

1 Language Primer

citations for this history are fragmented across the internet

Factor is a rather young language created by Slava Pestov in September of 2003. Its first incarnation targeted the Java Virtual Machine (JVM) as an embedded scripting language for a game. As such, its feature set was minimal. Factor has since evolved into a general-purpose programming language, gaining new features and redesigning old ones as necessary for larger programs. Today's implementation sports an extensive standard library and has moved away from the JVM in favor of native code generation. In this section, we cover the basic syntax and semantics of Factor for those unfamiliar with the language. This should be just enough to understand the later material in this thesis. More thorough documentation can be found via Factor's website, <http://factorcode.org>.

1.1 Combinators

Quotations, introduced in ??, form the basis of both control flow and data flow in Factor. Not only are they the equivalent of anonymous functions, the stack model makes them syntactically lightweight enough to serve as blocks of code, akin to the use of curly braces in Java or C. The higher-order words that make use of quotations on the stack are called *combinators*. It's simple to express familiar conditional logic and loops using combinators, as we'll show in Section 1.1.1. In the presence of explicit data flow via stack operations, even more patterns arise that can be abstracted away. Section 1.1.2 explores how we can use combinators to express otherwise convoluted stack-shuffling logic more succinctly.

1.1.1 Control Flow

1.1.2 Data Flow