The Rabbit Programming Language

Computer programming can be a very labor-intensive endeavor. It often takes large amounts of work and huge numbers of lines of code to accomplish even the simplest tasks. A new field of programming languages—functional programming languages—hopes to solve this problem by providing programmers with powerful tools to perform complex tasks in simple ways, using less code. These tools are unique to functional programming and are found to be incredibly powerful for writing simpler, shorter, and more concise code that is easier to debug, think about, and optimize. Unfortunately, they are currently too specialized, which makes them incredibly difficult to learn. The field of functional programming languages lacks a scripting language, an easy-to-use, broadly applicable language built for the average, everyday programmer. Instead, existing functional programming languages focus on mathematical purity and execution speed over general applicability and ease-of-use. A new programming language, Rabbit, written by the author, hopes to solve these problems. Rabbit is a functional scripting language, built to combine the power of functional programming with the ease-of-use and generality of scripting languages. Rabbit does this on top of the existing scripting, albeit non-functional, framework of the Python programming language, which has already proven itself to be easy to use, thus allowing Rabbit to also be easy to extend. It is thought that Rabbit's powerful functional programming features, combined with its easy extensibility, will make Rabbit useful for a variety of purposes, including the creation of Domain-Specific Languages, specialized languages built for a very specific purpose, to solve problems in robotics, engineering, mathematics, and the applied sciences.