#!/bin/bash

case $newlang in

en\_US)

a\_enter=' ⇢ Press enter to continue'

a\_back=' ⇢ RETURN'

a\_selection\_fun=' ⇢ Select an option:'

;;

esac

# varibles

export \_hora=$(printf '%(%H:%M:%S)T')

export \_fecha=$(printf '%(%D)T')

export numero='^[0-9]+$'

export texto='^[A-Za-z]+$'

export txt\_num='^[A-Za-z0-9]+$'

# 🔹color interfaces🔹colored interfaces

msg() {

COLOR[0]='\033[1;37m' #BRAN='\033[1;37m'

COLOR[1]='\e[31m' #RED='\e[31m'

COLOR[2]='\e[32m' #GREEN='\e[32m'

COLOR[3]='\e[33m' #YELLOW='\e[33m'

COLOR[4]='\e[34m' #BLUE='\e[34m'

COLOR[5]='\e[91m' #MAGENTA='\e[35m'

COLOR[6]='\033[1;97m' #MAG='\033[1;36m'

COLOR[7]='\e[36m' #teal='\e[36m'

COLOR[8]='\e[30m' #negro='\e[30m'

COLOR[9]='\033[34m' #blue='\033[1;34m'

NEGRITO='\e[1m'

SEMCOR='\e[0m'

case $1 in

-ne) cor="${COLOR[1]}${NEGRITO}" && echo -ne "${cor}${2}${SEMCOR}" ;;

-nazu) cor="${COLOR[6]}${NEGRITO}" && echo -ne "${cor}${2}${SEMCOR}" ;;

-nverd) cor="${COLOR[2]}${NEGRITO}" && echo -ne "${cor}${2}${SEMCOR}" ;;

-nama) cor="${COLOR[3]}${NEGRITO}" && echo -ne "${cor}${2}${SEMCOR}" ;;

-ama) cor="${COLOR[3]}${NEGRITO}" && echo -e "${cor}${2}${SEMCOR}" ;;

-verm) cor="${COLOR[3]}${NEGRITO}[!] ${COLOR[1]}" && echo -e "${cor}${2}${SEMCOR}" ;;

-verm2) cor="${COLOR[1]}${NEGRITO}" && echo -e "${cor}${2}${SEMCOR}" ;;

-verm3) cor="${COLOR[1]}" && echo -e "${cor}${2}${SEMCOR}" ;;

-teal) cor="${COLOR[7]}${NEGRITO}" && echo -e "${cor}${2}${SEMCOR}" ;;

-teal2) cor="${COLOR[7]}" && echo -e "${cor}${2}${SEMCOR}" ;;

-blak) cor="${COLOR[8]}${NEGRITO}" && echo -e "${cor}${2}${SEMCOR}" ;;

-blak2) cor="${COLOR[8]}" && echo -e "${cor}${2}${SEMCOR}" ;;

-azu) cor="${COLOR[6]}${NEGRITO}" && echo -e "${cor}${2}${SEMCOR}" ;;

-blu) cor="${COLOR[9]}${NEGRITO}" && echo -e "${cor}${2}${SEMCOR}" ;;

-blu1) cor="${COLOR[9]}" && echo -e "${cor}${2}${SEMCOR}" ;;

-verd) cor="${COLOR[2]}${NEGRITO}" && echo -e "${cor}${2}${SEMCOR}" ;;

-bra) cor="${COLOR[0]}${NEGRITO}" && echo -e "${cor}${2}${SEMCOR}" ;;

# -bar) cor="${COLOR[1]}===============================================" && echo -e "${SEMCOR}${cor}${SEMCOR}" ;;

-bar) cor="${COLOR[1]}┌──────────────────────────────────────────────┐" && echo -e "${SEMCOR}${cor}${SEMCOR}" ;;

-bar0) cor="${COLOR[1]}└──────────────────────────────────────────────┘" && echo -e "${SEMCOR}${cor}${SEMCOR}" ;;

-bar2) cor="${COLOR[7]}===============================================" && echo -e "${SEMCOR}${cor}${SEMCOR}" ;;

#-bar3) cor="${COLOR[1]}-----------------------------------------------" && echo -e "${SEMCOR}${cor}${SEMCOR}" ;;

-bar3) cor="${COLOR[1]} ──────────────────────────────────────────────" && echo -e "${SEMCOR}${cor}${SEMCOR}" ;;

-bar4) cor="${COLOR[7]}-----------------------------------------------" && echo -e "${SEMCOR}${cor}${SEMCOR}" ;;

esac

}

# 🔹text centering

print\_center() {

if [[ -z $2 ]]; then

text="$1"

else

col="$1"

text="$2"

fi

while read line; do

unset space

x=$(((45 - ${#line}) / 2))

for ((i = 0; i < $x; i++)); do

space+=' '

done

space+="$line"

if [[ -z $2 ]]; then

msg -azu "$space"

else

msg "$col" "$space"

fi

done <<<$(echo -e "$text")

