UNIT 6.3 GRADED ASSIGNMENT

Group members

Ifra Saleem (2303.khi.deg.003)
Umaima Siddiqui (2023.KHI.DEG.033)

UNIT 6.3 GRADED ASSIGNMENT

Task:

- Provide a template Sphinx command line command which auto documents your project. Explain what each flag and each element of your command do.
- What flag should you add if you use implicit namespaces in your project?

Solution:

The activity folder is attached with this file in which we use template Sphinx command line command which auto documents our project.

- Provide a template Sphinx command line command which auto documents your project. Explain what each flag and each element of your command do.

Command:

```
sphinx-apidoc  -o <OUTPUT_PATH> <MODULE_PATH> [EXCLUDE_PATTERN …] [OPTIONS]
```

- **sphinx-apidoc** is a tool that is used for the automatic generation of Sphinx resources using the autodoc extension. It is used to create API documentation. If any modules have side effects on import, these will be executed by autodoc when sphinx-build is run.
- **-o <OUTPUT_PATH>:** Here we mention the output directory where the generated documentation will be stored.
- **<MODULE_PATH>:** Here we mention the source directory where python modules/packages are located.
- **[EXCLUDE_PATTERN …]:** This allows us to specify the patterns that we want to exclude from API documentation generation.
- **[OPTIONS]:** We can add additional options to customize the documentation generation process. -f to overwrite existing files, -P to generate an HTML table of contents, and -M to include module-level docstrings.

- What flag should you add if you use implicit namespaces in your project?

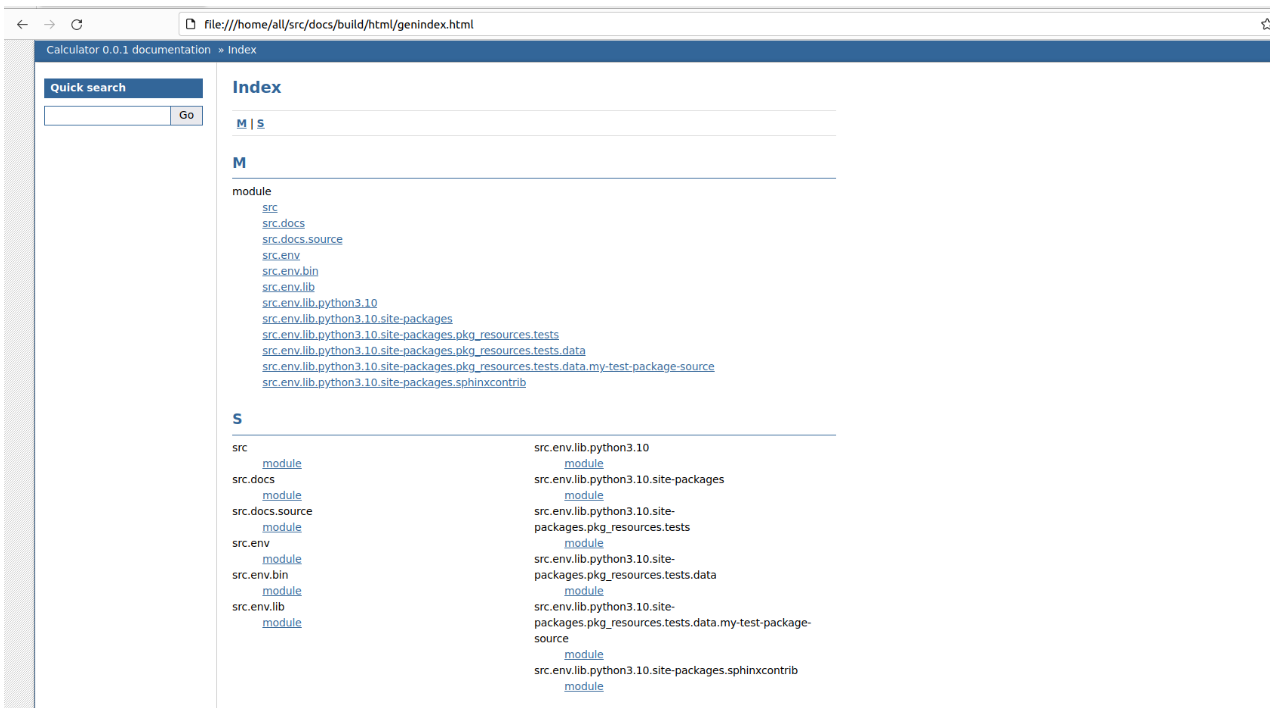
Command:

sphinx-apidoc -o <output_dir> <source_dir> [exclude_patterns] --implicit-namespaces [options]

We can use –e or --implicit-namespaces flag to Sphinx line command. By default, sphinx-apidoc processes sys.path searching for modules only. Python 3.3 introduced **PEP 420** implicit namespaces that allow module path structures such as `foo/bar/module.py` or `foo/bar/baz/__init__.py` (notice that `bar` and `foo` are namespaces, not modules).

This flag enables support for implicit namespaces, where submodules and subpackages are automatically recognized and documented as separate entities.

Output we got by running firefox ./build/html/index.html & in the activity.



file:///home/all/src/docs/build/html/genindex.html

Calculator 0.0.1 documentation » Index

Quick search

Go

Index

M | S

M

module
- src
- src.docs
- src.docs.source
- src.env
- src.env.bin
- src.env.lib
- src.env.lib.python3.10
- src.env.lib.python3.10.site-packages
- src.env.lib.python3.10.site-packages.pkg_resources.tests
- src.env.lib.python3.10.site-packages.pkg_resources.tests.data
- src.env.lib.python3.10.site-packages.pkg_resources.tests.data.my-test-package-source
- src.env.lib.python3.10.site-packages.sphinxcontrib

S

src
- module

src.docs
- module

src.docs.source
- module

src.env
- module

src.env.bin
- module

src.env.lib
- module

src.env.lib.python3.10
- module

src.env.lib.python3.10.site-packages
- module

src.env.lib.python3.10.site-packages.pkg_resources.tests
- module

src.env.lib.python3.10.site-packages.pkg_resources.tests.data
- module

src.env.lib.python3.10.site-packages.pkg_resources.tests.data.my-test-package-source
- module

src.env.lib.python3.10.site-packages.sphinxcontrib
- module