EPAM Python Software Engineer Training

Lesson 2: RST document format

Contents

1 Course

1.1 reStructuredText markup language

A **reStructuredText** is a flexible simple yet powerful markup language used as a documentation standard in many companies and organizations (e.g. Cisco, Google, IETF, PSF, etc). There are many tools allowing to generate different documents from **reST** format (e.g. HTML, PDF, Latex, etc). Some tools enhance or restrict specific **reST** features for their needs.

Acknowledge with **reST** markup language basics at an official page.

1.2 Python doc-strings

A simplified form of **reStructuredText** is used for **python doc-strings**. There are a lot of tools to generate an exhaustive documentation out of doc-strings. A PSF Documentation was generated this way.

Read PEP 257 about **python doc-string** conventions.

PEP 257 is not enough strict, hence, there are many corporate conventions built on top of it. I recommend to stick to Google doc-string conventions.

1.3 docutils package

A **docutils** is a python package to generate documentation out of **reST** format and, in particular, off **python doc-strings**. There are other popular tools to do this task, like *epidoc*, *Sphinx* etc.

Install **docutils** using *sudo pip install docutils* command.

1.4 sphinx project

A **Sphinx** is a new mainstream documentation standard for Python. It elaborates an **reST** format from **docutils** and extends it with a rich set of directives.

Personally, I don't like its documentation syntax and am not very familiar with it, but very likely that on your next project it will be required to use **Sphinx**, hence, it is mandatory for a Python developer to know about it.

2 Tasks

2.1 Truth Table

Create an **reST** document containing truth tables for a set of boolean expressions. It should be possible to convert this document to both HTML and PDF without parser errors. A result should have a proper structure with a heading, table of contents, sections, well-formatted tables.

The following boolean expressions should be considered:

- a and b or c
- (a and b) or c
- a and (b or c)
- a or not (b and a) or (not c or b)
- (a or not b) and (c or not a)