

# Step By Step: Guides for Git, Github, and Heroku

# Creating New Git/Github Repositories

1. Go into the directory containing your project
2. Initialize the local git repo
  - `git init`
3. Add all of your project files to the new repo by typing: `git add .`
4. Commit a version of your app to the local repo by with a clear message
  - `git commit -m "initial commit"`
5. Go to github.com and create a new remote repo
6. Link your local repo with your remote repo by typing:
  - `git remote add origin https://github.com/username/<new_repo>.git`
7. Push your local repo to the remote repo by typing:
  - `git push -u origin master`

# Git and Github Workflow

1. Edit and save files locally

Once you have a working version of your application

2. Check local git repo status:
  - `git status`
3. Add updated/new/deleted files to git repo:
  - `git add .`
4. Commit changes to local repo with informative message
  - `git commit -m "informative message"`
5. Push commit to remote repo on Github
  - `git push`

# Undoing Changes Using Git

1. Find commit hash
  - `git log`
2. Revert commit locally:
  - `git revert <commit hash>`
3. Accept or modify reversion comments in
  - Vim: `:wq`
  - GNU ed: `ctrl + X`
4. Push commit to remote repo on Github
  - `git push`



# Git Cheat Sheet

For more awesome cheat sheets  
visit [rebellabs.org](https://rebellabs.org)!



## Create a Repository

From scratch -- Create a new local repository

```
$ git init [project name]
```

Download from an existing repository

```
$ git clone my_url
```

## Observe your Repository

List new or modified files not yet committed

```
$ git status
```

Show the changes to files not yet staged

```
$ git diff
```

Show the changes to staged files

```
$ git diff --cached
```

Show all staged and unstaged file changes

```
$ git diff HEAD
```

Show the changes between two commit ids

```
$ git diff commit1 commit2
```

List the change dates and authors for a file

```
$ git blame [file]
```

Show the file changes for a commit id and/or file

```
$ git show [commit]:[file]
```

Show full change history

```
$ git log
```

Show change history for file/directory including diffs

```
$ git log -p [file/directory]
```

## Make a change

Stages the file, ready for commit

```
$ git add [file]
```

Stage all changed files, ready for commit

```
$ git add .
```

Commit all staged files to versioned history

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

Commit all your tracked files to versioned history

```
$ git commit -am "commit message"
```

Unstages file, keeping the file changes

```
$ git reset [file]
```

Revert everything to the last commit

```
$ git reset --hard
```

## Synchronize

Get the latest changes from origin (no merge)

```
$ git fetch
```

Fetch the latest changes from origin and merge

```
$ git pull
```

Push local changes to the origin

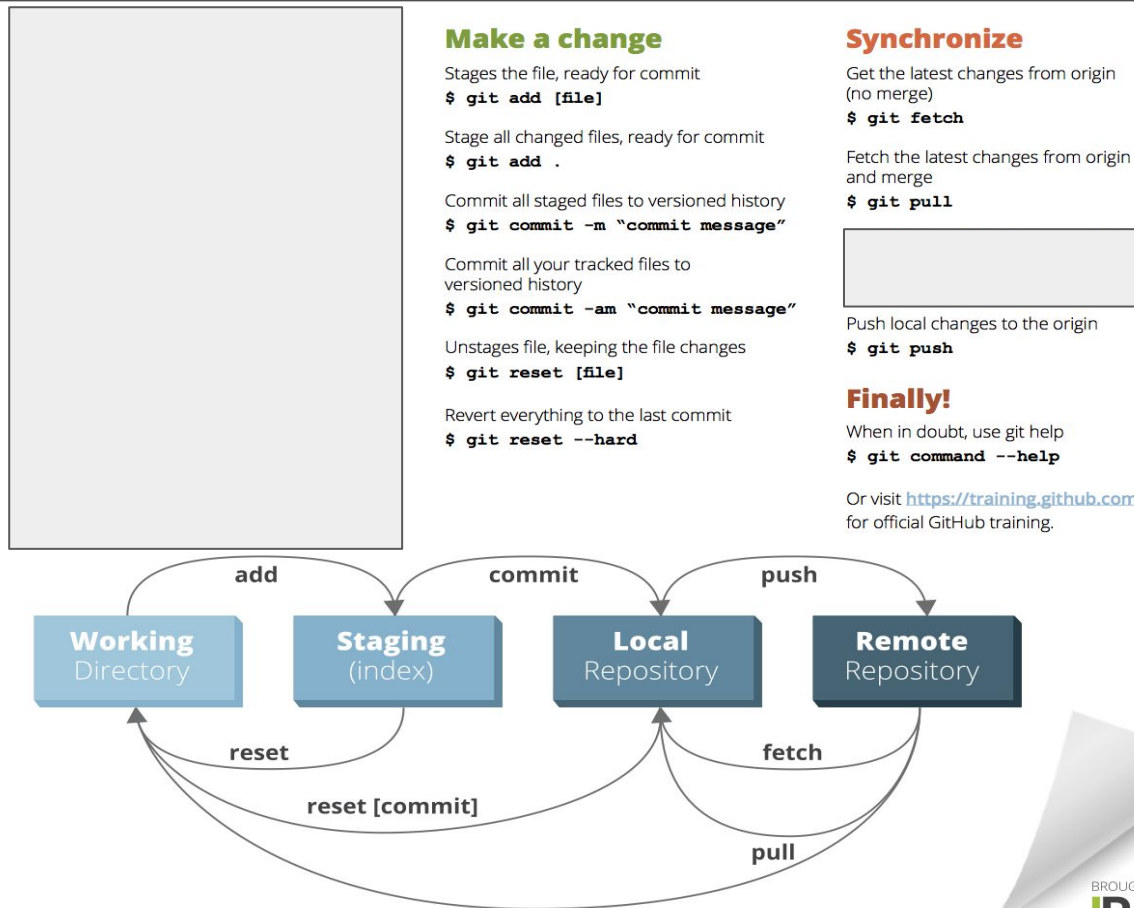
```
$ git push
```

## Finally!

When in doubt, use git help

```
$ git command --help
```

Or visit <https://training.github.com/> for official GitHub training.



BROUGHT TO YOU BY

**Rebel**



# Creating a New Heroku Site

1. Make sure you have a working version of your site locally
2. Commit any new changes to your local and github repos (see git and github workflow slide)
3. Make sure you have `requirements.txt` and `Procfile`
4. Login to Heroku
  - `heroku login`
5. Create the new site on Heroku
  - `heroku create <app-name> --buildpack heroku/python`
6. Push site to remote Heroku repo
  - `git push heroku master`
7. Check your website! (`<app-name>.herokuapp.com`)

# Manually Deploying Changes to Heroku App

1. Make sure you have a working version of your site locally
2. Commit any new changes to your local and github repos  
(see git and github workflow slide)
3. Push site to remote Heroku repo
  - `git push heroku master`
4. Check your website! (<app-name>.herokuapp.com)

# Debugging In Heroku

- When you see application error or internal error when you try to load your website, open the server log files
  - heroku log
- This command prints out the contents of the terminal on the heroku server