

The MT Programming Language

Ramsay Carslaw

September 23, 2020

Contents

1	Introduction	2
1.1	What is MT?	2
1.2	Who is MT for?	2
1.3	Who is this manual for?	2
1.4	Design Ideas	2
1.5	Installing MT	3
1.5.1	macOS	3
1.5.2	GNU/Linux	3
1.5.3	Windows	3
1.6	Contributing to MT	3
2	The Basics	4
2.1	Saying Hello	4
2.1.1	Hello World	4
2.1.2	Hello MT	4
2.1.3	Hello User	4
2.2	Numbers Game	5
2.3	The Truth, The Whole Truth and Nothing but the Truth	5
2.4	My Type	5

Chapter 1

Introduction

1.1 What is MT?

The MT programming language is an open source, dynamically typed programming language intended to bridge the gap for new programmers between python and C. MT was implemented in C using a source to byte code compiler into a bytecode interpreter. This means MT is a reasonably fast language.

1.2 Who is MT for?

MT is not intended for any serious development projects, more a tool for learning. MT is a good halfway point between Python and your next language for new programmers. If you are looking to write an operating system MT is probably not the language for you, however if you are looking for a language to practice your data structures then MT will be a safe choice.

1.3 Who is this manual for?

This manual is for people with some programming experience that would like to learn MT. It's not too high level so even basic programming experience should be enough to learn MT. More advanced users can use this book as a reference for syntax where as beginners can learn by completing the exercises at the end of each section.

1.4 Design Ideas

Although, MT supports object oriented programming this is by no means forced. It is perfectly fine to just use functions or code imperatively. Objects are in MT for the cases where it will make your life easier, they are not forced or necessary.

1.5 Installing MT

Depending on your platform getting MT is slightly different. Once you have followed the relevant instructions you can run 'mt' in your terminal to test if everything has installed correctly.

1.5.1 macOS

MT was developed on a mac so this one is pretty easy

```
git clone https://github.com/ramsaycarslaw/mt.git
cd mt
make
make install
```

This does require that the XCode command line tools are installed, to do this just download XCode from the Mac App store.

1.5.2 GNU/Linux

Depending on your GNU/Linux Distribution you may need to configure the makefile. By default the makefile uses clang which is not always installed on linux systems. To install clang use:

```
sudo pacman -S clang
```

Alternatively, change the CC line in the Makefile to whatever C compiler you use. (Note: This may effect the performance of MT). After that follow the following steps.

```
git clone https://github.com/ramsaycarslaw/mt.git
cd mt
make
make install
```

1.5.3 Windows

I don't have a windows machine to test any of this on but you will have to compile from source.

1.6 Contributing to MT

MT is open source and can be found on github. All help is appreciated as MT does not yet natively support arrays. This is the killer feature that needs to be added before MT can become a serious language. The current plan is to implement arrays in MT using classes but a more permanent solution needs to be found.

Chapter 2

The Basics

2.1 Saying Hello

2.1.1 Hello World

The very first program needed by any language is the classic hello world. Hello World in MT is very simple.

```
| print 'hello, world!';
```

There are a few interesting features of MT's hello world. First print is a keyword. Usually, print is a function and it's contents will be contained in brackets. In MT however print is a keyword which makes it more beginner friendly as it works the same as all the other keywords in MT and they can program without understanding the concept of a function. Aside from print being a keyword everything else is pretty standard. The double quotes denote a string and the semi-colon the end of a statement.

2.1.2 Hello MT

To run the program above save it as hello.mt in your favorite text editor. You can then navigate to the directory where you saved that file and run `mt hello.mt` to run the code. You should see 'hello, world!'. If MT warns you about the file not existing check you are in the right directory and that you saved the file. If you get an error like 'no such command mt' check the installation process as something has gone wrong.

2.1.3 Hello User

The next program we will cover is getting input from the users of your program. This is essential for any fully featured program so we will cover that next. This will also introduce a few other concepts like string concatenation.

```
| var name = input('What is your name? ');  
| print 'Hello, ' + name;
```

In this case input is actually a built in function as it needs the brackets. The message (or string) inside the brackets is printed as a prompt for the user to type in. The plus sign in this case is used to join the two strings so if the user types in bob for the input then the result will be “Hello, bob”. Notice the var keyword as well, this is used to tell MT you have a new variable.

2.2 Numbers Game

That covers the basic of getting text in and out of MT. Now it's time we actually compute something. You don't even need to create a file to do some of this. Open your terminal and type in 'mt'. You should see something that looks a bit like:

```
| mt>
```

This is called the REPL, you can use it to evaluate MT statements without a file. For example you can type in:

```
| mt> 10 + 12;
```

You will notice that nothing has happened. This is because we haven't told MT what to do with the result. To see the answer simply add a print beforehand.

```
| mt> print 10 + 12;  
22
```

This time it showed us the answer. This also works with the other basic operators

```
| mt> print 10 * 12;  
120  
mt> print 10 - 12;  
-2  
mt> print 10 / 12;  
0.833333  
mt> print 10 ^ 12;  
1e+12
```

There is no need to specify integers or floats in mt. This is because it is dynamically typed and can change during run time. You can also evaluate more complex statements involving parenthesis.

```
| mt> print (10 + 12 ^ (99 / 6 - (10 + 3.1415)));  
4221.44
```

2.3 The Truth, The Whole Truth and Nothing but the Truth

2.4 My Type