

# Documentation and Continuous Integration

Simon Pfreunschuh

September 23, 2020

## 1 Updating your weather\_app code.

- Update your clone of [https://github.com/simonpf/weather\\_app](https://github.com/simonpf/weather_app) repository from the last lecture.
- The code has been updated and now contains a command-line application implemented by the function `smhpy` in `weather_app.app`.
- Make Python install the command-line application by adding the following lines to your `setup.py`:

```
setuptools.setup(  
    ...  
    entry_points={  
        "console_scripts": [  
            "smhpy=weather_app.app:smhpy"  
        ]  
    }  
    ...  
)
```

- Reinstall the package.
- Test the installation by running `smhpy` from the command line. If that doesn't work you may have to add `~/.local/bin` to your `PATH` environment variable:

```
$ export PATH = ~/.local/bin:${PATH}
```

## 2 Adding Sphinx documentation

- Install Sphinx
- Follow the instruction from the lecture (or course homepage) to initialize the Sphinx documentation.
- Add the pages `installtion.rst`, `usage.rst` and `api_reference.rst` and reference them from `index.rst`
- Build the HTML documentation

## 3 Including Python docstrings in the documentation

- Enable the `autodoc` and `napoleon` extensions
- Add a file `api_referenc/api.rst` and reference it from `api_reference.rst`
- Include the documentation from the `weather_app.api` module using the `automodule` directive.

## 4 Automating unit tests

- Add a GitHub workflow to automate running unit tests to the repository
- Push the workflow to your personal fork of the repository
- Check the Actions tab on the GitHub page to make sure that workflows are running
- Add a badge to your README.md indicating the status of the workflow.