■ Complete Jekyll CV Website Setup and Update Guide

Repository: https://github.com/basuastro/basuastro.git

## 1. Prerequisites and Installation

Follow these steps the very first time you are setting up your Jekyll CV site locally on macOS.

### 1.1 Install Command Line Tools

```
xcode-select --install
```

### 1.2 Install Homebrew (macOS package manager)

```
/bin/bash -c "$(curl -fsSL https://raw.githubusercontent.com/Homebrew/install/HEAD/install.sh)"
```

### 1.3 Install Git

```
brew install git
```

### 1.4 Install Ruby via rbenv (to isolate Ruby versions)

```
brew install rbenv ruby-build
echo 'eval "$(rbenv init - zsh)"' >> ~/.zshrc
source ~/.zshrc
rbenv install 3.3.4
rbenv global 3.3.4
ruby -v
```

## 2. GitHub SSH Key Setup

If you already have a key, skip to step 2.3.

### 2.1 Generate a new SSH key

```
ssh-keygen -t ed25519 -C "your_email@example.com"
eval "$(ssh-agent -s)"
ssh-add ~/.ssh/id_ed25519
```

### 2.2 Add your public key to GitHub

Go to GitHub → Settings → SSH and GPG Keys → New SSH key → Paste the contents of:

```
pbcopy < ~/.ssh/id_ed25519.pub
```

### 2.3 Test your SSH connection

```
ssh -T git@github.com
```

You should see: Hi basuastro! You've successfully authenticated.

## 3. Clone and Build Your Site Locally

```
git clone git@github.com:basuastro/basuastro.git
cd basuastro
gem install bundler
bundle install
bundle exec jekyll serve --livereload
```

Open http://127.0.0.1:4000 to preview your site.

## 4. Create the GitHub Actions Workflow

Create a file `.github/workflows/pages.yml` with the following content:

```
name: Deploy Jekyll site to Pages

on:
  push:
    branches: ["main"]
  workflow_dispatch:

permissions:
  contents: read
  pages: write
  id-token: write

concurrency:
  group: "pages"
  cancel-in-progress: true

jobs:
  build:
    runs-on: ubuntu-latest
    steps:
      - uses: actions/checkout@v4
      - uses: actions/configure-pages@v5
        id: pages
      - uses: ruby/setup-ruby@v1
        with:
          ruby-version: '3.3'
          bundler-cache: true
      - run: bundle install --without development
      - run: bundle exec jekyll build --baseurl "${{ steps.pages.outputs.base_url }}"
        env:
          JEKYLL_ENV: production
      - uses: actions/upload-pages-artifact@v3
        with:
          path: _site
  deploy:
    needs: build
    runs-on: ubuntu-latest
    environment:
      name: github-pages
      url: ${{ steps.deployment.outputs.page_url }}
    steps:
      - id: deployment
        uses: actions/deploy-pages@v4
git add .github/workflows/pages.yml
git commit -m "Add GitHub Pages workflow"
git push
```

## 5. Enable GitHub Pages

Go to your repository on GitHub → Settings → Pages → Source → choose 'GitHub Actions' → Save.
After a few minutes, the Action will finish building and your site will go live at:
https://basuastro.github.io/basuastro/ (or https://basuastro.github.io/ if repo is named basuastro.github.io)

## 6. Understanding Your Site Structure

```
_config.yml          → main configuration (title, baseurl, plugins)
_projects/           → individual project markdown files
_pages/              → standalone pages (About, CV, etc.)
_data/navigation.yml → navigation bar entries
assets/img/          → images and thumbnails
```

## 7. When You Make Changes

Any time you modify your CV, projects, or content, follow these steps:

```
1■■ Edit locally (Markdown or YAML files)
   e.g., _projects/new_project.md, _pages/about.md, _data/navigation.yml

2■■ Preview locally before pushing:
   bundle exec jekyll serve --livereload

3■■ Commit and push your updates:
   git add -A
   git commit -m "Update CV or add new project"
   git push

4■■ GitHub Actions automatically rebuilds your site and redeploys it.
   Wait ~2–3 minutes, then refresh your live URL.
```

## 8. Troubleshooting and Tips

- Permission denied (publickey)
  → Ensure your SSH key is added and test with ssh -T git@github.com

- Gem installation errors
  → Use Ruby installed via rbenv, not system Ruby

- Site builds locally but not on GitHub
  → Check Actions logs under the 'Actions' tab

- Broken images or CSS
  → Verify url: and baseurl: in _config.yml

- Tip: Rename your repo to basuastro.github.io for a cleaner URL (no /basuastro suffix).

■ You’re all set!

From now on, your workflow is simply:

Edit → Preview locally → Commit → Push → Wait for GitHub Actions → Site updates live