

## Contents

# **1 Introduction: What, Why, and How?**

## **1.1 What is \English?**

## **1.2 What makes it different?**

## **1.3 What is good about these differences?**

## **1.4 How are these differences possible?**

# **2 How to Use the Language**

## **2.1 Compilation**

## **2.2 The Basic Basics**

### **2.2.1 `main(... Oh, Wait`**

### **2.2.2 Variables and Math**

Standard Number Types

Variable Declaration and Assignment

Standard Arithmetic Operators

### **2.2.3 Lists (and Strings)**

Making a List

Manipulating Lists By Index(es)

Strings Are Lists And More!

## **2.3 Code Blocks**

### **2.3.1 If and Its Friends**

### **2.3.2 Loops**

## **2.4 Operators**

### **2.4.1 Defining Them**

### **2.4.2 Using Them**

## **2.5 Types**

### **2.5.1 Defining Them**

### **2.5.2 Using Them**

### **2.5.3 Type Modifiers**

Defining Them

Using Them

#### **2.5.4 Units**

Defining Them  
Using Them

## **2.6 Errors**

### **2.6.1 Defining Them**

### **2.6.2 Using Them**

## **2.7 Input and Output**

### **2.7.1 Stdin Reading and Stdout Printing**

### **2.7.2 File Reading and Writing**

## **2.8 Implicit Conversion**

### **2.8.1 What?**

### **2.8.2 Why?**

### **2.8.3 Super-implicit?**

## **2.9 High Order Functionality**

### **2.9.1 What?**

### **2.9.2 Why?**

### **2.9.3 Missing Something? Partially Applied Operators**

### **2.9.4 Operator Modifiers**

### **2.9.5 Scope and Contextual Operators**

## **3 The Basis of the Language**

### **3.1 operator**

### **3.2 define**

### **3.3 asm**

### **3.4 run**

### **3.5 bits**

### **3.6 use**

## **4 The Not-So-Complicated Compilation**

### **4.1 Idea**

### **4.2 Optimizability**