

Milestone 1

Parser

The goal of this phase is to build a structured text recognition framework, a little bit like ANTLR, but with extensible grammars.

Core parsing technology (milestone 1)

- Grammar syntax - [done](#)
- Extensible parser - [done](#)
- Error messages - [done](#)
- Error recovering - [done](#)
- Automatic Parse Tree (detailed AST) generation - [done](#)
- Quasiquote (constructing and matching a code by patterns) - [done](#)
- Serializing AST back to document - [done](#)
 - PrettyPrint based on marker tags in grammar - [done](#)
 - HTML PrettyPrint - [done](#)
 - ToString based on PrettyPrint - [done](#)
- Testing framework - [done](#)
- Documentation, samples, SDK - [done](#)


IDE support

- Syntax highlighting (coloring) - [done](#)
- Outlining (aka folding) - [done](#)
- Parse errors highlighting and error list - [done](#)
- Matching constructs (braces, etc) - [done](#)

Grammar verification

- Nitra - [done](#) (bootstrapping)
- C# - [done](#)
- JSON - [done](#)
- URI - [done](#)
- Markdown like syntax - [done](#)

Completing this phase means we can do the following:

- Express language grammar in a concise form, without heavy-lifting on complexities of EBNF and such.
 -  If we fail to make grammar design clear and simple, all the rest will fail miserably.
- Generate fast parser and AST builder