

Let's Write your first python Script

Step 1 → Understand Variables
& Data types

Step 2 → Do some magic with them

Step 3 → Run it.

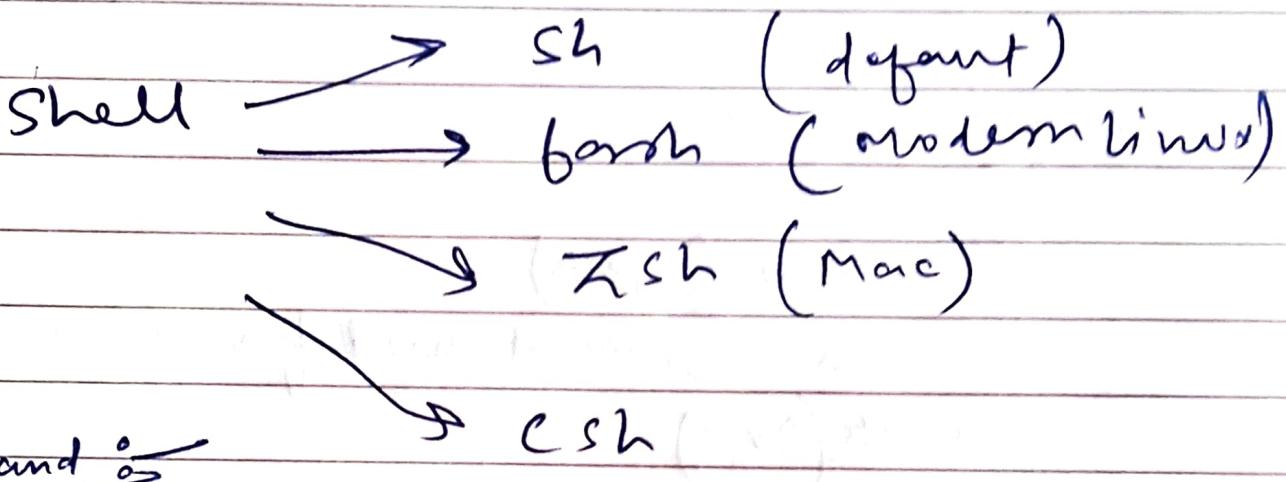
Shell Scripting.

- ① Basic script & giving permission.
- ② Variable, Constant, User inputs
- ③ if condition [] fi,
- ④ Package installer script
- ⑤ Loops
- ⑥ Arguments, functions
- ⑦ Error handling
- ⑧ awk, sed, grep, find
- ⑨ server bkp & dump project ,

Shell Scripting for Devops.

.sh files - Collection of shell commands.

Chmod +x executable.



>> printenv (print all environment Variables)

★ Variable → \$ Later here

if condition - if this or that .

if [" /etc/passwd"];
if, else, fi
] if [
else]
elif]

if [conditions]; then
if commands ; then
fi

fi
✓

Loops :-

① for loop :-

for i in {1...5}
do

echo \$i
done

#!/bin/bash.

<< Comments (Starts)

This is Comment used for upcoming pgm

Comments (Over)

for i in {1..5}

do

read -p "Enter the user name" user_name

sudo useradd -m \$user_name

echo "User \$user_name added &c"

done

mkdir -p day{1..90}
rm -r day{1..90}

→ for i in {1..90} → Variable
do

mkdir -p day\$i

done. } define Kr do Kha le sru Kha hoi

→ for ((i=1 ; i<=10 ; i++))
do
echo "\$i" ↘ Kha tak
echo "hello" ↗ jaega ↗ hoga
done. ↗ decrease.

① Argument

Terminal
\$> mkdir; ↗ argument

real time Use

{ echo "install \$1"
sudo apt update &&
sudo apt install \$1 -y
echo "Svr install \$1"

Ex: \$> ./filename.sh amit ankit
\$0 \$1 \$2

Note \$@ → print all arguments,
\$# → total number of arguments

#!/bin/bash

if [\$# -eq 0]

then

echo "Please file pass a file as argument"

echo "Usage : ./if_file_exist.sh<file>"

exit 1

fi

if [-f \$1]

then

echo "file exists"

else

echo "file not exist"

fi

Ⓐ functions: [reusable] nota hui

```
function_name() {  
    echo " haldi lagao"  
    echo " pani ne dalo"  
}  
} { } formate
```

for calling > haldi # function call
 |
 | function name.

① Example:

#!/bin/bash

<< usage

./function.sh hello → argument (main)

inside function call

install-package docker.io → argument (local)

usage

echo "\$1 is the main argument passed to script"

function define

```
install_package() {
```

 echo "\$1 is local argument"

 Sudo apt install \$1

} { }
 install_package docker.io.

- function bane ke yahan do and me kew while you call your function
- Argument pass kro do function me niche local argument bolenge.

★ Create User.

```
#!/bin/bash
```

<<cmf

- take user name as input
- take password as input
- create the user

cmf

```
read -p " Username" Username
read -p " password" password
if id "$Username" &gt;/dev/null; then
    echo " The user $Username exists"
    exit 1
else
    echo " the user $Username will be created"
fi
sudo useradd -m $Username -p $password
echo " User $Username added successfully"
```

`#!/bin/bash.`

`source " . / filename"`

`create_user` ↗ → function



Error handling :-

Example :-

`Mkdirs josh`

`echo "do prod work"`

Output

Errors : -----

do prod work

→ `#!/bin/bash`

`set -e`

shell script
Kya Karta hai
Ki agar error aya
to next line ko execute
Karta hai error ko
ignore Karta hai.
that's why.
error handling required

`<try> || <catch>`



O.K.

Note : Every error to be handled differently.
[mkdir, not mounted, disk]

`<<error detected >> || { fallback logic to handle
error }`