`PARSEARG`: TURNING `ARGPARSE` ON ITS HEAD THE DECLARATIVE WAY

THOMAS P. HARTE

CONTENTS

ABSTRACT. Python's module for writing command-line interfaces ("CLI") is `argparse`. There are many other packages for creating CLIs, but `argparse` is Python's standard. Creating a CLI with `argparse`—and especially a CLI that splits its functionality into sub-commands, such as `git init`, `git commit`, and so on—is an exercise in imperative programming: The structure of the CLI is specified command by command until the data that specify the behavior of each command-line argument are added. The imperative nature of this process obfuscates the CLI design. The diametric opposite is `parsearg`, which starts with a data structure containing the data that specify the behavior of each command-line argument, from which `parsearg` generates a parser using `argparse`. The declarative nature of the `parsearg` approach places the CLI design front and center with a data structure (a simple `dict` with an appropriate key schema for specifying a flattened data tree).

Date: 2020-11-18 Build: `(00736c900c3262b6819720c430e6b928547aa60a)`.