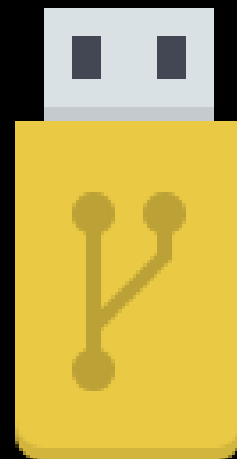# Informatics 3A
A World of Source Control

UNIVERSITY OF JOHANNESBURG

Source Control

Introduction

- What is source control?

- What tools are available?
  - Git
  - Team Foundation Server
  - CVS
  - Mercurial
  - Subversion

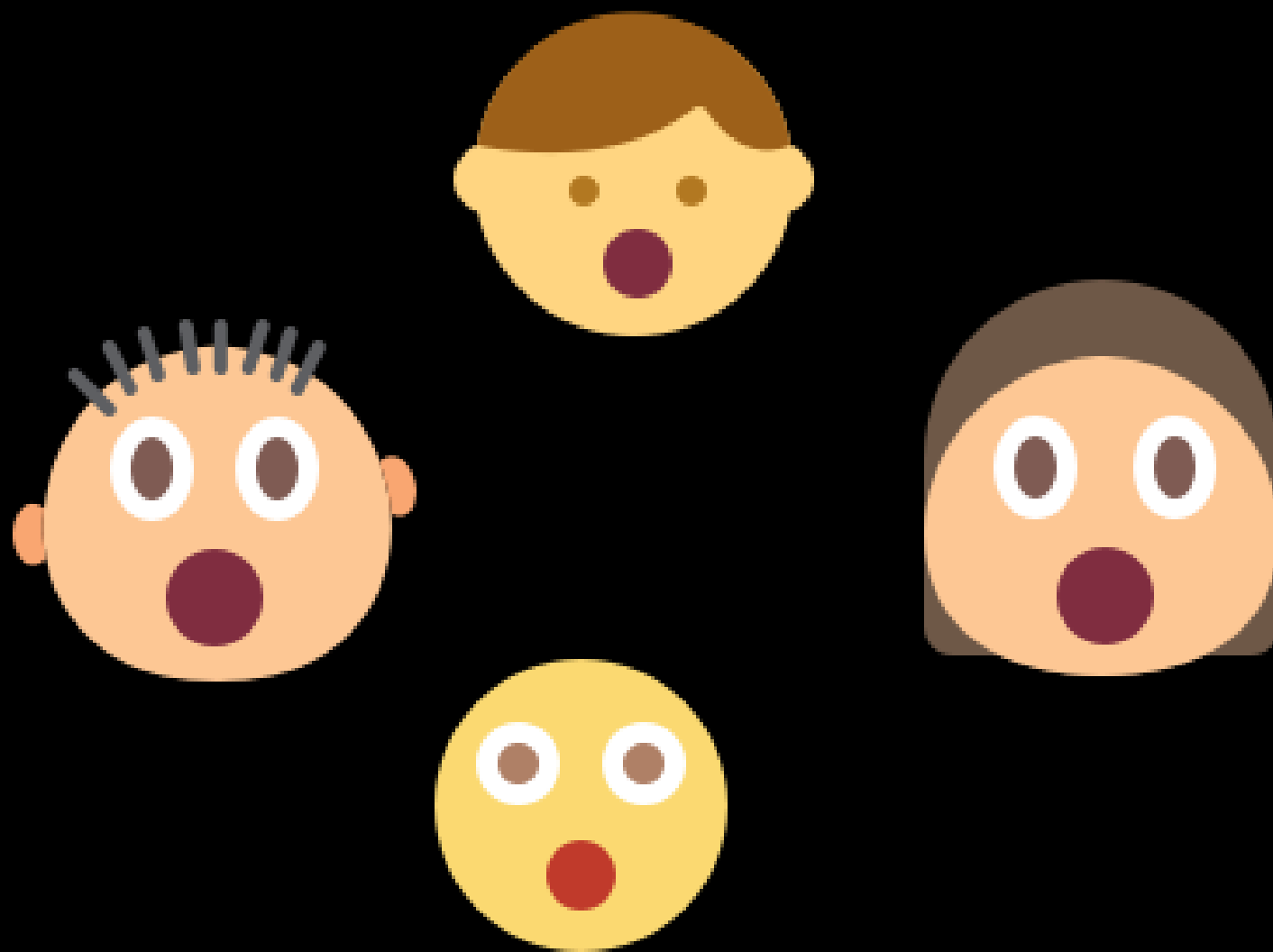# Where can I git it?

