Welcome

Wren is a small, fast, class-based concurrent scripting language

Think Smalltalk in a Lua-sized package with a dash of Erlang and wrapped up in a familiar, modern syntax.

```
System.print("Hello, world!")

class Wren {
  flyTo(city) {
    System.print("Flying to %(city)")
  }
}

var adjectives = Fiber.new {
  ["small", "clean", "fast"].each {|word| Fiber.yield(word) }
}

while (!adjectives.isDone) System.print(adjectives.call())
```

- **Wren is small.** The VM implementation is under 4,000 semicolons. You can skim the whole thing in an afternoon. It's *small*, but not *dense*. It is readable and lovingly-commented.
- **Wren is fast.** A fast single-pass compiler to tight bytecode, and a compact object representation help Wren compete with other dynamic languages.
- **Wren is class-based.** There are lots of scripting languages out there, but many have unusual or non-existent object models. Wren places classes front and center.
- **Wren is concurrent.** Lightweight fibers are core to the execution model and let you organize your program into an army of communicating coroutines.
- **Wren is a scripting language.** Wren is intended for embedding in applications. It has no dependencies, a small standard library, and an easy-to-use C API. It compiles cleanly as C99, C++98 or anything later.

If you like the sound of this, let's get started. You can even try it in your browser! Excited? Well, come on and get involved!