My actual ~/.bashrc with:

- Alias
- Completions
- Functions
- Fun Prompt

You need:

- bash
- ccze

```
#
# ~/.bashrc
#

# If not running interactively, don't do anything
[[ $- != *i* ]] && return

# include the global system file
if [ -f /etc/bashrc ]; then
    . /etc/bashrc
fi

# Add some color schema
if [[ "$TERM" != "linux" && -f ~/.dircolors.256colors ]]; then
    eval $(dircolors ~/.dircolors.256colors)
elif [[ -f ~/.dircolors ]]; then
    eval $(dircolors ~/.dircolors)
fi

# Add some nice folder in the path
PATH=$PATH:~/bin
export PATH

# Default editor
export EDITOR=vim

# Bash options
shopt -s checkwinsize     # check the window size after each command and, if
necessary, update the values of LINES and COLUMNS.
shopt -s cdspell          # autocorrects cd misspellings
shopt -s dotglob          # include dotfiles in pathname expansion
shopt -s autocd               # auto "cd" to a directory

# don't put duplicate lines and same sucessive entries in the history.
export HISTCONTROL=ignoreboth

# Colorized less command
export LESS_TERMCAP_mb=$'\E[01;31m' # begin blinking
export LESS_TERMCAP_md=$'\E[01;31m' # begin bold
```

```
export LESS_TERMCAP_me=$'\E[0m'     # end mode
export LESS_TERMCAP_se=$'\E[0m'     # end standout-mode
export LESS_TERMCAP_so=$'\E[01;44m' # begin standout-mode â€“ info
export LESS_TERMCAP_ue=$'\E[0m'     # end underline
export LESS_TERMCAP_us=$'\E[32m'    # begin underline


# Overload actual commands
alias grep='grep --color=auto'
alias dmesg='dmesg -HL'
alias vi='vim'
alias ls='ls -G --color'

# New aliases
alias fifo="echo $1 > ~/FIFO"
alias lc="ls -al | tr -s \" \" | cut -d\" \" -f3,4,9 | column -t"
alias ll="ls -altrh | tr -s \" \" | cut -d\" \" -f6,7,8,3,4,5,9 | column -t"
alias learning="whatis \$(compgen -c) 2>/dev/null | sort | less"
alias da='date "+%A, %B %d, %Y [%T]"'
alias horloge="watch -n1 -t \"date '+%H:%M:%S' | toilet -f future\""

## FileSystem Aliases ##
alias lsl='ls -lh | ccze -A'          # list files, dirs
alias usage='du -ch | grep total'     # size of current dir
alias dut="du -h -c -d 1"             # lst with size of elements in current
dir
alias size='sudo du -hsx * | sort -rh | head -10 | ccze -A'         # list
10 dir/files by size
alias sizevar='sudo du -a /var | sort -n -r | head -n 10 | ccze -A' # list
10 dir by size
alias sizehome="sudo du -a ~/ | sort -n -r | head -n 10 | ccze -A"  # list
10 dir by size
alias lsbig="echo listing files & directories by size | pv -qL 10 && ls -
lSrh | ccze -A"    # sort files dirs by size
alias listmod="ls -ltr | ccze -A"     # list modified files
alias listf="ls -F | ccze -A"         # list alphabeticaly
alias findit="sudo find / -name"      # find files
alias upd60="find . -mmin -60"        # updated files currentdir for last
60mins
alias rtupd60=" find / -mtime -1"     # updated files in / for last 60mins

## Network Aliases
alias netlisten='lsof -i -P | grep LISTEN | ccze -A'   #listening ports
alias nstat="netstat -p TCP -eWc | ccze -A"           #netstat
alias nstato="netstat -tuael --numeric-hosts --numeric-ports | ccze -A"
#netstat
alias ports='netstat -tulanp | ccze -A'  #netstat
alias speeddown="echo Displaying Download Speed Graph| pv -qL 10 &&
speedometer -rx eth0"
alias speedup="echo Displaying Upload Speed Graph | pv -qL 10 && speedometer
-tx eth0"
```

```
alias speedrw="echo Show R/W speeds | pv -qL 10 && dd bs=1000000 count=100
if=/dev/zero of=testfile & speedometer testfile"

## System Aliases ##
alias h='history | ccze -A'           # Nice history
alias sourcebash="source ~/.bashrc"   # source .bashrc
alias inxif="inxi -F | ccze -A"       # System information
