

FRY Language Reference

Tom DeVoe
tcd2123@columbia.edu

October 8, 2014

Contents

| | | |
|----------|-------------------------------|----------|
| 1 | Introduction | 1 |
| 2 | Lexical Conventions | 2 |
| 2.1 | Comments | 2 |
| 2.2 | Identifiers | 2 |
| 2.3 | Keywords | 2 |
| 2.4 | Constants | 2 |
| 3 | Types | 2 |
| 3.1 | Primitive Types | 2 |
| 3.2 | Compound Types | 2 |
| 4 | Meaning of Identifiers | 3 |
| 5 | Conversions | 3 |
| 6 | Expressions | 3 |
| 7 | Declarations | 3 |
| 8 | Statements | 3 |
| 9 | Scope | 3 |

1 Introduction

This document serves as a reference manual for the **FRY** Programming Language.

2 Lexical Conventions

2.1 Comments

Single line comments are denoted by the character, `#`. Multi-line comments are opened with `#!/` and closed with `/#`

2.2 Identifiers

An identifier is a string of letters, digits, and underscores. A valid identifier begins with an letter or an underscore. Identifiers are case-sensitive and can be at most 31 characters long.

2.3 Keywords

The following identifiers are reserved and cannot be used otherwise:

```
int      str      float  bool
Record   Array    Table   if
else     elif     in      Sort
trans
```

2.4 Constants

There is a constant corresponding to each Primitive data type mentioned in 3.1.

3 Types

3.1 Primitive Types

- `int` - 64-bit signed integer value
- `str` - An ASCII text value
- `float` - A double precision floating-point number
- `bool` - A boolean value. Can be either `true` or `false`

3.2 Compound Types

- `Array` - an ordered collection elements of the same data type.
- `Record` - a named collection of variables. Records behaves similar to structs in C.

- 4 **Meaning of Identifiers**
- 5 **Conversions**
- 6 **Expressions**
- 7 **Declarations**
- 8 **Statements**
- 9 **Scope**