Branching Steps

Create a new branch GitNewBranch

```
git branch GitNewBranch
```

List all local and remote branches

```
git branch     # for local
git branch -r  # for remote
git branch -a  # for both
```

Switch to the newly created branch

```
git checkout GitNewBranch
```

or (in newer Git versions)

```
git switch GitNewBranch
```

Add some files to the branch

```
echo "This is a feature branch file." > feature.txt
mkdir src
echo "console.log('Hello from branch');" > src/script.js
```

Commit the changes

```
git add .
git commit -m "Added feature files in GitNewBranch"
```

Check Git Status

```
git status
```

Expected:

pgsql

CopyEdit

```
On branch GitNewBranch
```

Merging Steps

Switch back to main (or master)

```
git checkout main
```

.Show differences between main and GitNewBranch

```
git diff main..GitNewBranch
```

This will list **command-line diff** output of changes in GitNewBranch.

View visual differences using P4Merge

```
git config --global diff.tool p4merge
git config --global merge.tool p4merge
```

Then run:

```
git difftool main GitNewBranch
```

P4Merge will open showing the visual differences between branches.

Merge GitNewBranch into main

```
git merge GitNewBranch
```

If there are no conflicts, you'll see:

csharp

CopyEdit

View Git Log Graphically

```
git log --oneline --graph –decorate
```

output:

```
*   3f3a1ab (HEAD -> main) Merge branch 'GitNewBranch'
|\
| * b8e1e53 (GitNewBranch) Added feature files in GitNewBranch
* | 1d7a9fb Initial commit
|/
```

Delete the merged branch

```
git branch -d GitNewBranch
```

To force delete (if not merged):

```
git branch -D GitNewBranch
```
