🚀 Git & GitHub Beginner Guide (Step-by-Step)

- **Git** is a distributed version control system, created by **Linus Torvalds** in 2008.
- **GitHub** is a cloud-based platform for hosting Git repositories, collaboration, and version management.

🔐 Step 1: Generate SSH Key for GitHub Access

1. **Search on Google:**

   ```
   Connecting to GitHub with SSH:

   Open the official GitHub Docs page titled "Generating a new SSH key and adding it to the ssh-agent".
   ```

2. Open Git Bash on your system.
3. Run the following command to generate a new SSH key (replace your email address):

   ```
   ssh-keygen -t ed25519 -C "your_email@example.com"
   ```

4. When prompted, press Enter to accept the default file location and set a passphrase if desired.
5. Once the SSH key is generated, save the location where it's stored. It is usually:

   ```
   ~/.ssh/id_ed25519
   ```

6. Start the SSH agent:

   ```
   eval "$(ssh-agent -s)"
   ```

   You'll see a line like:
   Agent pid 1234 (the number will vary)

🔑 Step 2: Add the SSH Key to Your GitHub Account

1. Go back to the same GitHub Docs page and click on the link:

   ```
   "Adding a new SSH key to your GitHub account"
   ```

2. Copy your SSH public key using this command:

   ```
   cat ~/.ssh/id_ed25519.pub
   ```

⚠ Replace id_ed25519.pub with your actual filename if it's different.

3. Copy the output (the entire key starting with ssh-ed25519).
4. Go to GitHub:

5. Paste the copied key in the Key field and give it a title (e.g., My Laptop SSH Key).
6. Click Add SSH key.

☑ Done! You've successfully connected GitHub with SSH and can now clone, push, and pull using SSH.

🛠 Step 1: Initial Git Setup (One-Time)

✓ **Configure your username and email in Git:**

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

✓ **Check your current Git config:**

```
git config --global user.name
git config --global user.email
```

📁 Step 2: Setting Up a Project with Git Locally

- **Navigate to your folder using Git Bash:**

```
cd F:          # Change to F drive
cd MyProject   # Enter your project folder
```

- **Initialize a Git repository:**

```
git init       # Start a new Git repository
code .         # Open the folder in VS Code
ls -lart       # Show all files, including hidden (.git)
```

📄 Step 3: Basic File Handling and Staging

- **Track your files and changes:**

```
git status          # Shows current status of files
```

- **Add files to the staging area:**

```
git add index.html     # Add single file
git add .              # Add all modified and new files
```

- **Commit changes:**

```
git commit -m "Initial commit"        # Commit with message
git commit                            # Opens vim to write commit message
```

- **Undo staged files:**

```
git restore --staged index.html       # Remove from staging area
```

- **Undo changes to files:**

```
git checkout about.html       # Revert about.html to last commit
git checkout -f               # Revert ALL files to last commit
```

🔍 Step 4: Explore Commit History & Differences

✓ **View commit history:**

```
git log               # Full commit history
git log -p -5         # Last 5 commits with changes
```

✓ **View changes:**

```
git diff              # Working directory vs staging
git diff --staged     # Staging vs last commit
```

✓ **Skip staging & commit directly:**

```
git commit -a -m "Direct commit without staging"
```

🧹 Step 5: Deleting Files & Ignoring Files

➢ **Delete files:**

```
rm filename.html              # Remove from disk
git rm filename.html          # Remove and stage for commit
```

```
git rm --cached filename.html      # Untrack but keep in working directory
```

➢ **Ignore unwanted files:**

```
touch .gitignore              # Create .gitignore file
# Add this inside the file
/mylogs.log                   # Ignore mylogs.log
```

🌿 Step 6: Working with Branches

✅ **Create and manage branches:**

```
git branch                    # List all branches
git branch feature1           # Create a new branch
git checkout feature1         # Switch to feature1
git checkout -b feature2      # Create and switch to feature2
```

✅ **Merge branches:**

```
git checkout master           # Switch back to main branch
git merge feature1            # Merge feature1 into master
```

☁️ Step 8: Connect Local Project to GitHub

🔗 **Create a new repository on GitHub**

🔗 **Connect your local repo:**

```
git remote add origin git@github.com:yourusername/repo-name.git
git branch -M main
```

```
git push -u origin main
```

🤝 Step 9: Contribute to Other Projects (Open Source Flow)

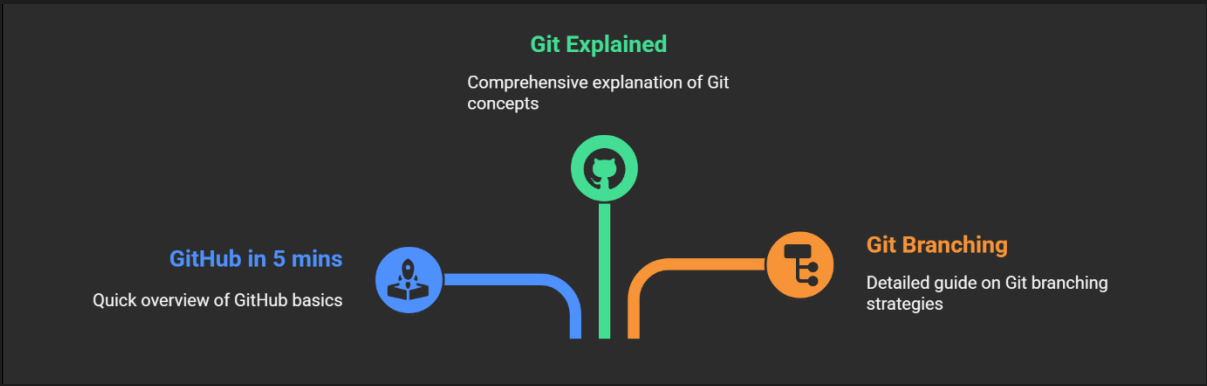
```
1. git clone https://github.com/username/repo-name.git
2. cd repo-name
3. git checkout -b feature/your-feature-name
4. # Make changes using your editor
5. git add .
6. git commit -m "Add: Short description"
7. git push origin feature/your-feature-name
```

