doc8

unknown

CONTENTS

Contents:

DOC8

doc8 is an *opinionated* style checker for rst (with basic support for plain text) styles of documentation.

1.1 Quick start

```
pip install doc8
```

To run *doc8*, just invoke it against any documentation directory:

```
$ doc8 cool-project/docs
```

1.2 Usage

```
$ doc8  -h

usage: doc8 [-h] [--config path] [--allow-long-titles] [--ignore code]
            [--no-sphinx] [--ignore-path path] [--ignore-path-errors path]
            [--default-extension extension] [--file-encoding encoding]
            [--max-line-length int] [-e extension] [-v] [--version]
            [path [path ...]]

Check documentation for simple style requirements.

What is checked:
    - invalid RST format - D000
    - lines should not be longer than 79 characters - D001
      - RST exception: line with no whitespace except in the beginning
      - RST exception: lines with http or https urls
      - RST exception: literal blocks
      - RST exception: rst target directives
    - no trailing whitespace - D002
    - no tabulation for indentation - D003
    - no carriage returns (use unix newlines) - D004
    - no newline at end of file - D005

positional arguments:
  path                    Path to scan for doc files (default: current
```

```
                            directory).

optional arguments:
  -h, --help              show this help message and exit
  --config path           user config file location (default: .config/doc8.ini, doc8.ini,
→tox.ini,
                          pep8.ini, setup.cfg).
  --allow-long-titles     allow long section titles (default: false).
  --ignore code           ignore the given error code(s).
  --no-sphinx             do not ignore sphinx specific false positives.
  --ignore-path path      ignore the given directory or file (globs are
                          supported).
  --ignore-path-errors path
                          ignore the given specific errors in the provided file.
  --default-extension extension
                          default file extension to use when a file is found
                          without a file extension.
  --file-encoding encoding
                          set input files text encoding
  --max-line-length int
                          maximum allowed line length (default: 79).
  -e extension, --extension extension
                          check file extensions of the given type (default:
                          .rst, .txt).
  -q, --quiet             only print violations
  -v, --verbose           run in verbose mode.
  --version               show the version and exit.
```

1.2.1 INI file usage

Instead of using the CLI for options the following files will also be examined for `[doc8]` sections that can also provide the same set of options. If the `--config path` option is used, these files will **not** be scanned for the current working directory and that configuration path will be used instead.

- `$CWD/doc8.ini`

- `$CWD/.config/doc8.ini`

- `$CWD/tox.ini`

- `$CWD/pep8.ini`

- `$CWD/setup.cfg`

- `$CWD/pyproject.toml`

An example section that can be placed into one of these files:

```
[doc8]

ignore-path=/tmp/stuff,/tmp/other_stuff
max-line-length=99
verbose=1
ignore-path-errors=/tmp/other_thing.rst;D001;D002
```

Note: The option names are the same as the command line ones (with the only variation of this being the `no-sphinx` option which from the configuration file will be `sphinx` instead).

1.2.2 Option conflict resolution

When the same option is passed on the command line and also via configuration files the following strategies are applied to resolve these types of conflicts.

Option	Overrides	Merges
`allow-long-titles`	Yes	No
`ignore-path-errors`	No	Yes
`default-extension`	Yes	No
`extension`	No	Yes
`ignore-path`	No	Yes
`ignore`	No	Yes
`max-line-length`	Yes	No
`file-encoding`	Yes	No
`sphinx`	Yes	No

Note: In the above table the configuration file option when specified as *overrides* will replace the same option given via the command line. When *merges* is stated then the option will be combined with the command line option (for example by becoming a larger list or set of values that contains the values passed on the command line *and* the values passed via configuration).

1.3 API

It is also possible to use *doc8* programmatically. To call *doc8* from a Python project, use:

```python
from doc8 import doc8

result = doc8(allow_long_titles=True, max_line_length=99)
```

The returned `result` will have the following attributes and methods:

- `result.files_selected` - number of files selected
- `result.files_ignored` - number of files ignored
- `result.error_counts` - dict of {`check_name:  error_count`}
- `result.total_errors` - total number of errors found
- `result.errors` - list of (`check_name, filename, line_num, code, message`) tuples
- `result.report()` - returns a human-readable report as a string

The `doc8` method accepts the same arguments as the executable. Simply replace hyphens with underscores.

Note: Calling `doc8` in this way will not write to stdout, so the `quiet` and `verbose` options are ignored.

TWO

INSTALLATION

At the command line:

```
$ pip install doc8
```

Or, if you have virtualenvwrapper installed:

```
$ mkvirtualenv doc8
$ pip install doc8
```

CONTRIBUTING

Before contributing to *doc8* or any other PyCQA project, we suggest you read the PyCQA meta documentation:

http://meta.pycqa.org/en/latest/

Patches for *doc8* should be submitted to GitHub, as should bugs:

https://github.com/pycqa/doc8

FOUR

INDICES AND TABLES

- genindex
- modindex
- search