# Git Workshop — 2-Page Cheat-Sheet

## 0 · First-Time Setup

```
git --version                                 # confirm install
git config --global user.name  "Your Name"
git config --global user.email "you@ou.edu"
git config --global init.defaultBranch main
touch .gitignore                              # list files/folders to sk
```

## 1 · Create / Get a Repository

```
git init                # start new repo in current directory
git clone <URL>         # copy existing repo
```

## 2 · Everyday Workflow

```
git status                          # what changed?
git add <file> | .                  # stage work
git commit -m "brief message"       # snapshot
git log --oneline --graph           # history
git diff [<base> <tip>]             # line-level changes
```

> **File states:** *working* ➜ *staged* ➜ *committed*

## 3 · Branching & Switching

```
git branch                  # list local branches
git branch <feat>           # create branch
git switch <feat>           # move to branch  (or: git checkout <feat>
git switch -c <feat>        # create **and** switch
```

## 4 · Merging & Conflicts

```
git merge <source-branch>         # into current branch
# conflict markers look like:
<<<<<<< HEAD
local change
=======
incoming change
>>>>>>>
# resolve, then:
git add <file>
git commit                        # finish merge
```

*Fast-forward = linear history · 3-way = true merge commit*

## 5 · Remotes (GitHub)

```
git remote add origin <URL>       # link once
git push -u origin main           # first push (sets upstream)
git push                          # thereafter
git pull                          # fetch + merge
git fetch                         # fetch only
```

*PR flow → push branch → open Pull Request → review → merge*

## 6 · Undo & Time-Travel

```
git restore <file>                # discard unstaged edits
git revert <commit>               # safe undo (new commit)
git reset --hard <commit>         # rewind branch (destructive)
git switch -                      # return to previous branch
```

*Detached HEAD?* `git switch main` (or new branch) to re-attach.

---

## 7 · Handy Shortcuts

```
alias gs='git status -sb'
alias gl='git log --oneline --graph --decorate --all'
git stash / git stash pop         # shelve work-in-progress
```

`.gitignore` examples → `*.log`, `node_modules/`, `*.DS_Store`

---

## 8 · Best Practices

- Commit **early** and **small**.
- One feature = one branch.
- Clear messages: `type: short summary` (e.g., `fix: handle null ID`).
- Keep `main` always deployable.
- Push **before** you pull requests from others.

---

### Further Help

`git help <command>` | Pro Git Book | GitHub Learning Lab