Documenting Your Python Project: A Beginner's Guide

Making Your Code Understandable and Maintainable

Michael Borck

Table of contents

Why Document Your Project?

- Improve code readability
- Help others understand your code
- Facilitate maintenance and collaboration

Why Use Docstrings?

- Documentation: Provides clear explanations of your code's functionality.
- Maintainability: Makes it easier for others (and yourself) to understand and maintain the code.
- Automatic Documentation: Tools like MkDocs and Sphinx can generate documentation from docstrings.
- Testing: With `doctest`, you can include examples that serve as both documentation and tests.

What are Docstrings?

- Inline documentation for modules, classes, functions, and methods
- Written as triple-quoted strings

Example: Function Docstring

```python
def add(a, b):
    """
    Add two numbers and return the result.

    Args:
        a (int or float): The first number.
        b (int or float): The second number.

    Returns:
        int or float: The sum of `a` and `b`.

    Examples:
    >>> add(2, 3)
    5
    >>> add(1.5, 2.5)
    4.0
    """
    return a + b
```

Writing a README.md

Purpose of README.md - Provide an overview of your project - Explain how to install and use it - Highlight key features and dependencies

Example: README.md

~~~
# Project Title

## Overview
Brief description of your project.

## Installation
```bash
pip install your_project
~~~

Usage

```python
from your_project import your_function
result = your_function()
print(result)
```

Features

- Feature 1
- Feature 2

License

MIT

~~~
# Introducing MkDocs

**What is MkDocs?**
- Simple static site generator for project documentation
- Easy to setup and use

# Installing MkDocs

**Install MkDocs**
```bash
pip install mkdocs
~~~

Setting Up MkDocs

Create MkDocs Project

```
mkdocs new my_project_docs
cd my_project_docs
```

Editing mkdocs.yml

Basic mkdocs.yml Configuration

```yaml
site_name: My Project Documentation

nav:
    - Home: index.md
    - API Reference: api.md
```

Writing Documentation Pages

Example: index.md

```
# Welcome to My Project

This is the documentation for My Project.

## Getting Started

Follow these instructions to get started.
```

Serving Documentation Locally

Serve Documentation

```
mkdocs serve
```

- Access the documentation locally at `http://127.0.0.1:8000`

Building the Documentation

Build Static Site

```
mkdocs build
```

- Generates static site in the `site/` directory

Summary

- Use **docstrings** for inline documentation
- Create a **README.md** for project overview
- Use **MkDocs** for project documentation

Next Session

- Focus on finalising the project.
- Distribution methods.