Script Building Assignment Project Brief

Scenario

In this assignment you are going to build 3 scripts that form a utility that will help you keep your filesystem tidy.

Script 1: cruft_remover.sh

The first script is a great utility that will help you to keep your system clear of unnecessary files (known informally as "cruft") by allowing you to delete files that haven't been modified for a certain period of time.

Script 2: folder_organiser.sh

The second script is another utility that will enable you to organise the files within a given folder into different subfolders based on their file types.

Script 3: toolkit.sh

Then finally you are going to wrap this all together by creating a third script which presents an interactive menu that your user can use to select which utility they want to run.

Your Tasks

Script 1: cruft_remover.sh

Step 1: In this script we want to ask the user which folder they want to remove "cruft" from, and to define how many days files should be unmodified for to be considered "cruft".

Perhaps the `read` command will come in handy?

Step 2: Next you need to create an array of the files that are eligible for removal

To do this you will need to use the `find` command.

You will need to do some research on how to use the find command, so be sure to read the `man` page!

Hint: In particular, check out the `-maxdepth`, `-type`, and `-mtime` options.

Hint: Remember to save the results into an array with the `readarray` command!

Step 3: You then need to iterate over the array you created in the last step, and present the user with an option to delete each file in the array.

Important: **You MUST** use the `-i` option when using the `rm` command so that you get a prompt to confirm whether you want to delete each file before any deletion takes place. After all, you wouldn't want to accidentally delete something important! **Internet of Things Academy Ltd accepts no liability for any deleted files or for any damages.**

Script 2: folder_organiser.sh

Step 1: In this script you are going to use the output of an `ls` command to generate the list of files in a folder that you are going to organise.

As a first step, construct a `while` loop that will iterate over the output of the `ls` command.

Hint: Do you remember process substitution?

Step 2: Use a `case` statement to glob the file names for their file extension and action commands based on their file extension.

For each of the file extensions that you choose to organise, create some logic to Move each file into its appropriate folder, creating that folder if necessary.

Here is where you should move each file type:

File extension	Subfolder
.jpg, .jpeg, .png	images
.doc, .docx, .txt, .pdf	documents
.xls, .xlsx, .csv	spreadsheets
.sh	scripts
.zip, .tar, .tar.gz, .tar.bz2	archives
.ppt, .pptx	presentations
.mp3	audio
.mp4	video
Anything else	Leave in current folder

Hint: You may find it useful to group together multiple extensions in a single case using the | symbol, like we did with the `city_selector` project (the case statement project).

Script 3: toolkit.sh

Step 1: With this script we just want to present the user with a menu where they can choose which of the two scripts they want to run and then have this script run the relevant script based on the user's selection.

Hint: You will need to use a `select` command and `case` statement to create the menu

Extra Guidance

Note: Now that you are writing full scripts, it's important that you start to use the 5 professional components that you learned during the first section of this course to make your scripts as professional as possible.

These are:

1. Author
2. Created Date
3. Last Modified Date
4. Description
5. Usage Information