

Research questions

- What is Spoofax?
- What is a REPL?
 - How is the execution context/environment handled?
 - Comparison/feature matrix of some popular REPLs
 - Python and the IPython project
 - R
 - Lisp (Scheme, Racket)
 - Haskell
 - AutoCAD
- What is Literate Programming?
 - What are the key differences between the following literate programming solutions?
 - IPython/Jupyter notebooks
More information: [Literate programming, RStudio, and IPython Notebook](#).
 - Swift playgrounds
More information: [Swift Playgrounds](#).
 - Org-mode in Emacs (with Babel)
Support for *tangling* source code within comments. More information: [The Org Manual](#)
 - How does Literate Programming relate to REPLs?
Think about e.g. reevaluation of expressions (cascading results to later/earlier results).
- How and where does a REPL fit within Spoofax?
 - How to specify language specific features (features that differ accross languages)?
 - How are language-specific REPL commands handled?
 - How to dynamically add definitions to earlier contexts?
There needs to be some way to distinguish language constructs which are definitions, and language constructs which evaluate to a value. See e.g. paplj, one would like to define classes intermixed with evaluating expressions.
 - How can a partial program (without entry point) be executed?
 - How can we integrate IDE specific features with generic REPL features?
 - What is the interaction with the editor views?
For example jumping to definitions when clicking on types or function names.
- How are plugins developed for Eclipse?
 - How are Spoofax's Eclipse plugins implemented?

Structure of the report

- Introduction
- Problem definition
 - What is Spoofax?
 - What is a REPL?
 - What is Literate Programming?

- Problem analysis
 - How and where does a REPL fit within Spoofax?
 - How are plugins developed for Eclipse?
- Requirement analysis
 - Minimal viable product
- Realisation of the product
 - Methodologies used during the project
 - Development tools
- Conclusion