CROCO python tools: pycroco + pycroco_gui

Why pycroco?

- Why python
  - Huge community with numerous high-level libraries
  - Ability to work at all scales: laptop, HPC with MPI or GPU
- Provide pre- and post-processing tools
  - Pre-processing: grid, initial conditions, boundary conditions, forcing...
  - Post-processing...
- Federate development efforts in python
  - A single place for all developments
  - Try to integrate most relevant/efficient/useful existing and new solutions
- Follow standards
  - Syntax, project architecture, documentation...
  - Compatible with operational and research contexts

```
conda install pycroco
pip install pycroco
```

Entry points

- The executable scripts
  - Easily accessible
  - Executables with commandline options
  - Designed for specific tasks

```
$ pycroco-download-samples –help
usage: pycroco-download-samples [-h] [--url URL] [-o OUTDIR]
```

- The library
  - Use pycroco like in tutorials
  - Do things like you want
  - Combine pycroco with your solutions

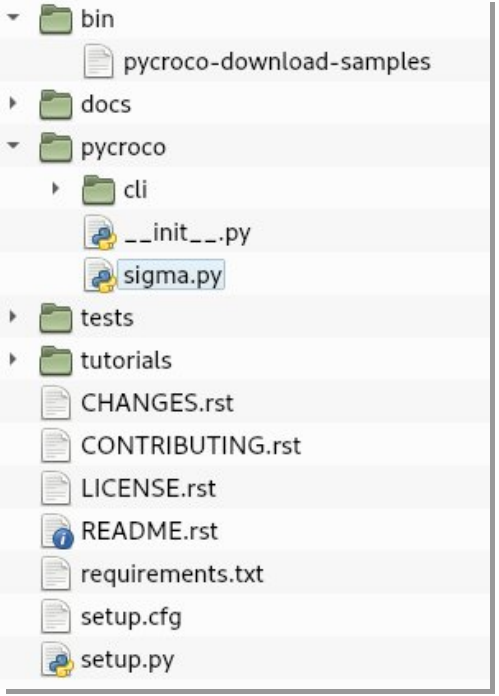
```
>>> from pycroco.sigma import scoord2z
>>> depth = scoord2z(...)
```

Development workflow

- The sandbox
  - https://gitlab.inria.fr/croco-ocean/pycroco_sandbox
  - Organized by functionality
  - Test and share candidate solutions
- pycroco
  - https://gitlab.inria.fr/croco-ocean/pycroco
  - Integrate solutions
- Continuous integration
  - Unit tests
  - Documentation with tutorials

Following standards

- Project structure
  - library + tests + docs + tutorials
  - setup
- Code and doc
  - PEP8
  - Doc structure
- Quality
  - Gitlab workflow like croco
  - Continuous integration
- Easy installation?
  - Python package index
  - Conda forge



Two minutes within the documentation

Automatically generated

https://croco-ocean.gitlabpages.inria.fr/pycroco/index.html