

# Hacking WeasyPrint

Assuming you already have the [dependencies](#), install the [development version](#) of WeasyPrint:

```
git clone git://github.com/Kozea/WeasyPrint.git
cd WeasyPrint
virtualenv --system-site-packages env
. env/bin/activate
pip install pytest Sphinx -e .
weasyprint --help
```

This will install WeasyPrint in “editable” mode (which means that you don’t need to re-install it every time you make a change in the source code) as well as [py.test](#) and [Sphinx](#).

## Documentation changes

The documentation lives in the `docs` directory, but API section references docstrings in the source code. Run `python setup.py build_sphinx` to rebuild the documentation and get the output in `docs/_build/html`. The website version is updated automatically when we push to master on GitHub.

## Code changes

Use the `py.test` command from the `WeasyPrint` directory to run the test suite.

Please report any bugs/feature requests and submit patches/pull requests [on Github](#).

## Dive into the source

The rest of this document is a high-level overview of WeasyPrint’s source code. For more details, see the various docstrings or even the code itself. When in doubt, feel free to [ask!](#)

Much like [in web browsers](#), the rendering of a document in WeasyPrint goes like this:

1. The HTML document is fetched and parsed into a tree of elements (like DOM)
2. CSS stylesheets (either found in the HTML or supplied by the user) are fetched and parsed
3. The stylesheets are applied to the DOM tree
4. The DOM tree with styles is transformed into a *formatting structure* made of rectangular boxes.
5. The formatting structure is rendered into a sequence of boxes.