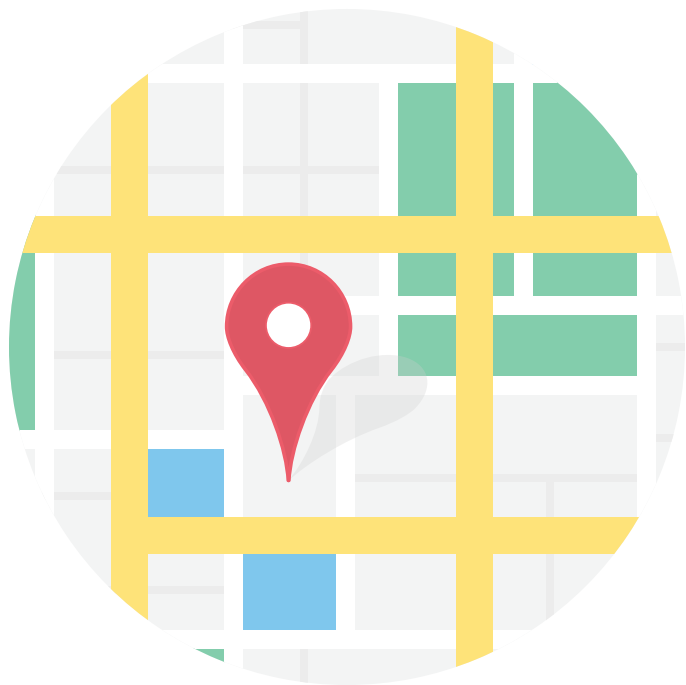


Desenvolvimento Aberto

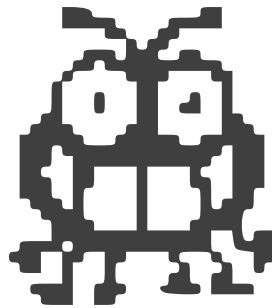
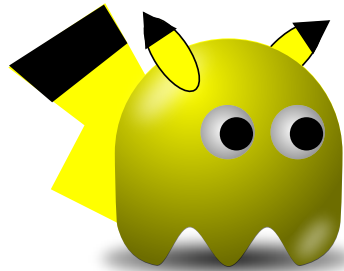


Documentação - parte 1

Igor dos Santos Montagner (igorsm1@insper.edu.br)

Avisos

Hoje acaba oficialmente o prazo para as duas skills abaixo.



Se você já terminou envie o PR para o repositório da disciplina.

Se não terminou é hora de se preocupar.

Documentação de software

Documentação de usuário:

Documentação de desenvolvimento:

Documentação de software

Documentação de usuário:

- Instalação
- Funcionalidades
- Onde obter ajuda

Documentação de software

Documentação de desenvolvimento:

- Como compilar (dependências, ferramentas usadas, etc)
- Como testar (dependências, ferramentas usadas, etc)
- Estilo de código e outras orientações relacionadas
- Organização do código e arquitetura da aplicação

Documentação de software

Às vezes as coisas se confundem! O que vocês colocariam na documentação de usuário do *Flask*?

E na de desenvolvedor?

Exemplo: Spyder



HOME BLOG

[Navigation](#)

[Overview](#)

[Installation](#)

[Command Line Options](#)

[Editor](#)

[IPython Console](#)

[Variable Explorer](#)

[Help](#)

[Debugging](#)

[Static Code Analysis](#)

[Profiler](#)

[Projects](#)

[File Explorer](#)

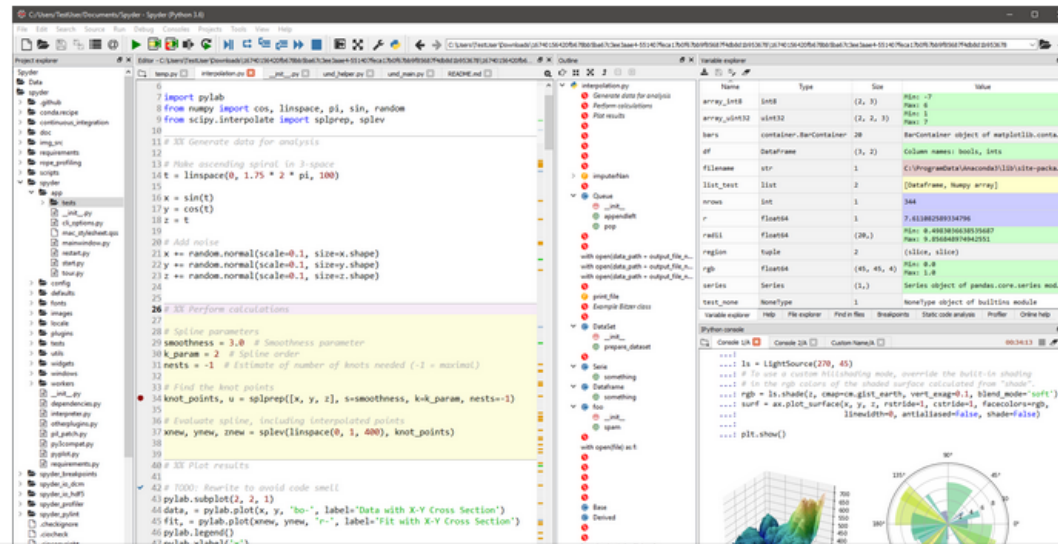
[Find in Files](#)

