Programming in Meruem

Melvic C. Ybanez

September 2015

Contents

Introduction

0.1 About the book

This book is a tutorial for the Meruem programming language, written by the people who developed the current version of Meruem. Our goal is to teach you the introductory concepts of functional programming and (to some extent) metaprogramming using the Meruem language.

0.2 Who should read this book

If you are someone looking for the next popular programming language to master and doesn't feel like learning new and more mathematical ways of solving problems for now, then this tutorial is not for you. There are many options for new languages out there, but Meruem is not (yet) one of them.

That said, if you are someone willing to spend a lot of time mastering not just a new programming language but also new programming paradigms, hoping that you will be able to apply all the knowledge you can gain from this book with any other programming languages you already know, then this book is for you.

0.3 Programming background required

This book is written primarily for imperative and/or object-oriented programmers who want to learn functional programming and metaprogramming, or people who don't know programming at all. If you are already familiar with functional programming and metaprogramming, then most of the contents here will not be new to you, but this book can still serve as a review material.

0.4 How to read this book

Most of the chapters in this book are not self-containing, so we recommend you read them in the proper order starting from chapter 1, especially if you are new to Lisp-like languages like Meruem.

Chapter 1

Starting Out

1.1 What is Meruem?

Meruem is a dynamically-typed, interpreted programming language that supports both *functional programming* and *metaprogramming*, and runs on top of the *Java Virtual Machine*(JVM).

Meruem is also a *Lisp* dialect. That means it has most, if not all, of the characteristics common to all Lisps, like *homoiconicity*, *macros*, and a small, simple and elegant core.

1.2 Why learn Meruem?

Meruem will change the way you think about programs, programming, and problems in general. The things that you will learn from this book will still be applicable to your day-to-day job as a programmer, even if you will be using a different and more mainstream programming language. This is because learning Meruem is not just learning a new programming language, it's learning completely new programming paradigms. Knowing different programming paradigms (imperative, OOP, FP, etc) is always a good thing since it would give you different ways of solving problems. After you've learned Meruem, you'd realize that there's more to programming than just *imperative programming*.

1.3 Overview of Programming Paradigms

Before we continue, let us first make a brief discussion about the different programming paradigms. We are not going to talk about all of them, though. We are just going to talk about the ones most programmers are familiar with (imperative/OOP programming), and the ones this book is going to focus on (functional programming and metaprogramming).

1.3.1 Imperative Programming

In **Imperative Programming**, you give the computer a sequence of statements for it to perform. Each of these statements can cause side effects. **Side effects** are changes (on a

state or something) that occur in some place (like outside of a function being invoked) when a function, command or statement is invoked or executed. For example, the following code will print the string `Hello World` to the screen:

```
1    print "Hello World"
```

That is a Python snippet. It is a side effecting statement, because something is printed when that line of code is run. The state of the console has changed. Another example is the modification of variable values or references:

```
1    x = int(raw_input('Enter a number'))
2    if x < 20:
3      x = 7
```

The above code takes an integer (actually it's an integer parsed from a string) from the keyboard, and store it to the variable `x`. If `x` is less than 20, then set it to 7. Line 3 is a side effecting assignment statement since you are destroying the old value and replacing it with a new one, making the value of `x` different from before. This is called a *desctructive assignment*. Line 1, however, is not a side effecting statement since assigning an initial value to a variable is not the same as changing it. This kind of assignment is known as *initialization*.

You'll learn more about side effects later in the book. (Though you won't learn much about imperative programming in general here.)

1.4 Installing Meruem

To program in Meruem, you need to install Java and download the Meruem interpreter.

1.4.1 The Java Virtual Machine

As I've said above, Meruem runs on the Java platform, which is a JVM (sometimes I just refer to it as "the JVM"). To be more accurate, the current version of Meruem actually gets ran by the Scala programming language, which runs on top of the JVM. What I mean by that is that the interpreter of Meruem is written in Scala.

But, just what is a JVM?

According to Wikipedia, a JVM is "an abstract computing machine that enables a computer to run a Java program". Essentially, without a JVM, we can't run Java programs.

So how do Scala programs run on it if it only understands Java bytecode? Simple, the Scala compiler generates Java bytecode. And since Meruem is written in Scala, then a Meruem code will eventually be converted to Java bytecode.

So we need to install a JVM in order to run our Meruem interpreter. To do that, we install a *Java Runtime Environtment*(JRE). Installing a JRE was what I meant earlier by installing Java. A JRE contains the JVM, libraries, and some other things we shouldn't worry about in this book. There are many instructions on the web on how to install a Java runtime environment on different platforms, such as this one: `https://www.java.com/en/download/help/download_options.xml`

Note: T here is also what is known as a *Java Development Kit*(JVM). You have to install it if you want to develop Java programs and not just being able to run them. A JVM already contains a JRE so you don't need to install both.

1.4.2 Downloading the interpreter

When you installed the JRE, you've already installed the JVM as well. . However, the JVM alone is not enough to run Meruem programs. That's because it only understands Java bytecodes