Documentation

Developed programs should be documented at multiple levels, from code comments, through API documentation, to installation and usage documentation. Comments at each level should take into account different target audience, from experienced developers, to end users with no programming skills.

Example of good documentation: A Guide to NumPy/SciPy Documentation

Markdown

Markdown is a lightweight markup language that allows you to create webpages, wikis and user documentation with a minimum of effort. Documentation written in markdown looks exactly like a plain-text document and is perfectly human-readable. In addition, it can also be automatically converted to HTML, latex, pdf, etc. More information about markdown can be found here:

http://daringfireball.net/projects/markdown/

http://en.wikipedia.org/wiki/Markdown

Retext is a markdown aware text editor, that can be used to edit markdown files and convert them into HTML or PDF. It can be found at:

https://github.com/retext-project/retext

Alternatively, 'pandoc' is a command line utility that can convert markdown documents to into several other formats (including latex):

http://johnmacfarlane.net/pandoc/

An Eclipse plugin for previewing the HTML generated by markdown is available on this page:

https://marketplace.eclipse.org/content/markdown-text-editor

Readme

Clear explanation of the goal of the project with pointers to other documentation resources.

Use GitHub flavoured markdown for, e.g., syntax highlighting. (If reStructuredText or another format that GitHub renders is idiomatic in your community, use that instead.) README is targeted towards developers, it is more technical than home page. Keeping basic documentation in README.md can be even useful for lead developer, to track steps and design