Advanced Shell Scripting

Conditions in shell script.
whenever you want to manage control flow. we can use if, elif and else block
Comparison Operators
Numeric Comparison

Operator	Meaning
-eq	Equal
-ne	Not Equal
-gt	greater than
-lt	less than
-ge	greater than or equal
-le	less than or equal

String operators:

Operator	Meaning
=	Equal
!=	Not Equal
>	greater than
<	less than

If you are using string that's fine else with number you have to use [[]] for adding condition

Repetitive Task: iterative process using loop
System Scrips

```
 8.SystemUpdate.sh  ×

  1    echo "***********update packages**********"
  2    sudo apt update -y
  3
  4    echo "**************upgrade packages**********"
  5    sudo apt upgrade -y
  6
  7    echo "**************Remove Obsolete Packages************"
  8    sudo apt autoremove -y
  9
 10    echo "**** System is Up to date and cleanup is completed *****"
```

To run the same open wsl and go to the folder

change drive in wsl: cd /mnt/d

```
sonam@DESKTOP-4F8ELLU:/mnt/d/PhysicsWalla/Devops-March/Github/ShellScripting/day-10$ ./8.SystemUpdate.sh
***********update packages**********
Hit:1 https://apt.releases.hashicorp.com noble InRelease
Hit:2 http://archive.ubuntu.com/ubuntu noble InRelease
Hit:3 http://security.ubuntu.com/ubuntu noble-security InRelease
Hit:4 http://archive.ubuntu.com/ubuntu noble-updates InRelease
Hit:5 http://archive.ubuntu.com/ubuntu noble-backports InRelease
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
All packages are up to date.
**************upgrade packages**********
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
Calculating upgrade... Done
0 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.
**************Remove Obsolete Packages************
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
0 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.
**** System is Up to date and cleanup is completed *****
```

tar command

tar -czvf backup.tar.gz /localtion/folder

-c --> Create Archive

-z --> Compressed Gzip

-v --> Get output in Version (see the progress)

-f --> file name for archive

Extract tar file

tar -xzvf backup.tar.gz

Provide the list of content

tar -tzvf backup.tar.gz