[Online Help](#)

[History Log](#)

[Internal Console](#)

Spyder: The Scientific Python Development Environment — Documentation



[<https://pythonhosted.org/spyder/>]













Exemplo: Spyder

Welcome to the Spyder IDE Wiki!

Spyder is a powerful interactive development environment for the Python language with advanced editing, interactive testing, debugging and introspection features.

This wiki contains

-  [Contributing to spyder](#)
-  [Current Funding and Development Status](#)
-  [Development information](#)
-  [Troubleshooting Guide and FAQ](#)
-  [SEPs: Spyder Enhancement Proposals](#)
-  [Roadmap](#)
-  [Projects using Spyder](#)
-  [Frequently asked questions](#)
-  [Plugin ideas](#)
-  [User Plugins](#)

▼ Pages **42**

[Home](#)

[Anaconda stopped funding Spyder](#)

[API decoupling](#)

[Beta version changelog](#)

[Changelog](#)

[Contributing to Spyder](#)

[Current Funding and Development Status](#)

[Dev: Coding Style](#)

[Dev: Cookbook](#)

[Dev: Debugging Spyder](#)

[Dev: Documentation](#)

[Dev: Github Workflow](#)

[Dev: Index](#)

[Dev: Spyder Internals](#)

Sistemas de documentação



SPHINX
Python Documentation Generator

Home

Get it

Docs

Extend/Develop

Welcome

Sphinx is a tool that makes it easy to create intelligent and beautiful documentation, written by Georg Brandl and licensed under the BSD license.

It was originally created for [the Python documentation](#), and it has excellent facilities for the documentation of software projects in a range of languages. Of course, this site is also created from reStructuredText sources using Sphinx! The following features should be highlighted:

- **Output formats:** HTML (including Windows HTML Help), LaTeX (for printable PDF versions), ePub, Texinfo, manual pages, plain text
- **Extensive cross-references:** semantic markup and automatic links for functions, classes, citations, glossary terms and similar pieces of information
- **Hierarchical structure:** easy definition of a document tree, with automatic links to siblings, parents and children
- **Automatic indices:** general index as well as a language-specific module indices
- **Code handling:** automatic highlighting using the [Pygments](#) highlighter
- **Extensions:** automatic testing of code snippets, inclusion of docstrings from Python modules (API docs), and [more](#)
- **Contributed extensions:** more than 50 extensions [contributed by users](#) in a second repository; most of them installable from PyPI

Sphinx uses [reStructuredText](#) as its markup language, and many of its strengths come from the power and straightforwardness of reStructuredText and its parsing and translating suite, the [Docutils](#).

What users say:

“Cheers for a great tool that actually makes programmers **want** to write documentation!”

A  project

Download

Current version: `pypi v1.7.7`

Install Sphinx with:

```
pip install -U Sphinx
```

Questions?