# Drives
alias uuid="echo â†’ Listing this system \($(hostname)\) UUID | pv -qL 10 &&
ls /dev/disk/by-uuid/ -alh | ccze -A" # UUID list
alias drives="echo â†’ Listing connected drives on \($(hostname)\) | pv -qL
10 && lsblk -f | ccze -A"              # list hdds, uuid's
# Systemd
alias kernelmsg="sudo journalctl -f _TRANSPORT=kernel | ccze -A"
# kernel messages
alias bootmsg="echo â†’ Boot Messages | pv -qL 10 && sudo journalctl -b |
ccze -A"  # boot messages
alias systemdmsg="sudo journalctl /usr/lib/systemd/systemd | ccze -A"
# dmesg
alias blame="systemd-analyze blame | ccze -A"
# WTF slowed down the boot
alias boot="echo â†’ Boot Time | pv -qL 10 && systemd-analyze | ccze -A"
# Boot time
alias units="echo â†’ Listing Units (Systemd) | pv -qL 10 && systemctl list-
units | ccze -A" # List of Units
alias errors="echo â†’ Systemd Journal Errors | pv -qL 10 && journalctl -b -
p err | ccze -A" # Errors
# Ram
alias meminfo='echo "RAM Information   " | pv -qL 10 &&free -m -l -t | ccze
-A'                    # RAM information
alias psmem='echo "Top Processes accesing RAM  " | pv -qL 10 && ps auxf |
sort -nr -k 4 | ccze -A' # Process eating RAM
# CPU
alias cpuinfo='lscpu | ccze -A'
#CPU info
alias pscpu='echo "Top Processes accesing CPU  " | pv -qL 10 && ps auxf |
sort -nr -k 3 | ccze -A' # Process eating CPU

# Terminal Clock
alias tclock="while sleep 1;do tput sc;tput cup 0 $(($(tput
cols)-29));date;tput rc;done &"

## Xrandr Aliases ##
# Switch Single head
alias singlehead='xrandr --output DP3 --off --output DP2 --off --output DP1
--off --output HDMI3 --off --output HDMI2 --off --output HDMI1 --off --
output LVDS1 --mode 1366x768 --pos 0x0 --rotate normal --primary'
# Switch Dual head (screen left of)
alias dualleft='xrandr --output DP3 --off --output DP2 --off --output DP1 --
off --output HDMI3 --off --output HDMI2 --off --output HDMI1 --off --output
LVDS1 --mode 1366x768 --pos 1920x592 --rotate normal --output VGA1 --mode
```

```
1920x1080 --pos 0x0 --rotate normal --primary'

alias FvwmRestart='killall trayer ; FvwmCommand Restart'

# User specific Aliases and functions
function remove_crap() { rm -r $@; }     # Removes everything passed as
argument recursively
function add_txt_ext() { mv $@ $@.txt; } # Add .txt extention to file passed
as argument
# copy a file and change its filename to filename-timestamp
function fbackup() { cp -iv $1{,.bkp.$(date -d @`stat -c %Z $1` +%Y%m%d-
%H%M%S)}; }
# Remove offending key line $1
function rmkeyssh() { sed -i "${1}d" ~/.ssh/known_hosts; }

## DICTIONARY FUNCTIONS ##
dwordnet () { curl dict://dict.org/d:${1}:wn; }
dacron () { curl dict://dict.org/d:${1}:vera; }
djargon () { curl dict://dict.org/d:${1}:jargon; }
dfoldoc () { curl dict://dict.org/d:${1}:foldoc; }
dthesaurus () { curl dict://dict.org/d:${1}:moby-thes; }

## Website Commands ##
cmdfu() { curl "http://www.commandlinefu.com/commands/matching/$(echo "$@" |
sed 's/ /-/g')/$(echo -n $@ | base64)/plaintext" ;}
down4me() { curl -s "http://www.downforeveryoneorjustme.com/$1" | sed '/just
you/!d;s/<[^>]*>//g';}

# Swap 2 files around, if they exist (from Uzi's bashrc).
function swap() {
 # Create temp file
 local TMPFILE=tmp.$$
 # Check availability of files
 [ $# -ne 2 ] && echo "swap: 2 arguments needed" && return 1
 [ ! -e $1 ] && echo "swap: $1 does not exist" && return 1
 [ ! -e $2 ] && echo "swap: $2 does not exist" && return 1
 # Move it
 mv "$1" $TMPFILE
 mv "$2" "$1"
 mv $TMPFILE "$2"
}

# Handy Extract Program
function extract() {
    if [ -f "$1" ] ; then
        case "$1" in
            *.tar.bz2)   tar xvjf "$1"     ;;
            *.tar.gz)    tar xvzf "$1"     ;;
            *.bz2)       bunzip2 "$1"      ;;
