

The background is a dark blue gradient. On the left, there is a large, semi-transparent circular image of a circuit board. Overlaid on this and the background are several geometric shapes: a blue parallelogram and a green parallelogram in the upper left, and a series of white, stepped, rectangular blocks in the upper right, resembling a microchip or a data structure.

# Building a Docs as Code system

Lisha Li



# About Me

- I live in Vancouver Canada
- 14+ years of experiences writing end-user and developer docs
- Experiences with doc infrastructure design:
  - Built a DITA XML documentation system
  - Built a Confluence documentation system
  - Built a Docs-as-Code documentation system

# Table of Contents

1. [What is Docs as Code?](#)
2. [Why Docs as Code](#)
3. [Manage the Docs](#)
4. [Static Site Generator](#)
5. [Write the Docs](#)
6. [Authoring Tools](#)
7. [Test the Docs](#)
8. [Docs as Code CI/CD workflow](#)
9. [Demo](#)
10. [Summary](#)



# What is Docs as Code?

An approach or system that you create and maintain your documentation using the same tools as you do with your programming code.



# Why Docs as Code?

- Keep your documentation in sync with your code
- Use the same tools to review code and documentation changes
- Writers and developers can work together
- Automated testing, building, and publishing



# Manage the Docs

Manage your documentation using a source control system like:

- GitHub
- Bitbucket
- GitLab



# Static Site Generator

A static site generator (SSG) creates HTML pages from a template and a given content source. Most static site generators accept markup language such as Markdown as a source.

The following website lists most of the SSG, their popularity and the templates they use:

<https://jamstack.org/generators/>

To customize the look of the HTML pages:

- CSS
- JavaScript
- Customize the default template
- Create your own template



# Write the Docs

Write the documentation in plain text markup and build the output with a SSG:

- Markdown -> Almost any SSG  
(<https://jamstack.org/generators/>)
- reStructuredText -> Sphinx (for Python)
- AsciiDoc -> Antora/ some SSG with plugins





# Authoring Tools

- Any text editor
  - Visual Studio Code (<https://code.visualstudio.com/>)
  - Atom (<https://atom.io/>)
  - Notepad++ (<https://notepad-plus-plus.org/>)
- Online IDE tool
  - Gitpod (<https://www.gitpod.io/>)

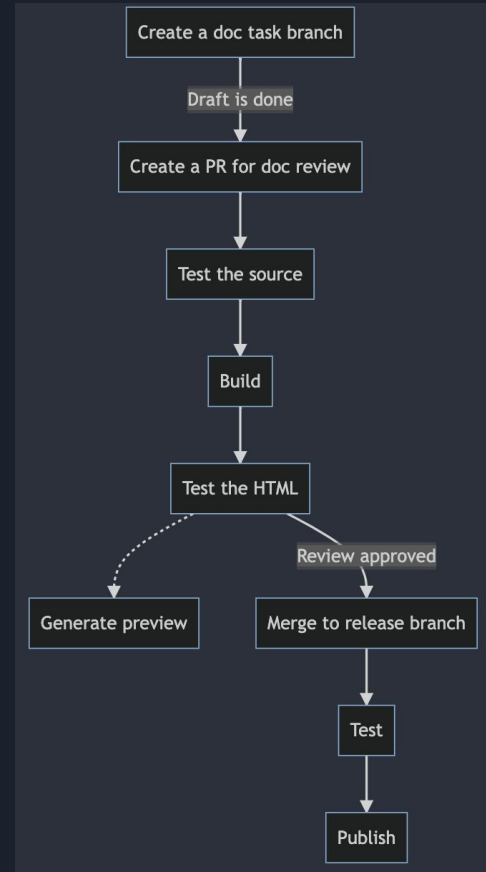


# Test the Docs

- Style and prose linter
  - Vale (<https://vale.sh/docs/>)
- HTML checker
  - Htmltest (<https://github.com/wjdp/htmltest>)
  - Muffet (<https://github.com/ravivqge/muffet>)
  - HTMLProofer (<https://github.com/gitorikian/html-proofer>)
  - LinkChecker (<https://github.com/linkchecker/linkchecker/>)
- Code linter
  - Pylint (<https://pylint.pycqa.org>)
  - JSLint (<https://www.jshint.com/>)

# Docs as Code CI/CD Workflow

- CI refers to continuous integration.
  - An automation process that new code/doc changes are regularly built, tested, and merged to a shared repository.
- CD refer to continuous deployment.
  - An automation process to release the code/doc changes from the repository to production.





# Demo

I will do a demo using a docs as code demo system I built in GitHub

<https://github.com/lilisha100/docs-as-code-demo>

- DocFX to build developer guide/client and REST API references
- Content source in markdown and from docfx-seed project
- Vale
- Htmltest
- Gitpod for doc editing and preview
- GitHub Pages to host the doc online
- GitHub Action to automate the workflow



# Summary

Things you need to set up a docs as code system:

- Git repo
- Markup language
- Authorizing tools
- Static Site Generator
- Doc testing tools
- Website design
- Website hosting
- CI/CD workflow



# Thank you!

LinkedIn: <https://www.linkedin.com/in/lishali/>

Email: [lilisha100@hotmail.com](mailto:lilisha100@hotmail.com)