

SCRIPTING FOR SECURITY

# WHIPTAIL PROGRAM

Eszter Karajz EC1836291



# WHAT IS WHIPTAIL

Whiptail is a program that allows shell scripts to display dialog boxes to the user for informational purposes, or to get input from the user in a friendly way.



## PART 1: CREATE FILES

1.1 Make a directory called menu then inside menu create 3 files:

1) menu.sh

2) fruits.csv

3) vegetables.csv

1.2 Populate csv files – you can put any range of fruits or vegetables into the csv files

```
menu — -zsh — 91x24
eszterkarajz@Eszters-MacBook-Pro ~ % cd Desktop/College/Scripting_for_security
eszterkarajz@Eszters-MacBook-Pro Scripting_for_security % mkdir menu
eszterkarajz@Eszters-MacBook-Pro Scripting_for_security % cd menu
eszterkarajz@Eszters-MacBook-Pro menu % touch menu.sh
eszterkarajz@Eszters-MacBook-Pro menu % cat > fruits.csv
apples,3
pears,4
plums,5
tomatoes,8
eszterkarajz@Eszters-MacBook-Pro menu % cat > vegetables.csv
onions,5
carrots,7
potatoes,9
celery,4
eszterkarajz@Eszters-MacBook-Pro menu %
```

## CREATING FILES

mkdir  
touch  
cat

```
ed Desktop/College/Scripting_for_se
ting_for_security % mkdir menu
ting_for_security % cd menu
% touch menu.sh
% cat > fruits.csv
```

```
% cat > vegetables.csv
```

```
eszterkarajz@Eszters-MacBo
eszterkarajz@Eszters-MacBo
eszterkarajz@Eszters-MacBoo
eszterkarajz@Eszters-MacBook
eszterkarajz@Eszters-MacBook-P
apples,3
pears,4
plums,5
tomatoes,8
eszterkarajz@Eszters-MacBook-Pro m
onions,5
carrots,7
potatoes,9
celery,4
```

## Part 2: Create a whiptail program in the file menu.sh with the following options:



### 2.1

The first screen should ask the user if they would like to proceed or exit the program.



### 2.2

The second screen should present a menu where the user can select whether they want to use grep or awk.



### 2.3

If the user selects awk the menu should take them to a different menu asking for the name of the file they wish to awk.



### 2.4

If the user selects grep the menu should take them to a different menu asking for the name of the file they wish to grep.



## 2.5

The user should be able to input either fruits.csv or vegetables.csv



## 2.6

The menu should awk the second column of the chosen csv and display the result back to the user if awk was selected.



## 2.7

The menu should grep the chosen csv and display the only lines containing the letter 'e' if grep was selected



## 2.8

The program should then loop back to the screen defined in 2.1

# RUN THE PROGRAM

SH MENU.SH

You are presented with the option to  
proceed or exit



# GREP OR AWK

Would you like to use grep or awk?

<Grep>

<Awk>

## IF THE USER SELECTS GREP

Grep filename

Enter the filename you wish to grep: fruits.csv or vegetables.csv

fruits.csv

<Ok>

apples,3  
pears,4  
tomatoes,8

potatoes,9  
celery,4

<Ok>

<Ok>



IF THE USER  
SELECTS AWK

Awk filename

Enter the filename you wish to awk: fruits.csv or vegetables.csv

vegetables.csv

<Ok>

3  
4  
5  
8

<Ok>

5  
7  
9  
4

<Ok>

Awk filename

Enter the filename you wish to awk: fruits.csv or vegetables.csv

fruits.csv

<Ok>

# Whiptail dialog boxes I used



████████████████████

MESSAGE  
BOX

--msgbox

████████████████████

INPUT BOX

--inputbox

████████████████████

YES/NO BOX

--yesno

## --INPUTBOX

The user is able to input a string as an answer.

## --NOCANCEL

The dialog box won't have a Cancel button.

## --TITLE

Specifies a title string to be displayed at the top of the dialog box.

```
function awk_function {  
    choice_awk=$(whiptail --inputbox "Enter the filename you wish to awk:  
    if [ "$choice_awk" == "fruits.csv" ];  
    then  
        fruit_function "awk"  
    else  
        veg_function "awk"  
    fi
```

```
you wish to awk: fruits.csv or vegetables.csv" 10 60 ""
```

```
"" --nocancel --title "Awk filename" 3>&1 1>&2 2>&3);
```

## --YESNO

This dialog box is useful for asking questions that require the user to answer either yes or no. The dialog box has a Yes button and a No button.

## --NO-BUTTON

Sets the text of the No button.

## --YES-BUTTON

Sets the text of the Yes button.

```
function grep_or_awk {  
    if (whiptail --title "Would you like to use grep or awk?"  
        then  
            grep_function  
        else  
            awk_function  
    fi  
}
```

```
--yesno " " --yes-button "Grep" --no-button "Awk" 8 70);
```

```
"Would you like to proceed or exit the program?" --yesno ""
```

```
function veg_function {  
    if [ $1 = "awk" ];  
    then  
        message="$(awk -F ',' '{print $2}' vegetables.csv )"  
    elif [ $1 = "grep" ];  
    then  
        message="$(grep 'e' vegetables.csv )"  
    else  
        echo $1  
        message="ERROR, INVALID CHOICE"  
    fi  
    whiptail --msgbox "$message" 20 50  
}
```

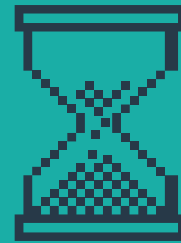
## MESSAGE BOX --MSGBOX

It has only a single OK button. You can use this dialog box to display a message. After reading it, the user can press the ENTER key and whiptail will exit.

```
3 clear
4
5 function fruit_function {
6     if [ $1 = "awk" ];
7     then
8         message="$(awk -F ',' '{print $2}' fruits.csv)"
9     elif [ $1 = "grep" ];
10    then
11        message="$(grep 'e' fruits.csv)"
12    else
13        echo $1
14        message="ERROR, INVALID CHOICE"
15    fi
16    whiptail --msgbox "$message" 20 50
17 }
```



**Message box**  
**--msgbox**



THANK YOU  
FOR  
LISTENING