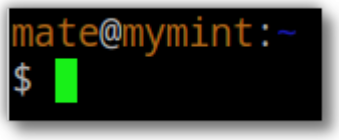
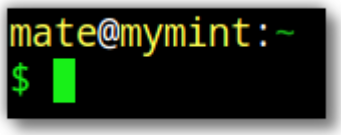
PS1

simples




```
export PS1="\[\e]0;\u@\h
\w\a\]\[\e[00;33m\]\u\[\e[0m\]\[\e[00;37m\]@\[\e[0m\]\[\e[00;33m\]\h\[\e[0m\]\[\e[0
0;37m\]:\[\e[0m\]\[\e[00;34m\]\w\[\e[0m\]\[\e[00;37m\]\n\$ \[\e[0m\]"
```




```
export PS1="\[$(tput sgr0)\]\033[38;5;15m\033[38;5;11m\u\[$(tput
sgr0)\]\033[38;5;15m\033[38;5;15m@\[$(tput
sgr0)\]\033[38;5;15m\033[38;5;11m\h\[$(tput
sgr0)\]\033[38;5;15m\033[38;5;15m:\[$(tput
sgr0)\]\033[38;5;15m\033[38;5;2m\w\[$(tput
sgr0)\]\033[38;5;15m\033[38;5;15m\n\[$(tput sgr0)\]\\$\[$(tput sgr0)\] \[$(tput
sgr0)\]"
```

soporte GIT

/vía: https://stackoverflow.com/questions/4133904/ps1-line-with-git-current-branch-and-colors

```
function we_are_in_git_work_tree {
    git rev-parse --is-inside-work-tree &> /dev/null
}

function parse_git_branch {
    if we_are_in_git_work_tree
    then
    local BR=$(git rev-parse --symbolic-full-name --abbrev-ref HEAD 2> /dev/null)
    if [ "$BR" == HEAD ]
    then
        local NM=$(git name-rev --name-only HEAD 2> /dev/null)
        if [ "$NM" != undefined ]
        then echo -n "@$NM"
        else git rev-parse --short HEAD 2> /dev/null
        fi
    else
        echo -n $BR
    fi
    fi
```

```
}

function parse_git_status {
    if we_are_in_git_work_tree
    then
    local ST=$(git status --short 2> /dev/null)
    if [ -n "$ST" ]
    then echo -n " + "
    else echo -n " - "
    fi
    fi
}

function pwd_depth_limit_2 {
    if [ "$PWD" = "$HOME" ]
    then echo -n "~"
    else pwd | sed -e "s|.*/\(.*/.*\)|\1|"
    fi
}

COLBROWN="\[\033[1;33m\]"
COLRED="\[\033[1;31m\]"
COLCLEAR="\[\033[0m\]"

# export all these for subshells
export -f parse_git_branch parse_git_status we_are_in_git_work_tree
pwd_depth_limit_2
export PS1="$COLRED<$COLBROWN
\$(pwd_depth_limit_2)$COLRED\$(parse_git_status)$COLBROWN\$(parse_git_branch)
$COLRED>$COLCLEAR "
export TERM="xterm-color"
```

ejemplos y fuentes de ideas

funciones: https://medium.freecodecamp.org/bash-shortcuts-to-enhance-your-git-workflow-5107d64ea0ff

__git_ps1()

```
export PS1="\n\[\033[1;30m\][$$:$PPID - \j:\!\[\033[1;30m\]]\[\033[0;36m\] \T
\
\[\033[1;30m\][\[\033[1;34m\]\u@\H\[\033[1;30m\]:\[\033[0;37m\]${SSH_TTY:-o} \
\[\033[0;32m\]+${SHLVL}\[\033[1;30m\]]
\[\033[1;37m\]\w\[\033[0;37m\]\[\033[1;34m\]\$(__git_ps1 \" (%s)\")
\[\033[0;37m\] \n\$ "
```

gestión de colores

```
set_prompt()
{
   local last_cmd=$?
   local txtreset='$(tput sgr0)'
   local txtbold='$(tput bold)'
   local txtblack='$(tput setaf 0)'
   local txtred='$(tput setaf 1)'
```

```
   local txtgreen='$(tput setaf 2)'
   local txtyellow='$(tput setaf 3)'
   local txtblue='$(tput setaf 4)'
   local txtpurple='$(tput setaf 5)'
   local txtcyan='$(tput setaf 6)'
   local txtwhite='$(tput setaf 7)'
   # unicode "□"
   local fancyx='\342\234\227'
   # unicode "□"
   local checkmark='\342\234\223'
   # Line 1: Full date + full time (24h)
   # Line 2: current path
   PS1="\[$txtbold\]\[$txtwhite\]\n\D{%A %d %B %Y
%H:%M:%S}\n\[$txtgreen\]\w\n"
   # User color: red for root, yellow for others
   if [[ $EUID == 0 ]]; then
       PS1+="\[$txtred\]"
   else
       PS1+="\[$txtyellow\]"
   fi
   # Line 3: user@host
   PS1+="\u\[$txtwhite\]@\h\n"
   # Line 4: a red "□" or a green "□" and the error number
   if [[ $last_cmd == 0 ]]; then
      PS1+="\[$txtgreen\]$checkmark \[$txtwhite\](0)"
   else
      PS1+="\[$txtred\]$fancyx \[$txtwhite\]($last_cmd)"
   fi
   # Line 4: green git branch
   PS1+="\[$txtgreen\]$(__git_ps1 ' (%s)')\[$txtwhite\]"
   # Line 4: good old prompt, $ for user, # for root
   PS1+=" \\$ "
}
PROMPT_COMMAND='set_prompt'
```

From:
https://matebcn.eu/ - **miguel angel torres egea**

Permanent link:
https://matebcn.eu/linux:bash:ps1

Last update: **04/09/2018 14:59**