psilo

a parallel, safe, inferencing list operation language for writing interesting programs. View it on GitHub.

© 2014 Gatlin Johnson gatlin@niltag.net

What is psilo?

psilo will be a programming language created with the philosophy that *all* programs essentially define (restricted) languages.

Technical Features (planned):

- No run-time garbage collection necessary owing to uniqueness types
- Static typing for compile-time verification and optimization
- Malleable syntax with macros
- Dead-simple parallelism with special array types
- Monadic continuations and iteratee composition made dead simple
- Orthogonal core syntax and semantics for your performance and my sanity

Philosophy:

- All programming is manipulating languages.
- Types define grammars; functions define parsers.
- The earlier a question may be answered, the better.
- If the computer can do it, it should.

Status

Psilo is still being designed. I have implemented a really simple evaluator as well as this website so that when the time comes, I won't have tedious tooling or process issues getting in the way of implementation.

Additionally, the simple interpreter might be of educational value.

How to build

You need the Glasgow Haskell Compiler and a number of libraries; I suggest starting off with the Haskell platform.

Clone the repository:

```
git clone https://github.com/gatlin/psilo
```

Set up a cabal sandbox:

```
cabal sandbox init
cabal configure
cabal install --only-dependencies
```

Then make with:

```
make
```

And return to the Edenic, pre-build post-checkout status of the code with

```
make clean
```

Questions / comments / hate mail

Use the Issues feature of GitHub or email me: gatlin@niltag.net.