- GIT

  `https://git-scm.com/`
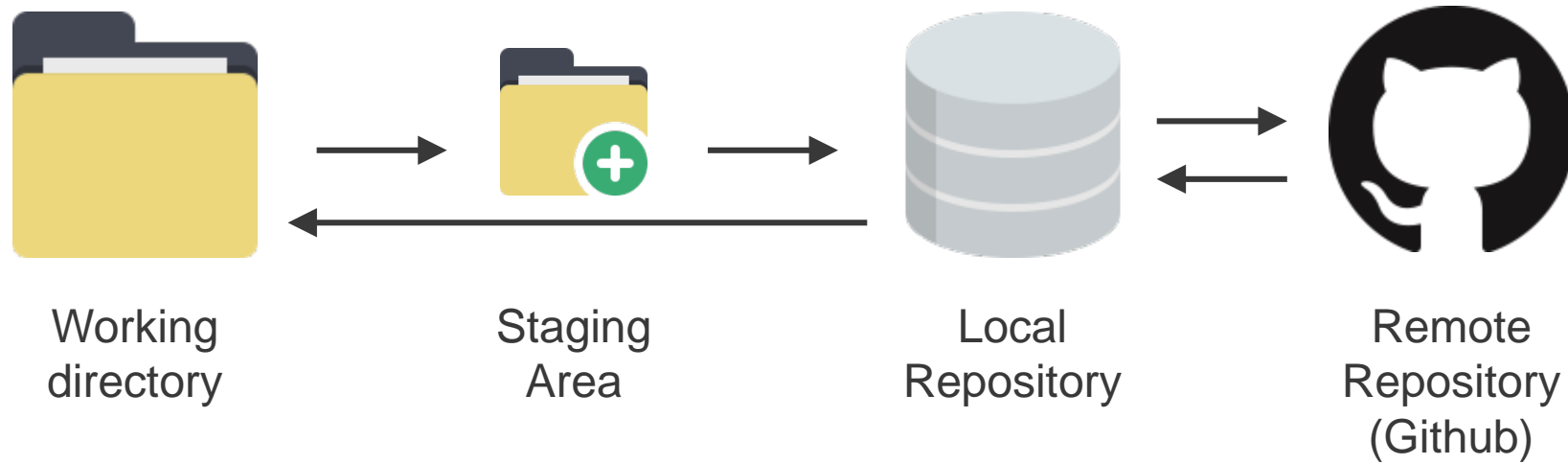
- TortoiseGIT

  `https://tortoisegit.org/`

GIT

Git Process

Working directory → Staging Area → Local Repository ⇄ Remote Repository (Github)

Initial Commands

- Personal details
    - `git config --global user.name "Your Name"`
    - `git config --global user.email "your@email"`
- Save credentials
    - `git config --global credential.helper store`
- Proxy (Computer Labs)
- `git config http.proxy http://username:password@152.106.240.140:3128`
- `git config https.proxy http://username:password@152.106.240.140:3128`
- Proxy (WiFi)
- `git config --global http.proxy http://username:password@10.200.254.1:3128`
- `git config --global https.proxy http://username:password@10.200.254.1:3128`

Only use `--global` if you're using your own machine.

Getting the Repository

- Create a working copy of a repository

```
git clone <repository>
```

What's going on?

- At any time!

```
git status
```

- What changed?

```
git diff
```

Adding & Committing

- Propose Changes to Staging Area

```
git add <filenames>
git add *
```

- Committing changes

```
git commit
git commit -m "commit message"
```

Pushing Changes

- Push changes to remote repository

```
git push

git push <remote> <branch>

git push origin master
```

Branches

master ━━

Branching

- Create new branch

```
git branch <new branch>
```

- Switching branches

```
git checkout <branch>
```

- Pushing branch to remote repository

```
git push <remote> <branch>
git push origin <branch>
```

Branch Name Conventions

1. Use grouping tokens (words) at the beginning of your branch names.
2. Define and use short lead tokens to differentiate branches in a way that is meaningful to your workflow.
3. Use slashes to separate parts of your branch names.
4. Do not use bare numbers as leading parts.
5. Avoid long descriptive names for long-lived branches.

- Suggested reading:
  http://nvie.com/posts/a-successful-git-branching-model/

Branch Name Conventions

- **develop**   Main development off of master
- **release**   Close to release / release candidate
- **feature**   Feature adding or expanding
- **wip**   Works in progress won't be finished soon
- **bugfix**   Bug fix or experiment
- **junk**   Throwaway branch created to experiment

