

```
ell-scripting % .
```

```
o the power of 2:
```

```
umber 2: 2.24
```

```
ell-scripting %
```

Shell Scripting Techniques

In this presentation, we will explore some useful shell scripting techniques with code, input-output samples, and related images. Let's get started!

Display Lines from a File

Have you ever wanted to extract specific lines from a file? With this shell script, you can enter the file name and line numbers, and it will display all the lines in between. Simple and efficient!

Input

File name, starting line
number, ending line
number

Code

```
echo "Enter the file  
name:"  
read file_name  
echo "Enter the starting  
line number:"  
read start_line  
echo "Enter the ending  
line number:"  
read end_line  
sed -n  
$start_line,$end_line\p  
$file_name
```

Output

Enter file name: a.txt
Enter the starting line : 2
Enter the ending line : 5
Result: i am new to

Delete Lines from a File

Is there a particular word that you want to delete from a file? This shell script will ask you for the file name and the word to delete, and it will remove all the lines containing that word. Faster than manual deletion, isn't it?

Input

File name, word to delete

Code:

```
echo "Enter the file
name:"
read file_name
echo "Enter the word to
delete:"
read delete_word
grep -v $delete_word
$file_name > temp.txt
mv temp.txt $file_name
cat $file_name
```

Output:

```
filename: a.txt word: hi

file having content before
removing word hi is

hi i am new to this linux
programing

file having content after
removing word

i am new to linux
programing
```

List of Files in a Directory

Want to see a list of all the files in the current directory? This simple shell script will provide a complete list of files that are present in the current directory. It's that easy!

Input

No input is required

Code

```
echo $(ls)
```

Output

```
a.txt b.txt s2.sh s3.sh  
s4.sh s.sh
```

Check Type of a File/Directory

Ever found yourself in a situation where you need to differentiate between a file and a directory? This shell script can check if a specific file name is a file type or a directory type. Simple yet useful!

Input

List of file/directory
names

Code:

```
read n_files
while [ $n_files -gt 0 ]; do
  echo "Enter file/directory
  name:"
  read file_name
  if [ -f $file_name ]; then
    echo "$file_name is a file"
  elif [ -d $file_name ]; then
    echo "$file_name is a
    directory"
  else echo "$file_name is
  neither a file nor a
  directory"
  fi ((n_files=n_files-1))
done
```

Output:

```
4
a.txt
it is File
b.txt it is File
Desktop
it is neither of them
jk it is neither of them
```

Copying a File to Another Directory

Do you need to create a copy of a file and place it in a different directory? This shell script takes the file name and the destination folder as inputs and creates a copy of the file in the specified folder. Say goodbye to manual copying!

Input

File name, destination
folder path

Code

```
echo "Enter the file  
name:"  
read file_name  
echo "Enter the  
destination folder:"  
read folder_name  
cp $file_name  
$folder_name
```

Output

A confirmation message
that the file has been
copied to the destination
folder

Count Occurrences of a Word in a File

Have you ever needed to check the number of times a specific word occurs in a file? With this shell script, you can do it in no time. Simply enter the file name and the word to search for, and it will count the number of occurrences. Efficient and precise!

Input

File name, word to count

Code:

```
echo "Enter the file
name:"
read file_name
echo "Enter the word to
be counted:"
read word
count=$(grep -o -i $word
$file_name | wc -l)
echo "The word $word
appears in $file_name
$count times."
```

Output:

The number of times the
word appears in the file

Creating a New Directory

Need to create a new directory in the current folder? With this simple shell script, you can create a new directory and assign a name to it. Say hello to easy organization!

Input

New directory name

Code:

```
echo "Enter the new  
directory name:"  
read dir_name  
mkdir $dir_name
```

Output:

A confirmation message
that the new directory
has been created