            *.rar)       unrar x "$1"      ;;
            *.gz)        gunzip "$1"       ;;
```

```
            *.tar)       tar xvf "$1"      ;;
            *.tbz2)      tar xvjf "$1"     ;;
            *.tgz)       tar xvzf "$1"     ;;
            *.zip)       unzip "$1"        ;;
            *.Z)         uncompress "$1"   ;;
            *.7z)        7z x "$1"         ;;
            *)           echo "'$1' cannot be extracted via >extract<" ;;
        esac
    else
        echo "'$1' is not a valid file!"
    fi
}

# Creates an archive (*.tar.gz) from given directory.
function maketar() { tar cvzf "${1%%/}.tar.gz"  "${1%%/}/"; }

# Create a ZIP archive of a file or folder.
function makezip() { zip -r "${1%%/}.zip" "$1" ; }


# Alias root to ssh as root directly
alias root='ssh -l root'
# tab completion for ssh hosts
SSH_COMPLETE=( $(cat ~/.ssh/known_hosts | \
cut -f 1 -d ' ' | \
sed -e s/,.*//g | \
uniq | \
egrep -v ^[0123456789]) )
complete -o default -W "${SSH_COMPLETE[*]}" ssh
complete -o default -W "${SSH_COMPLETE[*]}" root

# Auto-complete bticket with the ticket numbers
TICKET_COMPLETE=( $(/usr/bin/ls -1 ~/Bacula/ticket) )
complete -o default -W "${TICKET_COMPLETE[*]}" bticket

# auto completion for sudo whereis and man
complete -cf sudo
complete -cf whereis
complete -cf man

battery_status() {

  BATTERY=/sys/class/power_supply/BAT0
  CHARGE=`cat $BATTERY/capacity`
  BATSTATE=`cat $BATTERY/status`

  # Colors for humans
  NON='\033[00m'
  BLD='\033[01m'
  RED='\033[01;31m'
  GRN='\033[01;32m'
```

```
  YEL='\033[01;33m'

  COLOUR="$RED"

  case "${BATSTATE}" in
   'Charged')
           BATSTT="$BLD=$NON"
           ;;
   'Charging')
           BATSTT="$BLD+$NON"
           ;;
   'Discharging')
           BATSTT="$BLD-$NON"
           ;;
  esac

  # prevent a charge of more than 100% displaying
  if [ "$CHARGE" -gt "99" ]
    then
      CHARGE=100
  fi

  # prevent an error if the battery is not in the laptop (e.g. you have two
and take out the primary)
  STATE=`cat $BATTERY/present`
  if [ "$STATE" == '0' ]
    then
      echo -e "${RED}nobat"
      exit
  fi

  if [ "$CHARGE" -gt "15" ]
    then
      COLOUR="$YEL"
  fi

  if [ "$CHARGE" -gt "30" ]
    then
      COLOUR="$GRN"
  fi
  echo -e "${BATSTT}${COLOUR}${CHARGE}%${NON}"
}

##################################################
# Fancy PWD display function
##################################################
# The home directory (HOME) is replaced with a ~
# The last pwdmaxlen characters of the PWD are displayed
# Leading partial directory names are striped off
# /home/me/stuff          -> ~/stuff               if USER=me
# /usr/share/big_dir_name -> ../share/big_dir_name if pwdmaxlen=20
```

```
################################################
bash_prompt_command() {
    # How many characters of the $PWD should be kept
    local pwdmaxlen=35
    # Indicate that there has been dir truncation
    local trunc_symbol=".."
    local dir=${PWD##*/}
    pwdmaxlen=$(( ( pwdmaxlen < ${#dir} ) ? ${#dir} : pwdmaxlen ))
    NEW_PWD=${PWD/#$HOME/\~}
    local pwdoffset=$(( ${#NEW_PWD} - pwdmaxlen ))
    if [ ${pwdoffset} -gt "0" ]
    then
        NEW_PWD=${NEW_PWD:$pwdoffset:$pwdmaxlen}
        NEW_PWD=${trunc_symbol}/${NEW_PWD#*/}
    fi
}

function draw_line(){
  # Function that creates a line terminal wide

  # Sepratаor
  #local DOTS="â‰¡"

  for (( c=1; c<=$(tput cols); c++ ))
  do
    DOTS=$(echo -n $DOTS"â‰¡")
  done
}

function load_out() {
  echo -n "$(uptime | sed -e "s/.*load average: \(.*\...\), \(.*\...\),
\(.*\...\).*/\1/" -e "s/ //g")"
}

function load_color() {
