FRY Language Reference

Tom DeVoe
tcd2123@columbia.edu

October 12, 2014

Contents

1 Introduction

This document serves as a reference manual for the **FRY** Programming Language. **FRY** is a language designed for processing delimited text files.

2 Lexical Conventions

2.1 Comments

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

```
# This is a single line comment

#/ This is a
multi-line comment /#
```

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  Layout
    List   Table  if     else  elif
    in     Sort
```

2.4 Constants

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

- **Integer Constants** - Integer constants are whole base-10 numbers represented by a series of numerical digits (0 - 9).

```
# Integer Constant Examples
int x = 312342
int y = 111111112
int z = 8
```

- **Float Constants** - Float constants are similar to Integer constants in that they are base-10 numbers represented by a series of numerical digits. However, floats can also include a decimal separator.

```
# Float Constant Examples
float f1 = 1.158472
float f2 = 2457.89
float f3 = 19999.999999
```

- **String Constants -** String constants are represented by a series of ASCII characters surrounded by quotation-marks. Certain characters can be escaped inside of Strings with a backslash '´. These characters are:

Character	**Meaning**
`\n`	Newline
`\t`	Tab
`\\`	Backslash
`\"`	Double Quotes

```
# String Constant Examples
str s1 = "This is \t a string\n"
str s2 = "This. is. also—a—\"string!\""
str s3 = "42"
```

- **Boolean Constants -** Boolean constants can either have the case-sensitive value *true* or *false*.

```
# Boolean Constant Examples
bool b1 = true
bool b2 = false
```

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

- `List` - an ordered collection of elements of the same data type. Every column in a *Table* is represented as a List

- `Layout` - a collection of named data types. Layouts behave similar to structs from C. Once a Layout is constructed, that layout may be used as a data type. An instance of a Layout is referred to as a *Record* and every table is made up of records of the Layout which corresponds to that table.

- `Table` - a representation of a relational table. Every column in a table can be treated as a *List* and every row is a record of a certain *Layout*. Tables are the meat and potatoes of **FRY** and will be at the center of most programs.

4 Meaning of Identifiers

5 Conversions

6 Expressions

7 Declarations

8 Statements

9 Scope