The Reference Manual for ∃L

Version 3.14

The ∃L Development Team

March 4, 2016

Contents

1 Language Guide 1
1.1 types . . . 1
1.2 constants . . . 1
1.3 list . . . 1
1.4 tuple . . . 1
1.5 function . . . 1
1.6 control flow . . . 1

2 Usage 2
2.1 declare constants . . . 2
2.2 declare function . . . 2
2.3 pattern match . . . 2
2.4 Input and Output . . . 2
2.5 sample code . . . 2

Chapter 1

Language Guide

§1.1 types

ヨL provides fundamental types, including `Int` for integers, `Double` for floating-point values, `String` for textual data, and `Bool` for Boolean value. ヨL also provides Tuple and List as described in §1.3, §1.4.

§1.2 constants

You cannot use variable in ヨL-world. Alternatively, ヨL provides constant. The value of a constant cannot be changed once it is set in the future. This constraint ensures referential transparency.

§1.3 list

§1.4 tuple

§1.5 function

§1.6 control flow

Chapter 2

Usage

§2.1 declare constants

§2.2 declare function

Listing 2.1 function $fibo(n)$

```
fibo(n) :: Int -> Int
 = 0 [n == 0]
 = 1 [n == 1]
 = fibo(n-1) + fibo(n-2)
```

§2.3 pattern match

§2.4 Input and Output

§2.5 sample code

Here is the sample code, which returns a fibonacci number.

Listing 2.2 sample code

```
fibo(n) :: Int -> Int
 = 0 [n == 0]
 = 1 [n == 1]
 = fibo(n-1) + fibo(n-2)
where
 who :: Int
 =5
 hoge :: Int
 =6

m :: Double
=2
```