

ET Programming Language

Max Base

Copyright © 2019 Max Base

MAX BASE, ASREZ TEAM
ASREZ.COM

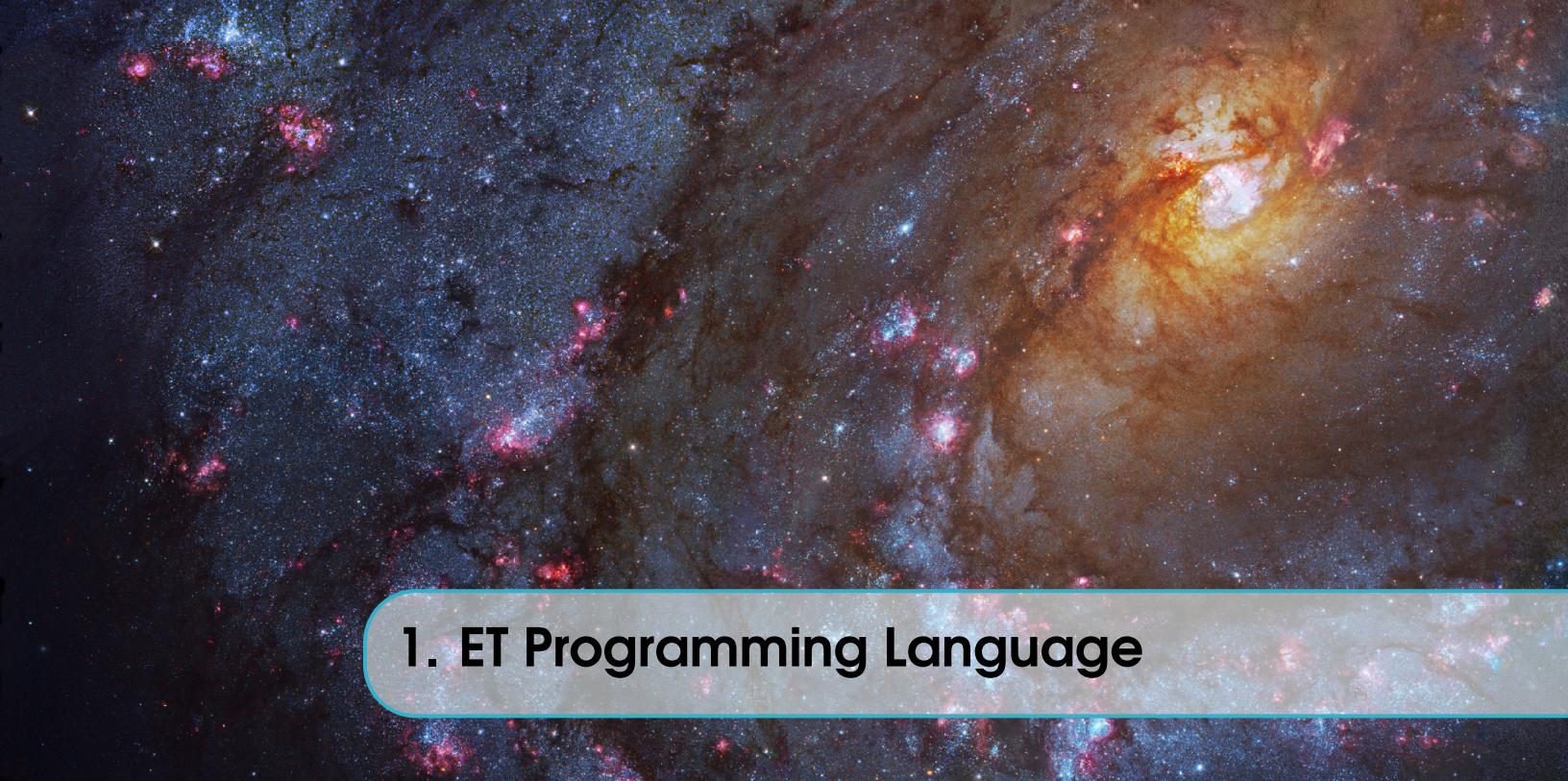
This research was done under the supervision of Dr. Pauline Barmby with the financial support of the MITACS Globalink Research Internship Award within a total of 12 weeks, from June 16th to September 5th of 2014.