  # Colour progression is important ...
  #   bold gray -> bold green -> bold yellow -> bold red ->
  #   black on red -> bold white on red
  #
  # Then we have to choose the values at which the colours switch, with
  # anything past yellow being pretty important.
  # Colors are defined in the load function / blinking for load > 2

  tmp=$(echo $(load_out)*100 | bc)
  let load100=${tmp%.*}

  if [ ${load100} -lt 70 ]
  then
    printf "${EMK}"
  elif [ ${load100} -ge 70 ] && [ ${load100} -lt 120 ]
  then
```

```
    printf "${EMG}"
  elif [ ${load100} -ge 120 ] && [ ${load100} -lt 200 ]
  then
    printf "${EMY}"
  elif [ ${load100} -ge 200 ] && [ ${load100} -lt 300 ]
  then
    printf "${BLINK}${EMR}"
  elif [ ${load100} -ge 300 ] && [ ${load100} -lt 500 ]
  then
    printf "${BLINK}${EMK}${BGR}"
  else
    printf "${BLINK}${EMW}${BGR}"
  fi
}

function load {

    case $TERM in
     xterm*|rxvt*)
         local TITLEBAR='\[\033]0;\u:${NEW_PWD}\007\]'
          ;;
     *)
         local TITLEBAR=""
          ;;
    esac

    local NONE="\[\033[0m\]"    # unsets color to term's fg color
    local BLINK="\[\033[5m\]"    # Blink text

    # regular colors
    local K="\[\033[0;30m\]"    # black
    local R="\[\033[0;31m\]"    # red
    local G="\[\033[0;32m\]"    # green
    local Y="\[\033[0;33m\]"    # yellow
    local B="\[\033[0;34m\]"    # blue
    local M="\[\033[0;35m\]"    # magenta
    local C="\[\033[0;36m\]"    # cyan
    local W="\[\033[0;37m\]"    # white

    # empahsized (bolded) colors
    local EMK="\[\033[1;30m\]"
    local EMR="\[\033[1;31m\]"
    local EMG="\[\033[1;32m\]"
    local EMY="\[\033[1;33m\]"
    local EMB="\[\033[1;34m\]"
    local EMM="\[\033[1;35m\]"
    local EMC="\[\033[1;36m\]"
    local EMW="\[\033[1;37m\]"

    # background colors
    local BGK="\[\033[40m\]"
```

```
    local BGR="\[\033[41m\]"
    local BGG="\[\033[42m\]"
    local BGY="\[\033[43m\]"
    local BGB="\[\033[44m\]"
    local BGM="\[\033[45m\]"
    local BGC="\[\033[46m\]"
    local BGW="\[\033[47m\]"

    local UC=$W                 # user's color
    [ $UID -eq "0" ] && UC=$R   # root's color

#  PS1="#\! âˆ´ ${EMG}\D{%H:%M:%S}${NONE} âˆ´
${UC}\u${EMK}@${EMG}\h${NONE}:$(tty) ${EMB} ${NEW_PWD}${NONE}\nâˆµ |
\[$(load_color)\]\$(load_out)${NONE} | ${EMK}â†’${W}>${EMW}>${NONE} "
#  PS1="${EMK}${DOTS}${NONE}\n\342\224\214\342\225\274[ \$(if [[ \$? == 0
]]; then printf '${EMG}'\342\234\223'${NONE}'; else printf
'${EMR}'\342\234\227'${NONE}'; fi)\
# âˆ´ \# âˆ´ ${EMG}\D{%H:%M:%S}${NONE} âˆ´
${UC}\u${EMK}@${EMG}\h${NONE}:$(tty)]${EMB} ${NEW_PWD}${NONE}\n\
#\342\224\224\342\224\200\342\224\200\342\225\274[
\[$(load_color)\]\$(load_out)${NONE} ] ${EMK}â†’${W}>${EMW}>${NONE} "

  PS1="${EMK}${DOTS}${NONE}\n\342\224\214\342\225\274 \$(if [[ \$? == 0 ]];
then printf '${EMG}'\342\234\223'${NONE}'; else printf
'${EMR}'\342\234\227'${NONE}'; fi) âˆ´ \# âˆ´ ${EMG}\D{%H:%M:%S}${NONE} âˆ´
\$(battery_status) âˆ´ ${UC}\u${EMK}@${EMG}\h${NONE}:$(tty)${EMB}
${NEW_PWD}${NONE}\n\342\224\224\342\224\200\342\224\200\342\225\274
$(load_color)\$(load_out)${NONE}  ${EMK}â†’${W}>${EMW}>${NONE} "

  unset DOTS


}

PROMPT_COMMAND="history -a; history -c; history -
r;bash_prompt_command;draw_line;load_color;load"
```

The prompt will looks like:

```
================================================================================
================================================================================
┌─ ✓ ∴ 66 ∴ 12:01:09 ∴ 99% ∴ warnaud@Yoru:/dev/pts/1 ~/Documents
└─ 0.48  →>>
```

You need the ~/.dircolors and ~/.dircolors.256colors files

From:
https://wiki.fortier-family.com/ - **Warnaud's Wiki**

Permanent link:
**https://wiki.fortier-family.com/dotfiles/bashrc**

Last update: **2020/12/17 05:23**