#Start the script.
#!/bin/bash

#Create three directories: `Images`, `Documents`, and `Videos`.
	# Use the `mkdir` command.-manualy created for now

```
#!/bin/bash

# Create directories if they don't exist
mkdir -p Images
mkdir -p Documents
mkdir -p Videos
```

# Search for files in the current directory.
	- Use a loop and commands like `find` or `ls`.

	Search files if jpeg found move to pictures

```
#!/bin/bash

# Create directories if they don't exist
mkdir -p Images
mkdir -p Documents
mkdir -p Videos

# Loop through all files in the current directory
for file in *; do
	if [ -f “$file” ]; then  # Check for file
		echo “$file” found  # Print out that file is found
	fi
done
```

# Check the extension of each file.
	- For example: `.jpg`, `.png`, `.txt`, `.mp4`, etc.

```
#!/bin/bash

# Create directories if they don't exist
```

```
mkdir -p Images
mkdir -p Documents
mkdir -p Videos

# Loop through all files in the current directory
for file in *; do
        # Loop through all files in the current directory
        if [ -f “$file” ]; then # Check if it's a regular file
                case "$file" in
                        *.jpg) echo “$file is ready to be move” ;;  # For image files
                esac
        fi
done
```

#. Move files to the appropriate directory based on their extension.
- Use the `mv` command.

# Skip files already inside the correct directory.
- Check if the file already exists in the target directory.

```
#!/bin/bash

# Create directories if they don't exist
mkdir -p Images
mkdir -p Documents
mkdir -p Videos

# Loop through all files in the current directory
for file in *; do
    if [ -f "$file" ]; then  # Check if it's a regular file
        case "$file" in
            *.jpg|*.png|*.gif)  # For image files
                mv "$file" Images/
                ;;
            *.txt|*.pdf|*.docx)  # For document files
                mv "$file" Documents/
                ;;
            *.mp4|*.avi|*.mkv)  # For video files
                mv "$file" Videos/
                ;;
```

```
                *)   # Skip files
                ;;
        esac
    fi
done
```

    # Display the final directory structure.
        - Use the `tree` or `ls` command to confirm changes.

```
#!/bin/bash

# Create directories if they don't exist
mkdir -p Images
mkdir -p Documents
mkdir -p Videos

# Loop through all files in the current directory
for file in *; do
    if [ -f "$file" ]; then  # Check if it's a regular file
        case "$file" in
            *.jpg|*.png|*.gif)  # For image files
                mv "$file" Images/
                ;;
            *.txt|*.pdf|*.docx)  # For document files
                mv "$file" Documents/
                ;;
            *.mp4|*.avi|*.mkv)  # For video files
                mv "$file" Videos/
                ;;
                *)   # Skip files
                ;;
        esac
    fi
Done

# Display the final directory structure
echo "Final directory structure:"
tree
```

    # End the script.

NANO:

```
#!/bin/bash

# Create directories if they don't exist
mkdir -p Images
mkdir -p Documents
mkdir -p Videos

# Loop through all files in the current directory
for file in *; do
    if [ -f "$file" ]; then  # Check if it's a regular file
        case "$file" in
            *.jpg|*.png|*.gif)  # For image files
                mv "$file" Images/
                ;;
            *.txt|*.pdf|*.docx)  # For document files
                mv "$file" Documents/
                ;;
            *.mp4|*.avi|*.mkv)  # For video files
                mv "$file" Videos/
                ;;
            *)  # For any other files, do nothing
                ;;
        esac
    fi
done

# Display the final directory structure
echo "Final directory structure:"
tree
```

UNDO.SH

```
#!/bin/bash

# Loop through all files in the target directories
for dir in Images Documents Videos; do
    if [ -d "$dir" ]; then  # Check if the directory exists
        for file in "$dir"/*; do
            if [ -f "$file" ]; then  # Check if it's a regular file
                # Move each file back to the current directory
                mv "$file" .
                echo "Moved $file back to the current directory."
            fi
        done
    fi
done
```

