Mason
Programming Language Specification

Mike Jasinski

December 15, 2020

Contents

Introduction

Mason is a system programming language that values simplicity, maintainability, productivity and performance. It is designed and built for modern computers, and aims to increase developer productivity by decreasing the friction between the programmer and the programming language. It features strong, static typing, complete control over data structures and memory allocation strategies, and a fast and lean standard library.

Notation

The syntax of a feature is described when the feature is described. Additionally, it can be inferred from code examples at various points in this document. A BNF definition may be added at a later point.

Lexical Elements

Source Code Representation

Source code is text encoded in UTF-8. While Mason makes no effort to support Unicode characters for the source itself, they can be used in comments and string literals.

Tokens

Source code is split into tokens during lexical analysis. Whitespace and newline characters are ignored and serve as token separators (except in string literals).

There are 3 types of tokens:

1. **Identifiers:** Tokens starting with `_` or an alphabetical character and further characters being `_` or alphanumeric. For example: `_test`, `var2`, `obj_type`.
2. **Symbols:** `.` `,` `:` `;` `"` `->` `+` `-` `*` `/` `^` `&` and so on. They can consist of multiple characters, such as `->`.
3. **Literals:**
   (a) **Integer Literals:**

(i) **Decimal Literals:** Optionally one of {`+`, `-`} followed by a combination of digits $[0, 9]$. If the literal has more than one digit, there must not be any leading digits equal to 0. For example, 00012085 is not a decimal literal, but 0, 22, and 910492 are.

(ii) **Binary Literals:** Not a very important feature, may be added at a later point.

(iii) **Hexadecimal Literals:** Not a very important feature, may be added at a later point.

(b) **Floating-Point Literals:**

(i) Decimal literal followed by a `.` and a combination of digits $[0, 9]$. For example: 3.14159, 509.12, 0.0.

(c) **String Literals:**

(i) UTF-8 text enclosed by `""`. For example: "Hello World!".

Comments

The token `//` and any text that follows until end of line is ignored.
The tokens `/*`, `*/` and any text in between is ignored.

Strings

...

Variables

Types

Standard Library

Allocators

Context System

Thread Safety

Scopes

Modules