

nom.sh

```

# fonctions
#####
# find from name in current directory
function ff() { find . -iname '*'$*'*' -ls | $PAGER; }

# generate a dated .bak from file
function bak() { cp $1 $1_`date +%Y-%m-%d_%H:%M:%S`.bak ; }

# infos -----
-
# generate space report
function space() { du -skh * | sort -hr ; }
# disk usage
function dduse() { echo -e "`df -h / | grep dev | awk '{print $5}'`
used -- `df -h / | grep dev | awk '{print $4}'` free" ; }
# mem usage
function mmuse() { echo -e "`free -m | grep Mem | awk '{print $3}'`M
used -- `free -m | grep Mem | awk '{print $7}'`M free" ; }
# processes
function my_ps() { ps @$ -u $USER -o pid,%cpu,%mem,bsdtime,command ; }
function pp() { my_ps f | awk '!/awk/ && $0~var' var=${1:-"."} ; }
# hardware -----
-
# processor
function core() { cat /proc/cpuinfo | grep "model name" | cut -c14- |
head -n 1 ; }
# graphic card
function graph() { lspci | grep -i vga | cut -d: -f3 ; }
# ethernet card
function ethcard() { lspci | grep -i ethernet | cut -d: -f3 ; }
# wireless card
function wfcard() { lspci | grep -i wireless | cut -d: -f3 ; }
# public ip address -----
-
function my_eip()
{
    if [ "$(cat /sys/class/net/enp0s25/operstate)" = "up" ] || [ "$(cat
/sys/class/net/wls1/operstate)" = "up" ];then
        MY_EXIP=$(wget -q -O - checkip.dyndns.org | sed -e
's/[^\[:digit:]\|\.]//g')
    else
        MY_EXIP=$(echo "not connected")
    fi
    # output
    echo -e " $MY_EXIP"
}
# infobox -----
-
function ii()

```

```

{
    clear
    echo
    echo -e ""
    echo -e "${cyan}    nakedeb Debian InfoBox"
    echo -e "    -----$NC"
    echo -e "${green} agenda$NC"
    echo -e "    `date +%A, %B %-d, %Y -- %I:%M %P`"
    echo -e "${red} processor information$NC"
    echo -e "    `core`"
    echo -e "${magenta} graphic information$NC"
    echo -e "    `graph`"
    echo -e "${blue} ethernet information$NC"
    echo -e "    `ethcard`"
    echo -e "${blue} wireless information$NC"
    echo -e "    `wfcad`"
    echo ""
    echo -e "${orange} kernel information$NC"
    echo -e "    `uname -a`"
    echo -e "${yellow} machine stats$NC"
    echo -e "    `uptime`"
    echo -e "${yellow} memory stats$NC"
    echo -e "    `mmuse`"
    echo -e "${yellow} disk stats$NC"
    echo -e "    `dduse`"
    echo -e "${yellow} external IP address$NC"
    echo -e "    `my_eip`"
    echo -e ""
    echo -e "${red} if R.Stallman was here...$NC"
    echo -e "`vrms`"
    echo ""
}

```

```

# archives -----
-
# extract
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          ;;

```

```
        *.xz)          unxz $1          ;;
        *)             echo "'$1' cannot be extracted via >extract<"
;;
    esac
    else
        echo "'$1' is not a valid file"
    fi
}

# compress
mktar() { tar cvf "${1%}/.tar" "${1%}/"; }
mktgz() { tar cvzf "${1%}/.tar.gz" "${1%}/"; }
mktbz() { tar cvjf "${1%}/.tar.bz2" "${1%}/"; }
mktxz() { tar cvJf "${1%}/.tar.xz" "${1%}/"; }
mkzip() { zip -r "${1%}/.zip" "${1%}/"; }

# bds .cbr archives
mkcbr() { for d in */; do zip -r "${d%}/.cbr" "$d"; done; }

# cli colors preview
function clipv()
{
    for i in {0..255} ; do
        printf "\x1b[48;5;%sm%3d\e[0m " "$i" "$i"
        if (( i == 15 )) || (( i > 15 )) && (( (i-15) % 6 == 0 )); then
            printf "\n";
        fi
    done
}
```

From:

<http://debian-facile.org/> - **Documentation - Wiki**

Permanent link:

http://debian-facile.org/utilisateurs:arpinix:config:bash_functions

Last update: **27/01/2024 22:12**

