Fun with flags

Command-line Arguments

Flags: Command-line interface (cli)

```
positional arguments:
  file                  The filename to apply filter to

options:
  -h, --help            show this help message and exit
  -o OUT, --out OUT     The output filename
  -g, --gray            Select gray filter
  -se, --sepia          Select sepia filter
  -sc SCALE, --scale SCALE
                        Scale factor to resize image
  -i {python,numba,numpy,cython}, --implementation {python,numba,numpy,cython}
                        The implementation
```

Command Line Arguments

- The command line interface (also known as **CLI**) is a means to interact with a command line script.

- Python comes with several different libraries that allow you to write a command line interface for your scripts, but the standard way for creating a CLI in Python is currently the Python `argparse` library.

- Flags are usually represented with a dash. E.g. `-v`

Command Line Arguments: Using `argparse` module

- The `argparse` module makes it easy to write user friendly command line interfaces.
- The program defines what arguments it requires, and `argparse` will figure out how to parse those out of `sys.argv`.
- The `argparse` module also automatically generates help and usage messages and issues errors when users give the program invalid arguments.

Command Line Arguments: Using `argparse` module - example

```
parser.add_argument(
    "-i",
    "--implementation",
    help="The implementation of the program",
    choices=["python", "cython"],
    default="python",
)
```

Command Line Arguments: Using `argparse` module

A good guide can be found here ☺

https://realpython.com/command-line-interfaces-python-argparse/

Ps. Don’t forget to tell the toml file to use the cli file (instapy.cli:main)

Run Example

- Running the following from the terminal:

```
instapy rain.jpg -g -i numba -o rain_gs.jpg
```

- This should result in a grayscale image of rain.jpg using the numba implementation with the filename “rain_gs.jpg”