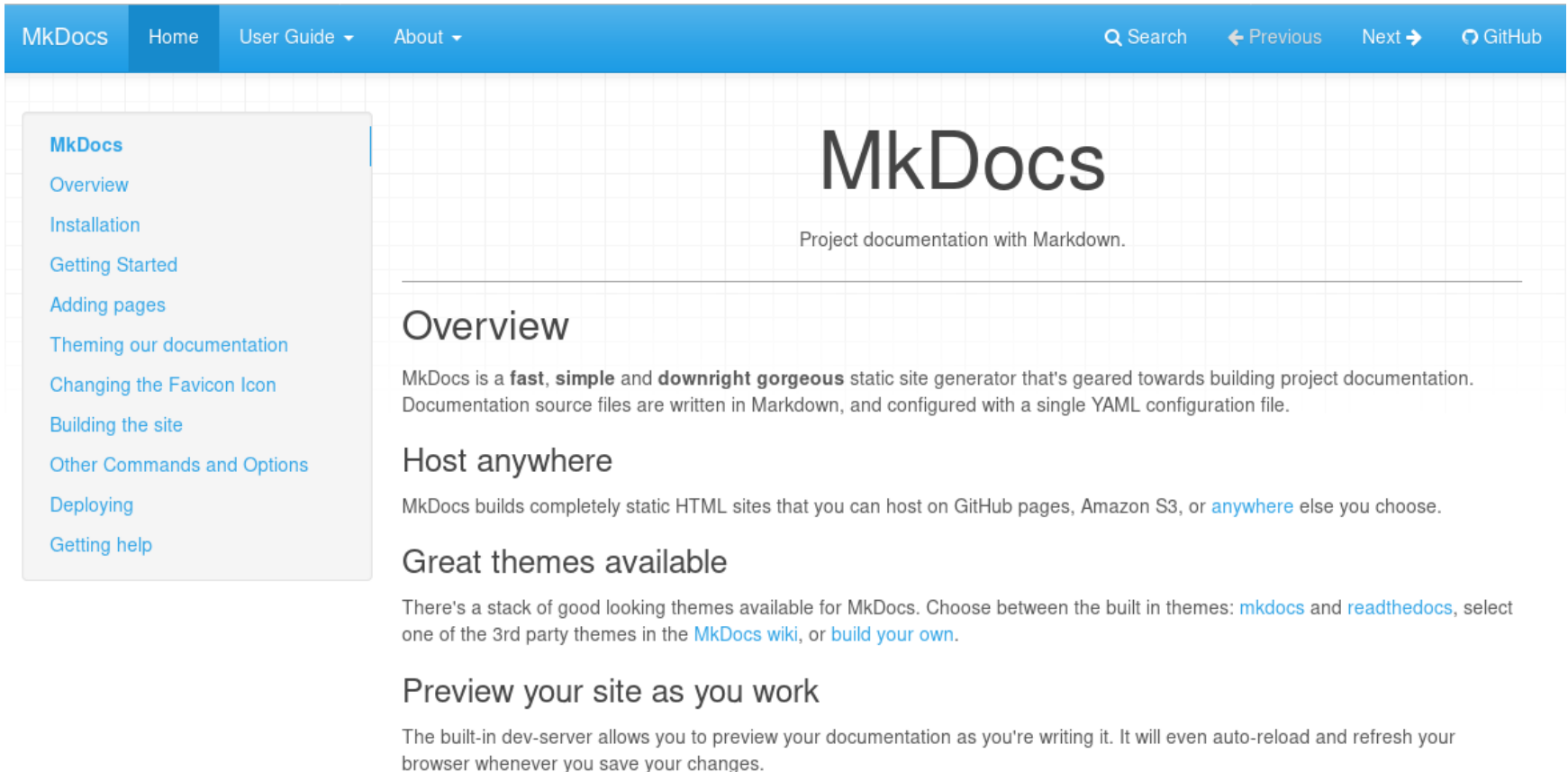
Suggestions?

Join the [sphinx-users](#) mailing list on Google Groups:

or come to the `#sphinx-doc` channel on FreeNode.

Sphinx

Sistemas de documentação



The screenshot shows the MkDocs website interface. At the top is a blue navigation bar with links for 'MkDocs', 'Home', 'User Guide', and 'About'. On the right side of the bar are 'Search', 'Previous', 'Next', and 'GitHub' links. A left sidebar contains a list of navigation items: 'MkDocs', 'Overview', 'Installation', 'Getting Started', 'Adding pages', 'Theming our documentation', 'Changing the Favicon Icon', 'Building the site', 'Other Commands and Options', 'Deploying', and 'Getting help'. The main content area features a large 'MkDocs' title and the subtitle 'Project documentation with Markdown.' Below this, there are sections for 'Overview', 'Host anywhere', 'Great themes available', and 'Preview your site as you work', each with a brief descriptive paragraph.

MkDocs

Project documentation with Markdown.

Overview

MkDocs is a **fast, simple** and **downright gorgeous** static site generator that's geared towards building project documentation. Documentation source files are written in Markdown, and configured with a single YAML configuration file.

