# Contributing to Ignite

This project is a community effort, and everyone is welcome to contribute !

If you are interested in contributing to Ignite, there are many ways to help out. Your contributions may fall

into the following categories:

1. It helps us very much if you could

- Report issues you’re facing

- Give a :+1: on issues that others reported and that are relevant to you

- Spread a word about the project or simply :star: to say "I use it"

2. Answering queries on the issue tracker, investigating bugs and reviewing other developers’ pull requests are

very valuable contributions that decrease the burden on the project maintainers.

3. You would like to improve the documentation. This is no less important than improving the library itself!

If you find a typo in the documentation, do not hesitate to submit a GitHub pull request.

4. You would like propose a new feature and implement it

- Post about your intended feature, and we shall discuss the design and

implementation. Once we agree that the plan looks good, go ahead and implement it.

5. You would like implement a feature or bug-fix for an outstanding issue

- Look at the issues labelled as

["help wanted"](https://github.com/pytorch/ignite/issues?q=is%3Aissue+is%3Aopen+label%3A%22help+wanted%22)

- Pick an issue and comment on the task that you want to work on this feature.

- If you need more context on a particular issue, please ask and we shall provide.

If you modify the code, you will most probably also need to code some tests to ensure the correct behaviour. We are using

`pytest` to write our tests:

- naming convention for files `test\_\*.py`, e.g. `test\_precision.py`

- naming of testing functions `def test\_\*`, e.g. `def test\_precision\_on\_random\_data()`

New code should be compatible with Python 3.X versions. Once you finish implementing a feature or bugfix and tests,

please run lint checking and tests:

#### Formatting Code without pre-commit

If you choose not to use pre-commit, you can take advantage of IDE extensions configured to black format or invoke black manually to format files and commit them.

```bash

pip install black

black .

# This should autoformat the files

git add .

git commit -m "....."

```

#### with pre-commit

To ensure the codebase complies with a style guide, we use [black](https://black.readthedocs.io/en/stable/) and [flake8](https://flake8.pycqa.org/en/latest/) to format and check codebase for compliance with PEP8.

To automate the process, we have configured the repo with [pre-commit hooks](https://pre-commit.com/) to use black to autoformat the staged files to ensure every commit complies with a style guide. This requires some setup, which is described below:

1. Install pre-commit in your python environment.

2. Run pre-commit install that configures a virtual environment to invoke black and flake8 on commits.

```bash

pip install pre-commit

pre-commit install

```

3. When files are committed:

- If the stages files are not compliant with black, black will autoformat the staged files. If this were to happen, files should be staged and committed again. See example code below.

- If the staged files are not compliant with flake8, errors will be raised. These errors should be fixed and the files should be committed again. See example code below.

```bash

git add .

git commit -m "Added awesome feature"

# DONT'T WORRY IF ERRORS ARE RAISED.

# YOUR CODE IS NOT COMPLIANT WITH FLAKE8 or BLACK

# Fix any flake8 errors by following their suggestions

# black will automatically format the files so they might look different, but you'll need to stage the files again for committing

# After fixing any flake8 errors

git add .

git commit -m "Added feature"

```

#### Run lint checking

```bash

flake8 ignite/ tests/ examples/

```

#### Run tests:

To run a specific test, for example `test\_terminate` from `test\_engine.py`

```bash

pytest tests/ignite/engine/test\_terminate -vvv -k test\_terminate

```

To run all tests with coverage (assuming installed `pytest-cov`):

```bash

CI\_PYTHON\_VERSION=<your python version, e.g 3.7> sh tests/run\_cpu\_tests.sh

# for example

# CI\_PYTHON\_VERSION=3.7 sh tests/run\_cpu\_tests.sh

```

#### Send a PR

If everything is OK, please send a Pull Request to https://github.com/pytorch/ignite

If you are not familiar with creating a Pull Request, here are some guides:

- http://stackoverflow.com/questions/14680711/how-to-do-a-github-pull-request

- https://help.github.com/articles/creating-a-pull-request/

## Writing documentation

Ignite uses [Google style](http://sphinxcontrib-napoleon.readthedocs.io/en/latest/example\_google.html)

for formatting docstrings. Length of line inside docstrings block must be limited to 120 characters.