Extend

Language Reference Manual

Ishan Kolluri, Kevin Ye, Jared Samet, Nigel Schuster

October 19, 2016

Contents

1. Introduction to Extend

2. Types and Literals

2.1. Primitive Data Types

2.2. Ranges

2.2.1. Range Slicing

2.2.2. The Underscore Symbol

2.3. Integers and Strings

2.3.1. Unary Operations

3. Expressions

3.1. Operators

3.1.1. Multiplication

3.1.2. Addition

3.1.3. Unary

3.2. Booleans

3.3. Variable Declaration

3.4. Variable Assignment

3.5. Conditionals

4. Functions

Functions lie at Extend's core; however, they are not *first class objects*. Since it can be verbose to write certain operations in Extend, the language will feature a comprehensive number of built-in

and standard library function. An important built-in function will be I/O (see section 5).

4.1. Format

Every function in Extend follows the same format, but allows some optional declarations. As in most programming languages the header of the function declares the parameters it accepts and the return type. The simplest function is this:

```
[1,1] foo([1,1] arg) {
  return arg;
}
```

This function simply returns whatever value is passed into it. The leading `[1,1]` marks the return dimensions. `foo` is the function name. In parentheses the function arguments are declared, again with dimensions of the input. The body of the function follows, which in this case is only the return statement.

4.2. Dimension Assignment

4.3. Application across Dimensions

4.4. Dependencies Illustrated

5. I/O

5.1. File

5.2. Input Arguments

5.2.1. How to run a program

5.2.2. Main function

6. Example Program