DYALOG

2021

Packaging Dyalog Tools

Brian Becker

Building a Treefort

- Buy a hammer, or better yet, borrow one
- Buy some standard dimensional lumber
- Buy some standard fasteners (nails, etc)
- Buy or borrow a ladder
- Build the treefort

Building a Treefort the (old) APL way

- Dig in the ground for iron ore
- Smelt the ore to fashion your own hammer, nails and saw
- Use the saw to cut down a tree and fashion your own lumber
- Build your own ladder
- Build the treefort

Building a Treefort the (old) APL way

How many of us have written (at least) one set of utilities to

- manipulate character data (delete_blanks, center, etc)?
- work with dates (daysdiff, day_of_week, etc)?
- manage data on file (openfile, exists, etc)?

Building a Treefort the (old) APL way

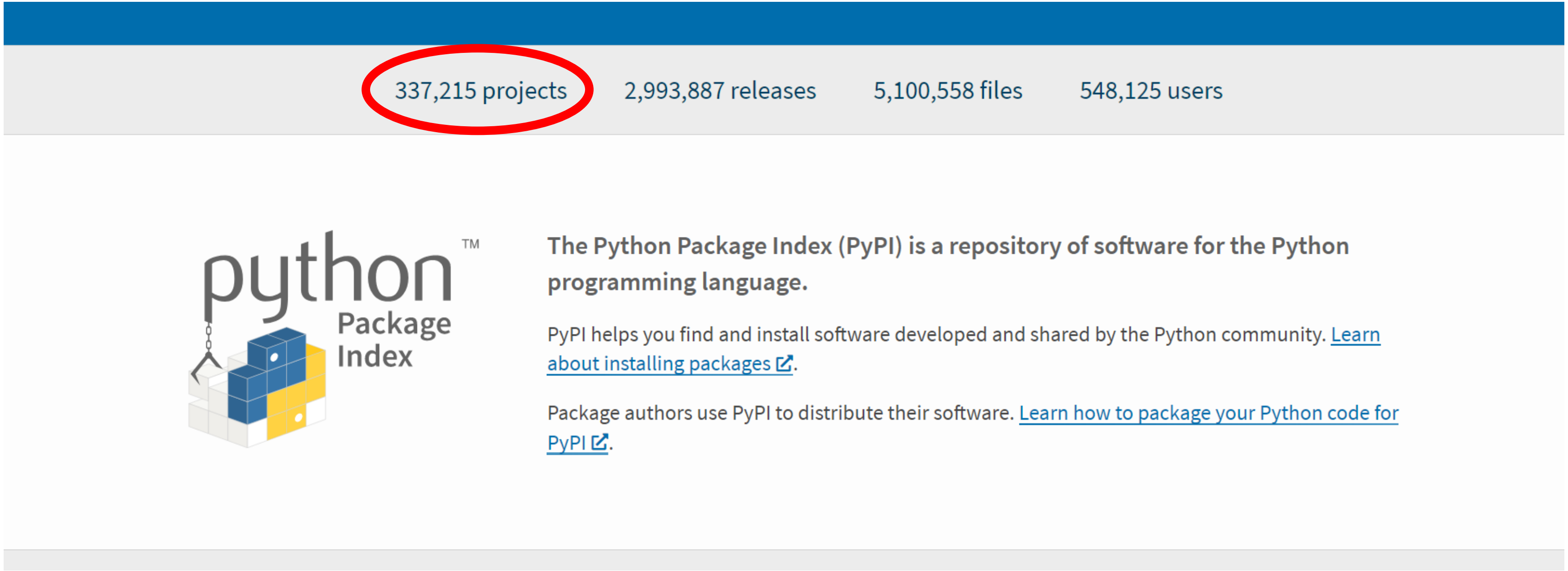
Building a Treefort the (old) APL way

WHY?

Agenda

- Packages
- Package Management
- Dyalog's Commitment

Packages



337,215 projects   2,993,887 releases   5,100,558 files   548,125 users

The Python Package Index (PyPI) is a repository of software for the Python programming language.

PyPI helps you find and install software developed and shared by the Python community. Learn about installing packages.

Package authors use PyPI to distribute their software. Learn how to package your Python code for PyPI.

Packages

- "Package" – a tool
  - In APL, it could be a namespace, class, function, or a collection of them, plus non-APL assets (HTML files, shared libraries, etc)
  - Candidates include Jarvis, HttpCommand, subsets of dfns workspace, Dyalog Cryptographic Library, and others in the pipeline

Package Management

- Introducing Tatin
  - Designed over a 2-year period by Gilgamesh Athoraya, Paul Mansour, Morten Kromberg, and Kai Jaeger.
  - Prototyped by Gil, implemented by Kai
  - Davin Church has been an active beta tester
- www.tatin.dev
- Supports package consumers and package developers
  - Today we'll talk (mostly) about package consumption

Tarte Tatin
A delicious way to package apples

Demo

- Create an application that uses 3 packages
  - Jarvis – web service server from Dyalog
  - HttpCommand – information retrieval from Dyalog
  - FilesAndDirs – package from APLTeam
    - which depends on package APLTreeUtils2 and package OS
      - which also depends on package APLTreeUtils2

Recap

- `]LoadPackage` – load package into workspace
- `]InstallPackage` – save package to file
- `]LoadDependencies` – load saved packages
- Pre-loaded packages or Dynamically loaded?
- Add your code
- Deploy however best makes sense

Semantic Versioning

- Major.minor.patch – `HttpCommand 4.0.0`
  - Patch – bumped when there is no change in behavior
  - Minor – bumped when functionality is added, but everything else remains the same
  - Major – bumped when there is a change in the API, functionality is removed, renamed, or parameters change
    - For upgrade purposes
      Foo-1.0.0 is a different package than Foo-2.0.0

Tatin Is and Isn't...

- It's a package manager
  - For consumers - incorporate packages in your applications
  - For producers – provides a means of publishing your packages
- It's a package repository (registry)
- It's not a workflow tool
  As a consumer, it will not force you into using a particular development workflow

Benefits of Using Tatin

- Reproducibility
  - You know the version of each package
- Dependency Management
  - Tracked and managed automatically
  - Resolves dependency conflicts
    - Foo depends on Moo 1.1
      Goo depends on Moo 1.2
- Packages can have non-APL resources
  - Shared libraries (dlls), HTML, data files, etc.

Dyalog's Commitment

- Supporting the community
  - Publishing our packages on www.tatin.dev
  - Encouraging and helping others to do the same
- Completeness – it's more than just the code...
  - Documentation
  - Training materials
  - Test suites
- Version discipline
  - Use semantic versioning
  - Creating more timely releases

Questions?