircolors -b)"
      alias ls='ls --color=auto'
      #alias dir='dir --color=auto'
      #alias vdir='vdir --color=auto'
      alias grep='grep --color=auto'
      alias fgrep='fgrep --color=auto'
      alias egrep='egrep --color=auto'
  fi
  # colored GCC warnings and errors
  #export GCC_COLORS='error=01;31:warning=01;35:note=01;36:caret=01;32:locus=01:quote=01'
  # some more ls aliases
  #alias ll='ls -l'
  #alias la='ls -A'
  #alias l='ls -CF'
  # Alias definitions.
  # You may want to put all your additions into a separate file like
  # ~/.bash_aliases, instead of adding them here directly.
  # See /usr/share/doc/bash-doc/examples in the bash-doc package.
  if [ -f ~/.bash_aliases ]; then
      . ~/.bash_aliases
  fi
  # enable programmable completion features (you don't need to enable
  # this, if it's already enabled in /etc/bash.bashrc and /etc/profile
  # sources /etc/bash.bashrc).
  if ! shopt -oq posix; then
    if [ -f /usr/share/bash-completion/bash_completion ]; then
      . /usr/share/bash-completion/bash_completion
    elif [ -f /etc/bash_completion ]; then
      . /etc/bash_completion
    fi
  fi
```

Now add the new alias:

```
      alias showip=" ip addr |grep '[0-9]\{1,3\}\.[0-9]\{1,3\}\.[0-9]\{1,3\}\.[0-9]\{1,3\}'"
```

save the file and exit:

```
      ctrl-x - yes - enter
```

You have to restart the bash for any new aliases to work

```
. .bashrc
```

If you want to see if the alias is now listed, from the command prompt just type:

```
alias
```

You should now be able to type your new command at the command line and get an output:

```
showip
    inet 127.0.0.1/8 scope host lo
    inet 192.168.27.30/24 brd 192.168.27.255 scope global eth0
```

From:
https://wiki.alanwalker.uk/ - **WalkerWiki - wiki.alanwalker.uk**

Permanent link:
https://wiki.alanwalker.uk/doku.php?id=using_aliases_in_raspbian&rev=1469445265

Last update: **2023/03/09 22:35**