First release, Jun 2019



Contents

1	ET Programming Language	5
1.1	Introduction	5
1.2	Data Types	6
1.3	Variable	7
1.3.1	Define Variable	7
1.3.2	Story	7
1.3.3	Name	7



1. ET Programming Language

1.1 Introduction

The ET language has been designed by Max Base in 2012, Which was later developed by the Asrez team. They tried to design a modern, more appropriate language. As simple as possible for humans. Some features of the ET language:

- ET Language is a middle-level language. Programming languages can be divided into three categories: high-level languages, middle level, low-level languages. (Table 1-1) The reason for the middle-level language is that it is also low because it is capable of being closely related to the hardware, like the assembly. And on the other side it is close to human expression and has simple commands and is readable for humans then this's a feature of the high level languages.
- ET language is a flexible and powerful language that does not create any restrictions for the programmer, and you can take and create whatever you think.
- ET language is a portable language. This means that you can run the code written in a system on another system. Without causing trouble or trouble.
- ET language is a multi-platform language. This means that when you design an application for the Windows operating system. You can run it on another operating system, including Linux series.
- The ET programming language has the ability to embed on a platform for another platform. So you can design a program for Windows. You will not need to own a Windows operating system. And this is a unique and wonderful feature.
- The ET programming language has tiny keywords. This means that the number of keywords in this language is small. But this does not make you curious or can not produce any program. However, the number of keywords is not a reason for the strength and speed of the compiler. But these are the reasons for learning this programming language easier and faster.

auto	const	break	continue	return
if	for	foreach	loop	while
do	default	switch	else	elseif
elif	extern	struct	enum	static
goto	union	case	register	typedef
assembly	import	ifdef	ifndef	define

- ET Language is a system language. which you can even design system programs. System programs are programs that allow you to exploit hardware and software communication. Some system programs are: operating system / interpreter / compiler / database / word processor and assembler and ...
- There is a close connection between the ET and the assembly. This means that you can also insert assembly code into this language. Of course, consider that they are controlled by a separate processing engine.
- ET Language is a sensitive letter. This means that words in lowercase letters are different. So be sure to type the letters you type in the letters you type.
- The ET language instructions are included in the following features:
 - There is no limit to the number of words per line.
 - It is recommended to write just one command per line
 - Ability to use; at the end of each instruction is optional
 - Each command can be written in several lines and lines. (You can create spaces or empty space between your guides).
 - You can write comment in your code that you should use the principles of communication.

1.2 Data Types

The purpose of programming is to receive inputs and process them to produce an output. Keep in mind that input values is the most important part of your work.

In the programming language, there are several types of data that we can use in the place where it is needed: float, int, string, char, void, bool, null, ...

The type is to store individual char (such as 'a', 'b'), The type is for storing int for keeping Integer (such as 4, 32, 169) and the float type is used for decimal number such as 15.4, 64.26) And we will explain the type of void later.

But keep in mind that each of them has limitations. So we have several types with the name int, each of which has a limit. We can choose them according to the required range.

int8	8bit, 1byte	-127 until 128
uint8	8bit, 1byte	-127 until 128
int16	16bit, 2byte	-127 until 128
uint16	16bit, 2byte	-127 until 128
int32	32bit, 3byte	-127 until 128
uint32	32bit, 3byte	-127 until 128
int64	64bit, 4byte	-127 until 128
uint64	64bit, 4byte	-127 until 128
size	64bit, 4byte	-127 until 128
float16	16bit, 2byte	-127 until 128
ufloat16	16bit, 2byte	-127 until 128
float32	32bit, 3byte	-127 until 128
ufloat32	32bit, 3byte	-127 until 128
char	8bit, 1byte	-127 until 128
uchar	16bit, 2byte	-127 until 128
mchar	16bit, 2byte	-127 until 128
bool	1bit	true or false
null	1bit	empty

1.3 Variable

A variable is a name for memory words that we put data into. And we may change them and use them during the implementation of the program. To refer to their value, we use the same name, which is the reason we name them so that they can be easily accessed.

To name variables, we can use characters from a to z or A to Z. As well as numbers and _ char.

We do not have a number limit for the length of name of variables.

Note that the name of the variables can not start with the number. (The first letter of the name can not be a number)

1.3.1 Define Variable

As mentioned, variables are memory spaces. So, as the data has the type. We must also specify the type of variables. So the variable definition method will be:

In this case, the variable type can be one of the values in Table 2

'VariableName dataType' ...

1.3.2 Story

1.3.3 Name

The reason for naming ET for this time was initially related to the concept of electronics and technology. But later on the suggestion of Javad Sabet the concept of the name has been changed. This language also means the language of the earth (earth tongue).

■ **Definition 1.3.1 — Cone.** A set $K \in \mathbb{R}^n$, when $x \in K$ implies $\alpha x \in K$.

■ **Example 1.1 — Iudcator function.** $\delta_c(x) = \begin{cases} 0 & x \in C \\ +\infty & \text{elsewhere} \end{cases}$

$\text{dom}\delta_c(x) = C$

■ **Definition 1.3.2 — III.** A function $f : \mathbb{R}^n \rightarrow \bar{\mathbb{R}}$ is

Theorem 1.3.1 $f : \mathbb{R}^n \rightarrow \bar{\mathbb{R}}$ is convex $\iff \forall x, y \in \mathbb{R}^n, \alpha \in (0, 1), f(\alpha x + (1 - \alpha)y) \leq \alpha f(x) + (1 - \alpha)f(y)$.

```
<html>
<head>
<title>Hello</title>
</head>
<body>
Hello
</body>
</html>
```