}

print\_title() {

if [[ -z $2 ]]; then

text="$1"

else

col="$1"

text="$2"

fi

while read line; do

unset space

x=$(((53 - ${#line}) / 2))

for ((i = 0; i < $x; i++)); do

space+=' '

done

space+="$line"

if [[ -z $2 ]]; then

msg -azu "$space"

else

msg "$col" "$space"

fi

done <<<$(echo -e "$text")

}

sub\_title() {

if [[ -z $2 ]]; then

text="$1"

else

col="$1"

text="$2"

fi

while read line; do

unset space

x=$(((54 - ${#line}) / 2))

for ((i = 0; i < $x; i++)); do

space+=' '

done

space+="$line"

if [[ -z $2 ]]; then

msg -azu "$space"

else

msg "$col" "$space"

fi

done <<<$(echo -e "$text")

}

# 🔹titles and headings

title() {

clear

msg -bar

if [[ -z $2 ]]; then

print\_center -blu1 "$1"

else

print\_center "$1" "$2"

fi

msg -bar0

}

main\_title() {

clear

msg -bar

if [[ -z $2 ]]; then

print\_title -blu1 "$1"

else

print\_title "$1" "$2"

fi

msg -bar0

}

title\_sub() {

clear

msg -bar

if [[ -z $2 ]]; then

sub\_title -blu1 "$1"

else

sub\_title "$1" "$2"

fi

msg -bar0

}

# completion of tasks🔹home

enter() {

msg -bar

text="►► ${a\_enter:-Press enter to continue} ◄◄"

if [[ -z $1 ]]; then

print\_center -ama "$text"

else

print\_center "$1" "$text"

fi

read

}

# option, return back/back🔹option, go back go back

back() {

msg -bar

echo -ne "$(msg -verd " [0]") $(msg -verm2 ">") " && msg -bra "\033[1;41m${a\_back:-GO BACK}"

msg -bar0

}

exit2home() {

msg -bar

echo -ne "$(msg -verd " [0]") $(msg -verm2 ">") " && msg -bra "\033[1;41m${a\_back:-EXIT}"

msg -bar0

}

inst() {

print\_center -verm2 "${a9:-❗ Please goto Github and reinstall the tool}"

print\_center -verm2 "${a9:-❗ Please goto Github and reinstall the tool}"

print\_center -verm2 "${a9:-❗ Please goto Github and reinstall the tool}"

}

# menu maker (options 1, 2, 3,...)

menu\_func() {

local options=${#@}

local array

for ((num = 1; num <= $options; num++)); do

echo -ne "$(msg -verd " [$num]") $(msg -verm2 ">") "

array=(${!num})

case ${array[0]} in

"-vd") echo -e "\033[1;33m[!]\033[1;32m ${array[@]:1}" ;;

"-vm") echo -e "\033[1;33m[!]\033[1;31m ${array[@]:1}" ;;

"-fi") echo -e "${array[@]:2} ${array[1]}" ;;

-bar | -bar2 | -bar3 | -bar4) echo -e "\033[1;37m${array[@]:1}\n$(msg ${array[0]})" ;;

\*) echo -e "\033[1;37m${array[@]}" ;;

esac

done

}

# 🔹Numerical selection option

function selection\_fun() {

local selection="null"

local range

if [[ -z $2 ]]; then

option=$1

col="-nazu"

else

option=$2

col=$1

fi

# Check if option is too large

if (( option > 13 )); then

echo " ⇢ Option is too large. Please enter a smaller number."

return 1

fi

# Initialize range array

range=( $(seq 0 $option) )

while true; do

# Ask for selection

a\_selection\_fun=" ⇢ Enter your selection: "

msg "$col" "${a\_selection\_fun} " >&2

read selection

# Check if input is within range

if [[ $selection == [0-$option] ]]; then

break

fi

# Handle invalid input

echo "Invalid input, please enter a number between 0 and $option."

tput cuu1 >&2 && tput dl1 >&2

done

echo $selection

}

in\_option() {

unset option

if [[ -z $2 ]]; then

msg -nazu " $1: " >&2

else

msg $1 " $2: " >&2

fi

read option

echo "$option"

}

in\_option\_down() {

dat=$1

length=${#dat}

cal=$((22 - $length / 2))

line=''

for ((i = 0; i < $cal; i++)); do

line+='╼'

done

echo -e " $(msg -verm3 "╭$line╼[")$(msg -azu "$dat")$(msg -verm3 "]")"

echo -ne " $(msg -verm3 "╰╼")\033[37;1m> " && read option

}

del() {

for ((i = 0; i < $1; i++)); do

tput cuu1 && tput dl1

done

}

export -f msg

export -f selection\_fun

export -f menu\_func

export -f print\_center

export -f title

export -f back

export -f inst

export -f enter

export -f in\_option

export -f in\_option\_down

export -f del

export -f exit2home