



Clang-REPL: Interactive Analysis

Mentors: Vassil Vasilev, David Lange



Introduction

A User documentation is designed to assist end users to use a product or service. Good user documentation is very important because it provides an avenue for users to learn how to use a software, its features, tips, tricks and also resolve common problems encountered when using the software. It also reduces support cost and is part of the corporate identity of the product: a good user documentation is a sign of healthiness of the product, the developer team. Without a good user documentation, a user may not know how to do the above things effectively and efficiently. User Documentations can play a pivotal role in ensuring a product's success because great communication is and will always be at the heart of any business or product and a great documentation just takes that communication and puts it in a manageable framework that everyone can access for success.



Scope

In this project I will audit the existing documentation for the Clang-REPL (interactive C++) and identify the gaps in the information or presentation from the point-of-view of new, physics-oriented users and incorporate feedback from developers in the project and the community.



The scope of the interactive analysis component continues with:

- Demonstrate interactive use-cases of Clang REPL in the LLVM documentation by developing basic documentation and tutorials apt for Clang REPL
- Develop advanced Clang REPL tutorials and integrate Clang REPL into Xeus and contribute to a community blog post on them
- Review and enhance the developers documentation, examples and tutorials



Success Metrics

The Interactive Analysis Development success metrics include:

1. Publishing at least 2 blog posts on <https://blog.llvm.org>
2. Publishing at least 3 LLVM help document in <https://clang.llvm.org/docs/>
3. Publish at least two tutorials about more advanced Clang REPL features at <https://compiler-research.org/tutorials/>
4. Reviewing and improving developer's documentation



5. Restructure the documentation.
6. Update the documentation to fit the modern versions of Clang REPL.
7. Rewrite the user documentation to make it easier to understand.
8. We can make it community driven so users can report any issues of the documentation and also be able to make changes that will be reviewed and merged by the Clang REPL team.



Any Questions?



Thank you!