

NEA coverpage

Lucas standen

7949

ABSTRACT

1. Analysis

1.1. Overview

Zippy LANG

A next generation language, for general use. Designed for keeping code simple, neat and readable.

Zippy will be a modern functional language. It will be interpreted like python/perl and lisp. Its syntax should be similar to lisp.

Some interesting features that Zippy will have is treating everything in functions, scopes that allow for manual but safe memory management, and interoperability with C librarys

1.2. Examples of similar projects

Since Zippy is a high level programming language it should be compared to other languages.

Since it is interpreted, i should compare it to other such languages.

Zippy is by far not the first language, and I'm only one person, so I can't expect to beat others in everything. However a few small goals will give the project a good scope.

1.2.1. Python

Python is a high level OOP language that was designed in 1991. It was made to make programming easy while still being able to use some of C's functions.

Although it has become standard for many usecases, it is slow and inefficient, and very bloated.

1.2.2. Lisp

Lisp is the second ever programming language, developed at MIT, it is the first functional language, creating many common features like higher order functions, recursion, and garbage collection. It is generally not used anymore as it feels old compared to other functional languages, like ocaml or haskell.

1.2.3. Perl

Perl is scripting lang designed for use in linux, when bash is too slow, or not suited for the job.

Perl is often described as the glue of the universe (see xkcd <https://3d.xkcd.com/224/>). Its syntax is quite strange however and it is slow. Making it poorly suited for general use.

1.3. Questionares

1.3.1. Questions 1 for Amy C

1.3.1.1. What do you find the most important in a language? (eg: speed, readability)

Speed, readability, debugging ease and disk space efficiency.

1.3.1.2. What tools are important for a language to have? (eg: pkg-manager, ide integration)

IDE integration (things like tab complete and debugging tools), a package manager, and the ability to interact with the user through the command line easily.

1.3.1.3. What features do you like from other languages (eg: advanced memory management, haskells terse syntax)

The ability to pass the memory reference of an object or function and a collection of built-in or standard functions like "print", "split", or "sort"

1.3.1.4. What do you want to program in this language (eg: websites, low level systems)

Lightweight command line tools and web backends.

1.3.1.5. Do you intend to use graphics in the programs you write?

No

1.3.1.6. Would you prefer a language that focuses on ease of use, or power of the code?

I like a good balance between the two.

1.3.1.7. What were your last 3 projects? (could they have been written in zippy?)

A website, a small command-line tool and a midi keyboard (program runs on a Raspberry Pi Pico)

1.3.1.8. How many languages would you use on a single project? (could zippy be used in your codebase?)

I try to use as little languages in a project as possible, so likely not in an existing project.

1.3.1.9. Do you care for low level control, or would you prefer high level abstractions?

I think low-level control is very important, but high-level abstractions are convenient, so a good balance between the two is best.

1.3.1.10. Would you be happy to develop libraries for things that aren't already implemented (eg: an sql lib)

Potentially if it is simple enough to implement new things.

2. Design

3. Technical Solution

4. Testing

5. Evaluation