TinyPEG: A Parsing Expression Grammar Library

Welcome to the documentation for TinyPEG, a lightweight and easy-to-use Parsing Expression Grammar (PEG) library implemented in Python.

What is TinyPEG?

TinyPEG is a parser library that allows you to define grammars using Parsing Expression Grammar notation and use them to parse and analyze text. It's designed to be:

- **Simple**: Easy to understand and use
- **Flexible**: Adaptable to various parsing needs
- **Educational**: A great tool for learning about parsers and language design

Getting Started

- Preface: Introduction to parsing and grammar concepts
- Contents: Full table of contents
- Chapter 1: Understanding PEG Parsers
- Chapter 2: TinyPEG Library Overview
- Chapter 3: Building Your First Parser
- Chapter 4: Example Parsers
- Chapter 5: Creating a Tiny Programming Language

Quick Example

```python from src.peg.fixed_parsers import PEGParser, Rule, Regex from src.peg.fixed_core import GrammarNode

# Create a simple number parser

class SimpleParser(PEGParser): def **init**(self): super().**init**()
```

```
    # Define a simple grammar for numbers
    self.grammar = GrammarNode(
        name="Simple",
        rules=[
            Rule("Number", Regex("[0-9]+"))
        ]
    )
```

Use the parser

parser = SimpleParser() result = parser.parse("42") print(f"Parsed result: {result}") # Output: Parsed result: 42 ```

For a complete working example, see the Simple Fixed Test in the examples directory.

License

This project is open source and available under the MIT License.

Copyright