

How to write horrible documentation

Engineering

PyIstanbul #PythonSaatı
May 25, 2021

Pradyun Gedam
Software Engineer, Python Infrastructure

TechAtBloomberg.com

Bloomberg

About Me

- @pradyunsg on Twitter, GitHub, Mastodon, etc.
- Curator of sphinx-themes.org
- Author of multiple **Sphinx extensions**
- Author of **Furo**, a modern Sphinx theme
- Maintainer of **pip**, packaging and more
- Moderator on **PyPI**

TechAtBloomberg.com

© 2021 Bloomberg Finance L.P. All rights reserved.

Bloomberg
Engineering

Where to get these slides

<https://pradyunsg.me/talks/>

How to write horrible documentation

All the “documentation” stuff is a waste of time anyway.

Quick Guide

- Make discovery of information more difficult
- Leave gaps
- Provide inaccurate information
- Assume things about the reader
- Make it difficult to understand
- Inconsistency is the secret sauce
- Make it look and feel ancient
- Make it difficult to work with

Make discovery of information more difficult

What if... you can't find something you know is documented?

- Spread information across multiple pages
- Don't provide a clear separation of content
- Have long pages, with lots of unrelated, but useful-in-context information

Make discovery of information more difficult

Search...

- Don't provide it
- Make it so bad that it's basically useless
- Make it difficult to locate or use
- Require exact matches

Leaves gaps

We shouldn't document **everything**.

Let's abuse that!

- Omit details that are relevant, but “not important”
- Include references to things that aren’t documented
- Works best with a bad site structure

Provide inaccurate information

Make “mistakes” and hide them

- Use paragraphs to describe step-by-step actions
- Provide slightly incorrect descriptions
- doctests can prevent you from having incorrect code snippets

Provide inaccurate information

Avoid discussing pitfalls

- Make warnings obscure by burying them in text
- Leave it unclear which option is preferred
- Don't document that a certain approach is deprecated

Make assumptions about the reader

Assume who they are

- Gendered references
- Accessibility

Make assumptions about the reader

Assume what they know

- Keep it unclear what the expected background knowledge is
- Jargon!
 - Serves as shorthand for “well-understood” concepts
 - Makes it clear that the author knows the nuances and details
- Acronyms!

Make it difficult to understand

Mix “levels” of information

- Mix details when discussing high-level concepts
- Jargon helps obscure this

Make it difficult to understand

Glossary pages!

- Define terms with “simple” words
- Cross-reference other terms in definitions

Inconsistency is the secret sauce

Use a mix of writing styles

- You can do this by using MyAwesomeTomato to get red color
- Using MyAwesomeCarrot, it is possible for a user to get orange colours

Inconsistency is the secret sauce

Have multiple “sources of truth” for information

- Can you slip in a contradiction? Even better!

Inconsistency is the secret sauce

Include typos!

- Bonus points if things have very similar names (PyPI, PyPy, mypy)
- Use multiple name styles
 - Pip is a package manager
 - pip comes installed with Python
 - PIP has 6 maintainers

Make it look and feel ancient

Modern web design guidelines are a fad

- Odd font *choices*
- Difficult to read font sizes
- Non-responsive design

Make it look and feel ancient

Use “bad” site designs

- Poor navigational capabilities
- Slightly misaligned elements
 - to annoy those with a web design background
- Low contrast colour choices

Make it difficult to work with

Introduce complexity in the workflows

- “Custom” documentation generators and build pipelines!
 - Avoid standard tooling or make it difficult to use
- Create fragile “helpers”
- Unclear build instructions

Make it difficult to work with

Introduce complexity in the content

- Treat documentation like code and make it DRY
- Use custom HTML to make *that one thing* look nice

Conclusion

There are lots of ways to make documentation worse

Many of these happen organically too

You now know how to spot these patterns

Shoutouts!

CGPGrey – “7 Ways to Maximize Misery 😞”

<https://www.youtube.com/watch?v=LO1mTELoj6o>

Eric Holscher – “A beginner’s guide to writing documentation”

<https://www.writethedocs.org/guide/writing/beginners-guide-to-docs/>

Daniele Procida – “A Systematic Framework for Technical Documentation Authoring”

<https://diataxis.fr/>

Sviatoslav Sydorenko – helped me come up with this talk’s concept!

<https://github.com/webknjaz>

Bloomberg

Engineering

Questions?

TechAtBloomberg.com

© 2021 Bloomberg Finance L.P. All rights reserved.

Thanks!

<https://www.bloomberg.com/careers>

Engineering

Bloomberg

TechAtBloomberg.com