# 🧠 GIT & GITHUB CHEATSHEET FOR DATA SCIENCE

## 🌎 What is Version Control?

Version Control is a system that records changes to your code or project files over time so you can easily track versions, collaborate, and revert to earlier stages when needed.

It helps developers and data scientists manage their code efficiently and work together without confusion.

### 💡 Why Version Control is Important in Data Science

📊 Data science projects involve multiple files, datasets, scripts, and experiments. Version control tools like Git and GitHub help you:

✅ Track Changes: Know who changed what and when. ✅ Collaborate Easily: Work with teammates without overwriting each other's work. ✅ Experiment Safely: Use branches to try new ideas without breaking the main project. ✅ Reproduce Results: Keep code versions aligned with data and models. ✅ Backup & Access Anywhere: Store projects in the cloud (GitHub) for easy sharing. ✅ Professional Practice: Most organizations expect version-controlled workflow for ML, AI, or analytics projects.

In short: Version control = Organized, Reproducible, Collaborative Data Science.

## 🔹 1. Basic Setup Commands

In [ ]:
```
git config --global user.name "Your Name"
git config --global user.email "your_email@example.com"
```

✅ **Sets your identity for commits.**

## 🔹 2. Initialize a Repository

In [ ]:
```
git init
```

✅ **Creates a new local Git repository.**

## 🔹 3. Connect Local Repo to GitHub

In [ ]:
```
git remote add origin https://github.com/username/repo-name.git
```

✅ **Links local project to GitHub repo.**

## 🔹 4. Add & Commit Changes

In [ ]:
```
git add .
git commit -m "Your commit message"
```

✅ **Adds all files and commits changes locally.**

## 🔹 5. Push Code to GitHub

In [ ]:
```
git branch -M main        # Rename master to main (if needed)
git push -u origin main   # Push first time
git push                  # Later pushes
```

## 🔹 6. Pull Updates from GitHub

In [ ]:
```
git pull origin main
```

✅ **Fetches and merges changes from remote repo.**

## 🔹 7. Check Repo Status

In [ ]:
```
git status
```

✅ **Shows modified/untracked files.**

## 🔹 8. View Commit History

In [ ]:
```
git log
```

✅ **Shows the list of all commits.**

## 🔹 9. Branching (for safe experiments)

In [ ]:
```
git branch new-feature
git checkout new-feature
# OR
git switch -c new-feature
```

✅ **Creates and switches to a new branch.**

## 🔹 10. Merging Branches

In [ ]:
```
git checkout main
git merge new-feature
```

✅ **Merges the branch into the main project.**

## 🔹 11. Clone a Repository

In [ ]:
```
git clone https://github.com/username/repo-name.git
```

✅ **Copies a remote repo to your system.**

## 🔹 12. Delete Branch

In [ ]:
```
git branch -d branch-name
```

✅ **Deletes a local branch.**

## 🔹 13. Undo Mistakes

In [ ]:
```
git restore filename           # Undo changes in a file
git reset --hard HEAD~1        # Undo last commit
```

## 🔹 14. Common Workflow (Simple Reminder)

In [ ]:
```
1  git add .
2  git commit -m "Message"
3  git push origin main
```