Updating

- Update local repository to match remote (current branch)
    - `git pull`

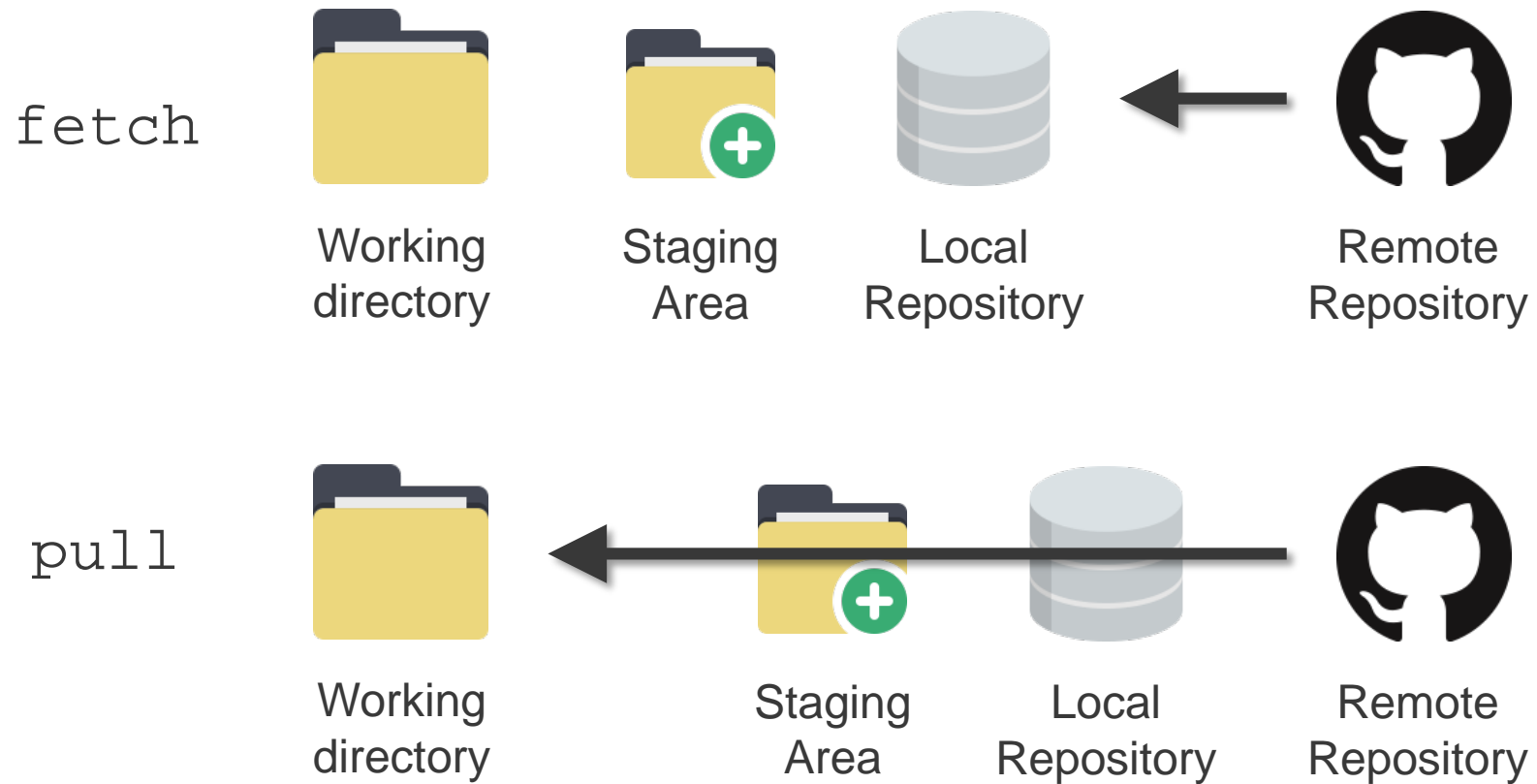- Update local repository to match all remote branches
    - `git pull --all`

- Update local repository without merging
    - `git fetch`

Updating

Merging

- Merging another branch with current

```
git merge <branch>
```

- If merging results in conflicts
  - Cannot auto-merge
  - Resolve conflicts first!

```
git commit
```

- Oh $%!^

```
git merge --abort
```

Mistakes?

- Revert file to current commit

```
git checkout -- <filename>
```

- Revert ALL files to current commit

```
git checkout -- .
```

- Remove untracked files

```
git clean -df
```

Big Mistake?

- See reference log

```
git reflog
```

- Revert ALL files to current commit

```
git revert <commit-ish id>
```

Not ready for a commitment?

- Stash current changes

```
git stash
```

- Reapplying stashed changes

```
git stash apply
git stash pop
```

- Clear this mess

```
git stash clear
```

Tagging

- Creating tag

```
git tag <tag name> <commit-ish id>
```

- Get that tag to the remote repository

```
git push --tags
```

Additional Reading

- https://www.youtube.com/watch?v=MJUJ4wbFm_A

- http://rogerdudler.github.io/git-guide/

- http://sweetme.at/2013/09/29/essential-git-commands-cheat-sheet/

- https://zeroturnaround.com/rebellabs/git-commands-and-best-practices-cheat-sheet/
  - http://files.zeroturnaround.com/pdf/zt_git_cheat_sheet.pdf

- http://git-scm.com/book/en/v2

- https://www.codecademy.com/learn/learn-git

- https://try.github.io

Demonstration

Demonstration

- Pay attention!

- I'm too [face] to care about making slides now.

- That, or too [face with wine glass]