Host anywhere

MkDocs builds completely static HTML sites that you can host on GitHub pages, Amazon S3, or [anywhere](#) else you choose.

Great themes available

There's a stack of good looking themes available for MkDocs. Choose between the built in themes: [mkdocs](#) and [readthedocs](#), select one of the 3rd party themes in the [MkDocs wiki](#), or [build your own](#).

Preview your site as you work

The built-in dev-server allows you to preview your documentation as you're writing it. It will even auto-reload and refresh your browser whenever you save your changes.

Mkdocs

Sistemas de documentação



Read the Docs

Create, host, and browse documentation.

Sign up

or [Log in](#)

Technical documentation lives here

Read the Docs simplifies software documentation by automating building, versioning, and hosting of your docs for you.

Free docs hosting

We will host your documentation for free forever. There are no tricks. We help 89,019 open source projects share their docs.

Webhooks

Whenever you push code to your favorite version control system, whether that is Git, Mercurial, Bazaar, or Subversion, we will automatically build your docs so your code and documentation are never out of sync.

<https://readthedocs.org/>

Atividade de Hoje

Vocês receberão um *zip* com o código de um software e zero instruções. Vocês deverão:

1. Aprender como rodar o software
2. Documentar os passos que vocês seguiram para fazê-lo
3. Fazer uma análise crítica do projeto com relação a
 - Arquitetura
 - Qualidade do código
 - Segurança da solução
4. Criar uma documentação de usuário e uma de desenvolvimento baseado em suas anotações

Objetivos desta atividade

1. Entender na prática a necessidade de documentação;
2. Praticar suas habilidades de leitura de código e entendimento da arquitetura de uma aplicação desconhecida;
3. Obter primeira experiência com documentação de desenvolvimento e de usuário.

Não são objetivos desta atividade

1. Mostrar nenhum tipo de boa prática de programação
2. Melhorar o software analisado
3. Criticar o desenvolvedor que trabalhou antes no projeto

Agora

Vamos fazer o roteiro **pelo menos até a pergunta 5**:

1. Aprender como rodar o software
2. Documentar os passos que vocês seguiram para fazê-lo

Duração: 30~45 minutos

Agora

3. Fazer uma análise crítica do projeto com relação a
 - Arquitetura
 - Qualidade do código
 - Segurança da solução

Duração: 15 minutos para reolhar o código, 15 minutos para discussões.

Para casa

4. Criar uma documentação de usuário e uma de desenvolvimento baseado em suas anotações



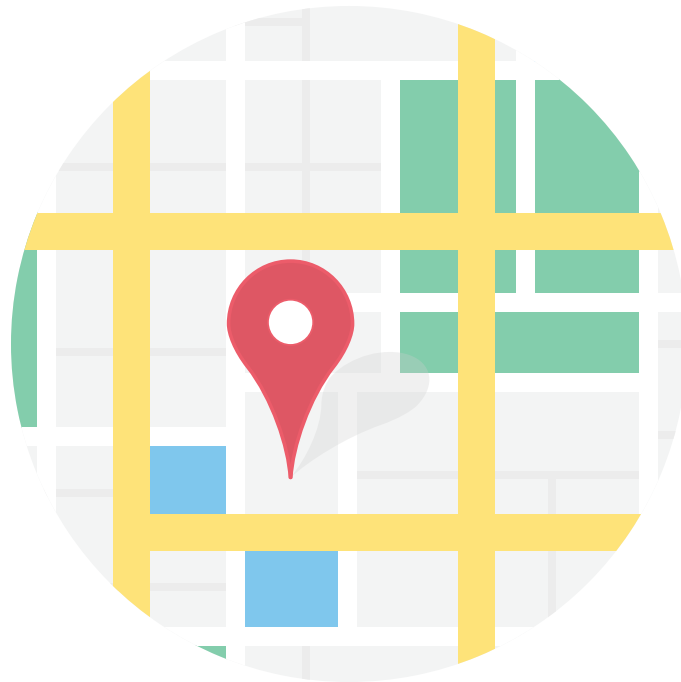
Skill *Caminho encontrado*: entender como um software sem documentação funciona e criar um guia de uso.

Mais detalhes da entrega no roteiro físico.

Próxima aula

- Apresentação das documentações criadas
- Tradução de software: Internacionalização e localização

Desenvolvimento Aberto



Documentação

Igor dos Santos Montagner (igorsm1@insper.edu.br)