



**Workshop 2020**  
**September 24, 2020, online**

**Thomas van Binsbergen (CWI)**  
ALE Associate team

**Title:** « A principled approach to REPL interpreters »

**Abstract:**

Read-eval-print-loops (REPLs) allow programmers to test out snippets of code, explore APIs, or even incrementally construct code, and get immediate feedback on their actions.

However, even though many languages provide a REPL, the relation between the language as is and what is accepted at the REPL prompt is not always well-defined.

Furthermore, implementing a REPL for new languages, such as DSLs, may incur significant language engineering cost.

In this paper we survey the domain of REPLs and investigate the (formal) principles underlying REPLs. We identify and define the class of sequential languages, which admit a sound REPL implementation based on a definitional interpreter, and present design guidelines for extending existing language implementations to support REPL-style interfaces (including computational notebooks). The obtained REPLs can then be generically turned into an exploring interpreter, to allow exploration of the user's interaction.

The approach is illustrated using three case studies, based on MiniJava, QL (a DSL for questionnaires), and eFLINT (a DSL for normative rules). We expect sequential languages, and the consequent design principles, to be stepping stones towards a better understanding of the essence of REPLs.