Docker Commands

sudo yum update -y # Update all system packages
sudo yum install docker -y # Install Docker
sudo service docker start # Start Docker service
sudo docker version # Show Docker version (detailed)
sudo docker login # Login to Docker Hub
sudo docker pull ubuntu # Download Ubuntu image
sudo docker pull redis # Download Redis image
sudo docker images # List downloaded images
sudo docker ps # List running containers
sudo docker ps -a # List all containers (running + stopped)
sudo docker run redis # Run a Redis container
docker tag <id> name:tag # Tag an image (example: aj:best)
touch filename # Create a new file
nano filename # Edit file in nano (for Dockerfile)
(FROM ubuntu
RUN apt-get update
MAINTAINER <name>)
chmod 777 filename # Give full permission to file
sudo docker build . # Build image from Dockerfile
sudo docker build -t <imagename>:1.0 . # Build image with custom name & version

---

Git Commands
git status # Check the status of your repository
git init # Initialize a new Git repository
git add <file_name> # Add a specific file to staging
git add . # Add all files to staging (no "space add")
git commit -m "My First Task" # Save changes in local repository with a message
git config --global user.email "you@example.com" # Set global email for commits
git config --global user.name "Your Name" # Set global username for commits
git clone https://github.com/<username>/<repo>.git # Clone a GitHub repo into local system
git branch # List all branches
git branch <branch_name> # Create a new branch
git checkout <branch_name> # Switch to that branch
git checkout -b <branch_name> # Create and switch to a new branch in one step
git add . # Stage all changes
git commit -m "My workspace" # Commit changes with a message
git checkout main # Switch back to main branch
git merge <branch_name> # Merge a branch into main
git push origin main # Push main branch to GitHub
git push origin <branch_name> # Push a specific branch to GitHub
==========

LINUX COMMANDS: ubuntu

pwd : path of current location // to see the current location

ls : list down all directorires files for the rescpective location //list down the things sthat we have

mkdir <directory name> : // make new directory

rmdir : remove the directory

rm <filename>// remove /delete files

whoami : user name // finds the user's name

cd <directory name> // change / move to particular directory

date : check the cuurent date (shows)

date +%D : change the format of date

date +%T : format for time

date +%H:%M : format Hours & Minutes

cd .. // coming back from directory (opting out from directory)

ls -lt // all details of directory (recently make first)

ls -ltr // Reverse the order

touch <filename.extension> // create a file {touch abc.txt}

nano <filename.ext> // editor

cat <filename.ext> // read the files

clear // Clears all the terminal

mv <file_name> <directory>

Selinum Codes

```
import org.openqa.selenium.By;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.chrome.ChromeDriver;
import io.github.bonigarcia.wdm.WebDriverManager;
public class Test {
	public static void main(String[] args) {
		// TODO Auto-generated method stub
		WebDriverManager.chromedriver().setup();
		WebDriver driver = new ChromeDriver();
		driver.get("https://www.youtube.com/");
		driver.findElement(By.name("serach_query")).sendKeys("name");
//		driver.findElement(By.name("btnK")).click();
	}
}----------------------------------------------------------------------------------------------------
	public static void main(String[] args) {
	WebDriverManager.chromedriver().setup();
	WebDriver driver = new ChromeDriver();
	driver.get("https://www.saucedemo.com");
	driver.findElement(By.xpath("//*[@id=\"user-name\"]")).sendKeys("standard_user");
	driver.findElement(By.name("password")).sendKeys("secret_sauce");
	driver.findElement(By.id("login-button")).click();
```

Linux Codes

```
# While loop example: prints numbers 0 to 9
#!/bin/bash
a=0
while [ $a -lt 10 ]
do
echo $a
a=`expr $a + 1`
done
```

```
# Until loop example: prints numbers 0 to 10
#!/bin/bash
a=0
until [ $a -gt 10 ]
do
echo $a
a=`expr $a + 1`
done
```

```
# For loop with break: stops when value reaches 5
#!/bin/bash
for a in 1 2 3 4 5 6 7 8 9 10
do
if [ $a -eq 5 ]
then
break
fi
echo "Iteration no $a"
done
```

```
# For loop with continue: skips printing when value is 5
#!/bin/bash
for a in 1 2 3 4 5 6 7 8 9 10
do
if [ $a -eq 5 ]
then
continue
fi
echo "Iteration no $a"
done
```

```
# Infinite loop using while true: prints message every 1 second
#!/bin/bash
while true
do
echo "Hi, I am infinity loop"
sleep 1
done
```

```
# Simple if-fi example: checks if Name equals "Girish"
#!/bin/bash
Name="Girish"
if [ "$Name" = "Girish" ]; then
echo "His name is Girish. It is true."
fi
```

```
# If-else example: checks if Age is 18 or older
#!/bin/bash
Age=17
if [ "$Age" -ge 18 ]; then
echo "You can vote"
else
echo "You cannot vote"
fi
```

```
# If-elif-else example: checks age for voting eligibility
#!/bin/bash
Age=17
if [ "$Age" -ge 18 ]; then
echo "You can vote"
elif [ "$Age" -eq 17 ]; then
echo "You can vote after one year"
else
echo "You cannot vote"
fi
```

```
# Nested if-else example: subject check + marks validation
#!/bin/bash
echo "Enter subject"
read subject
if [ "$subject" = "Linux" ]
then
echo "Enter Marks"
read marks
if [ $marks -ge 30 ]
then
echo "You passed"
else
echo "You failed"
fi
else
echo "Wrong Subject"
fi
```
