

Documentation

November 2022 | Online | Wouter Klijn

Overview

- Homepage or Github.io landing page
- Tutorials
- How-to guides
- Technical reference and API documentation

Homepage or Github.io landing page

- First contact, should answer:
 - What is the software?
 - Is it important for me, should I care?
 - How much effort is it to learn?
 - Further reading:
 - License
 - Bug tracker
 - Source code
 - Further documentation
 - Installation details
- Download (button)!**

Tutorials

- First entry point for new users
 - Check technical level with a real (new) user
 - You as a developer cannot assess the level

The very first tutorial

- No rocket science
- Do one thing, from beginning to end
- Online live coding example: google collab

Give the user a working example within 30 minutes

<http://blog.parse.com/learn/engineering/designing-great-api-docs/>
<https://jacobian.org/writing/what-to-write/>

How-to guide

- How to perform a specific action or function
- Do not double information
 - Refer back to tutorials and how-to with hyperlinks
- In depth and comprehensive
- Not a dry summations of functions
 - It is not an API documentation

Technical/Reference/API Documentation

- The details of your project
- List all your public functions and classes
 - Name
 - Arguments
 - Description
 - (Source file, inheritance hierarchy, example)
- Typically generated from source-code
- Expensive to get good
 - E-mail to developer can be an alternative for some (smaller, internal) projects?

Further reading

- Other interesting sources of documentation:
 - Source code
 - Issue tracker
 - Internal chat server
 - Meeting notes
 - Email
 - **Tests !!!**

<https://medium.com/@MrJamesFisher/documentation-black-holes-facd0c3b9ed7>

- Articles series on documentation:

<https://jacobian.org/writing/great-documentation/>

Questions