—----don t peak too much

```
#!/bin/bash

# Directory Organizer Script
# This script organizes files into directories based on their extensions.

# Step 1: Start the script
# Print a starting message
echo "Starting Directory Organizer Script..."

# Step 2: Create directories for Images, Documents, and Videos
# Check if directories already exist, and create them if they don't
echo "Creating directories..."
mkdir -p Images Documents Videos

# Step 3: Search for files in the current directory
# List all files in the current directory (excluding directories)
echo "Searching for files..."
files=$(find . -maxdepth 1 -type f)

# Step 4: Loop through each file and check its extension
for file in $files; do
   # Extract the file extension using parameter expansion
   extension="${file##*.}"
         *.jpg)
   filename=$(basename "$file")

   # Step 5: Move files to the appropriate directory
   case "$extension" in
      jpg|png|gif)
         # Image files
         if [ ! -f "./Images/$filename" ]; then
            mv "$file" "./Images/"
            echo "Moved $filename to Images/"
         else
```

```
            echo "File $filename already exists in Images/"
        fi
        ;;
    txt|pdf|doc|docx)
        # Document files
        if [ ! -f "./Documents/$filename" ]; then
            mv "$file" "./Documents/"
            echo "Moved $filename to Documents/"
        else
            echo "File $filename already exists in Documents/"
        fi
        ;;
    mp4|avi|mkv)
        # Video files
        if [ ! -f "./Videos/$filename" ]; then
            mv "$file" "./Videos/"
            echo "Moved $filename to Videos/"
        else
            echo "File $filename already exists in Videos/"
        fi
        ;;
    *)
        # Skip files with unrecognized extensions
        echo "Skipped $filename (unrecognized extension: .$extension)"
        ;;
  esac
done

# Step 6: Display the final directory structure
echo "Final directory structure:"
tree || ls -R

# Step 7: End the script
echo "Directory organization complete. Exiting."
```

*########## Pseudocode for Challenge 2: Directory Organizer ##########*

```
# Step 1: Start the script
# Print a starting message to indicate the script has begun.
PRINT "Starting Directory Organizer..."

# Step 2: Create directories for categories
# Check if the `Images`, `Documents`, and `Videos` directories exist.
```

```
# If they don't exist, create them.
IF "Images" directory does not exist THEN
   CREATE directory "Images"
END IF

IF "Documents" directory does not exist THEN
   CREATE directory "Documents"
END IF

IF "Videos" directory does not exist THEN
   CREATE directory "Videos"
END IF

# Step 3: Search for all files in the current directory
# Exclude directories and only work with regular files.
SET files = FIND all files in the current directory (exclude directories)

# Step 4: Loop through each file
FOR each file IN files DO
   # Step 4.1: Extract the file extension
   SET extension = Extract the extension of the file (e.g., `.jpg`, `.txt`)

   # Step 5: Move the file to the appropriate directory based on its extension
   IF extension is "jpg", "png", or "gif" THEN
      # Check if the file is already in the "Images" directory
      IF file does not exist in "Images" THEN
         MOVE file to "Images"
         PRINT "Moved file to Images directory."
      ELSE
         PRINT "File already exists in Images directory. Skipping."
      END IF

   ELSE IF extension is "txt", "pdf", "doc", or "docx" THEN
      # Check if the file is already in the "Documents" directory
      IF file does not exist in "Documents" THEN
         MOVE file to "Documents"
         PRINT "Moved file to Documents directory."
      ELSE
         PRINT "File already exists in Documents directory. Skipping."
      END IF

   ELSE IF extension is "mp4", "avi", or "mkv" THEN
      # Check if the file is already in the "Videos" directory
      IF file does not exist in "Videos" THEN
```

```
            MOVE file to "Videos"
            PRINT "Moved file to Videos directory."
        ELSE
            PRINT "File already exists in Videos directory. Skipping."
        END IF

    ELSE
        # Step 6: Skip files with unrecognized extensions
        PRINT "Skipped file: Unrecognized extension."
    END IF
END FOR

# Step 7: Display the final directory structure
# Use a command to display the directory structure for confirmation.
PRINT "Final directory structure:"
SHOW directory tree structure

# Step 8: End the script
# Print a message indicating that the script has finished.
PRINT "Directory organization complete. Exiting."
```

```
#!/bin/bash
# Directory Organizer

# Create the directories if they don't exist
mkdir -p Images Documents Videos

# File names
files=("file.jpg" "file.png" "file.txt" "file.mp4")

# Loop through the files and create them if they don't exist
for file in "${files[@]}"; do
    if [ ! -e "$file" ]; then
        touch "$file"
        echo "$file created."
    else
        echo "$file already exists."
    fi
done

# Loop current directory
for file in *; do
    if [ -f "$file" ]; then
        # Extract the file extension
        ext="${file##*.}"
```

```
    case "$ext" in
        # Image files
        jpg|png|gif)
            if [ ! -f "./Images/$file" ]; then
                mv "$file" "./Images/"
                echo "Moved $file to Images/"
            else
                echo "File $file already exists in Images/"
            fi
            ;;
        # Document files
        txt|pdf|doc|docx)
            if [ ! -f "./Documents/$file" ]; then
                mv "$file" "./Documents/"
                echo "Moved $file to Documents/"
            else
                echo "File $file already exists in Documents/"
            fi
            ;;
        # Video files
        mp4|avi|mkv)
            if [ ! -f "./Videos/$file" ]; then
                mv "$file" "./Videos/"
                echo "Moved $file to Videos/"
            else
                echo "File $file already exists in Videos/"
            fi
            ;;
        # Unknown file types
        *)
            echo "Unknown file type for $file"
            ;;
    esac
  fi
done
```