- **Create a Pull Request:**
  1. Go to your forked repo on GitHub
  2. Click on **"Compare & pull request"**
  3. Add title and description
  4. Click **Create Pull Request**

📚 Useful Resources to Learn Git & GitHub

- 📘 GitHub Git Cheatsheet
- 📗 Pro Git Book (git-scm.com)
- 📥 GitHub Training Sheet (PDF)
- 🧠 Git Basics by Microsoft

📺 Video Tutorials for Visual Learning



- 🚀 GitHub in 5 mins – Fireship
- 📁 Git Explained – Tech With Tim
- 🔁 Git Branching – Academind

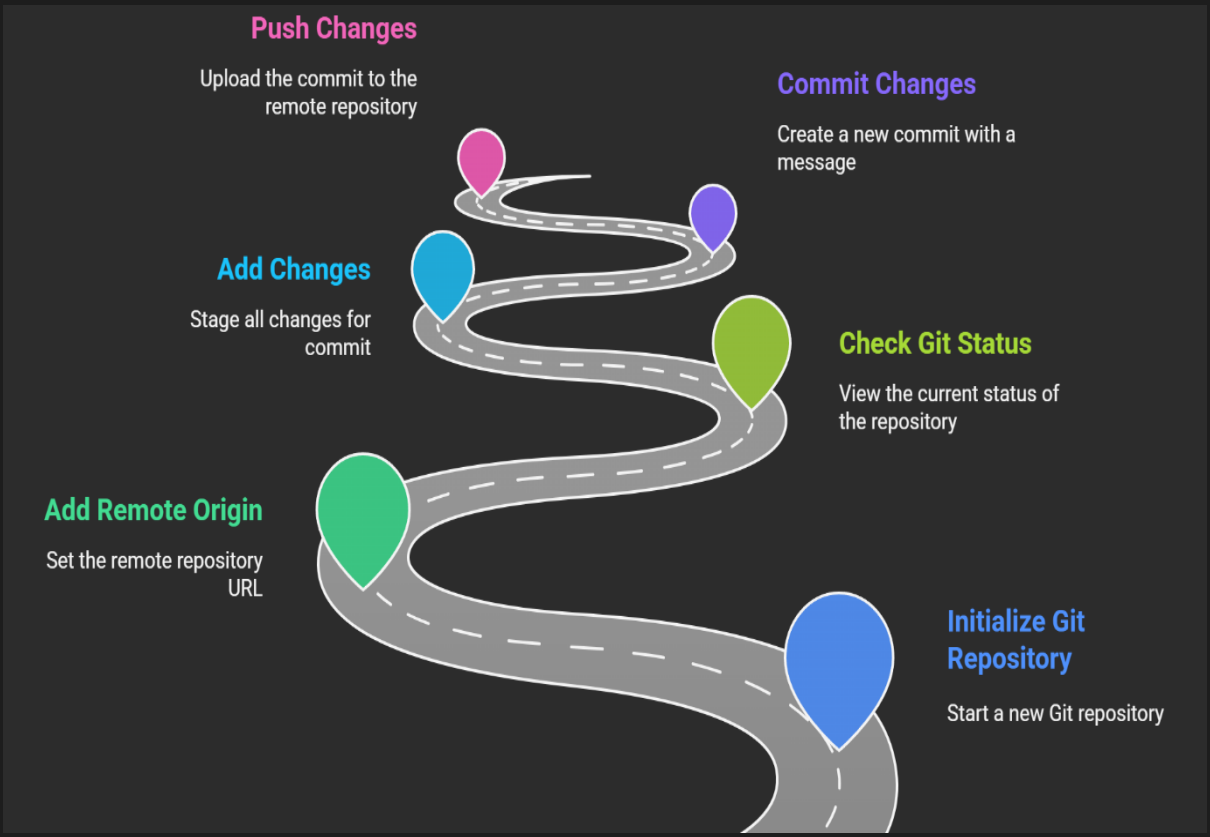
☑️ Handy Git Shortcuts

Command	Description
git status -s	Short summary view
touch file.html	Create file via Git Bash
ls	List files
rm -rf file.html	Force remove file
git reset <commit-id>	Undo commits above this one

🎓 Summary: Git Workflow at a Glance

```
1. # Setup
2. git init
3. git remote add origin <repo-url>
```

```
# Everyday Workflow
git status
git add .
git commit -m "Message"
git push origin branch-name
```



💬 Let's Connect

I'd love to hear from you, collaborate, or simply connect over tech and ideas. Feel free to explore more or reach out through the links below.

🌐 Portfolio 🐱 GitHub 🔗 Linkedin Instagram

Keep Learning. Keep Building.

Thanks again for reading — wishing you continued success and growth in your development journey!

— **Rishabh Singh**