PLY (Python Lex-Yacc)

Willkommen auf der PLY-Homepage. PLY ist eine Implementierung von lex und yacc Parsing-Tools für Python. Wenn Sie nicht die geringste Ahnung haben, was Das heißt, Sie sind wahrscheinlich am falschen Ort. Ansonsten behalten lesen.

Kurz gesagt, PLY ist nichts weiter als ein einfacher Lex/Yacc Implementierung. Hier ist eine Liste der wesentlichen Merkmale:

- Es ist vollständig in Python implementiert.
- Es verwendet LR-Parsing, das einigermaßen effizient und gut ist geeignet für größere Grammatiken.
- PLY bietet die meisten Standard-lex/yacc-Funktionen einschließlich Unterstützung für leere Produktionen, Vorrangregeln, Fehlerbehebung und Unterstützung für mehrdeutige Grammatiken.
- PLY ist einfach zu bedienen und *sehr* umfangreich Fehlerüberprüfung.
- PLY versucht nicht, mehr oder weniger zu tun, als das grundlegende lex/yacc bereitzustellen Funktionalität. Mit anderen Worten, es ist kein großes Parsing-Framework oder eine Komponente eines größeren Systems.

Die ursprüngliche Version von PLY wurde 2001 für entwickelt Verwendung in einem Kurs zur Einführung in Compiler, in dem Studenten verwendet haben es, einen Compiler für eine einfache Pascal-ähnliche Sprache zu bauen. Durch seine Verwendung in einer Unterrichtsumgebung, viel Arbeit wurde in die Bereitstellung gesteckt umfangreiche Fehlerprüfung. Zudem war diese Erfahrung gewöhnungsbedürftig allgemeine Usability-Probleme lösen. Seitdem eine Vielzahl von Inkrementelle Verbesserungen wurden am System vorgenommen.

Aktueller Status

Die Arbeit an PLY-4.0 hat begonnen und stellt eine umfassende Überarbeitung dar Modernisierung für Python 3.6+. Sie können diese Version nur von der erhalten GitHub-Repository unter https://github.com/dabeaz/ply . Wenn Sie nach einer Legacy-Version suchen, sollten Sie weiterhin Version 3.11 verwenden.

Verknüpfungen

- Laden Sie PLY-3.11
- Ältere Versionen: [1.0 | 1.1 | 1.2 | 1.3.1 | 1.4 | 1,5 | 1.6 | 1.7 | 1.8 | 2.0 | 2.1 | 2.2 | 2.3 | 2.5 | 3.0 | 3.1 | 3.2 | 3.3 | 3.4 | 3.6 | 3.7 | 3.8 | 3.9 | 3.10]
- Sehen Sie sich die aktuelle Dokumentation an .
- Schauen Sie sich ein einfaches Beispiel .
- Ein Artikel über PLY im Dr. Dobb's Journal. Von Shannon Behrens.
- Folien von Daves PyCon'07 Talk über PLY.
- Wie man einen Parser mit PLY von Andrew Dalke konstruiert.
- PLY im Vergleich mit Pyparsing und ANTLR von Andrew Dalke.
- Matthieu Amiguet hat ein Tutorial zum Erstellen eines Compilers mit PLY geschrieben (auf Französisch).

Einige Projekte mit PLY

- Pyrata. Ein auf Python-Regeln basierender Feature-Struktur-Analysator.
- pysmi . Ein SNMP-MIB-Parser.
- pyfranka . Ein Python-Modul und Tools zum Arbeiten mit IDL-Modellen (Interface Definition Language) von Franca.
- Pycparser . AC-Parser in Python implementiert.
- cppheaderparser . Ein C++-Header-Parser.
- ZXBasic . Ein Cross-Compiler übersetzt BASIC in Z80-Assembler.
- CGCC . AC GCode-Compiler für industrielle CNC-Maschinen
- OE-lite. Ein eingebettetes Linux-Build-System.
- [Fügen Sie Ihr Projekt hier hinzu, indem Sie eine E-Mail an "dave" auf "dabeaz.com" senden]

Die Unterstützung

- Die GitHub-Seite (Problemverfolgung, Repository-Zugriff usw.) ist auf GitHub verfügbar.

Wichtiger Hinweis

PLY wird nicht mehr als pip-installierbares Paket gepflegt. Obwohl es sind keine neuen Features geplant, es wird weiterhin gepflegt und modernisiert. Wenn Sie die neueste Version verwenden möchten, müssen Sie dies überprüfen es von der PLY GitHub Seite. Wenn Sie nach einem Parser suchen Generator mit einem moderneren Blick auf das SLY-Projekt .