LUA - ENVIRONMENT

> ### *Try it Option Online*
>
> *We have set up the Lua Programming environment online, so that you can compile and execute all the available examples online. It gives you confidence in what you are reading and enables you to verify the programs with different options. Feel free to modify any example and execute it online.*
>
> *Try the following example using our online compiler available at [CodingGround]*
>
> ```
> #!/usr/local/bin/lua
>
> print("Hello World!")
> ```
>
> *For most of the examples given in this tutorial, you will find a Try it option in our website code sections at the top right corner that will take you to the online compiler. So just make use of it and enjoy your learning.*

Local Environment Setup

If you are still willing to set up your environment for Lua programming language, you need the following softwares available on your computer − *a* Text Editor, *b* The Lua Interpreter, and *c* Lua Compiler.

Text Editor

You need a text editor to type your program. Examples of a few editors include Windows Notepad, OS Edit command, Brief, Epsilon, EMACS, and vim or vi.

Name and version of the text editor can vary on different operating systems. For example, Notepad will be used on Windows, and vim or vi can be used on Windows as well as Linux or UNIX.

The files you create with your editor are called source files and these files contain the program source code. The source files for Lua programs are typically named with the extension **".lua"**.

The Lua Interpreter

It is just a small program that enables you to type Lua commands and have them executed immediately. It stops the execution of a Lua file in case it encounters an error unlike a compiler that executes fully.

The Lua Compiler

When we extend Lua to other languages/applications, we need a Software Development Kit with a compiler that is compatible with the Lua Application Program Interface.

Installation on Windows

There is a separate IDE named "SciTE" developed for the windows environment which can be downloaded from http://code.google.com/p/luaforwindows/ download section.

Run the downloaded executable to install the Lua IDE.

Since its an IDE, you can both create and build the Lua code using the same.

In case, you are interested in installing Lua in command line mode, you need to install MinGW or Cygwin and then compile and install Lua in windows.

Installation on Linux

To download and build Lua, use the following command

```
$ wget http://www.lua.org/ftp/lua-5.2.3.tar.gz
$ tar zxf lua-5.2.3.tar.gz
$ cd lua-5.2.3
$ make linux test
```

In order to install on other platforms like aix, ansi, bsd, generic linux, mingw, posix, solaris by replacing Linux in make Linux, test with the corresponding platform name.

We have a helloWorld.lua, in Lua as follows −

```
print("Hello World!")
```

Now, we can build and run a Lua file say helloWorld.lua, by switching to the folder containing the file using cd, and then using the following command −

```
$ lua helloWorld
```

We can see the following output.

```
hello world
```

Installation on Mac OS X

To build/test Lua in the Mac OS X, use the following command −

```
$ curl -R -O http://www.lua.org/ftp/lua-5.2.3.tar.gz
$ tar zxf lua-5.2.3.tar.gz
$ cd lua-5.2.3
$ make macosx test
```

In certain cases, you may not have installed the Xcode and command line tools. In such cases, you won’t be able to use the make command. Install Xcode from mac app store. Then go to Preferences of Xcode and then switch to Downloads and install the component named "Command Line Tools". Once the process is completed, make command will be available to you.

It is not mandatory for you to execute the "make macosx test" statement. Even without executing this command, you can still use Lua in Mac OS X.

We have a helloWorld.lua, in Lua, as follows −

```
print("Hello World!")
```

Now, we can build and run a Lua file say helloWorld.lua by switching to the folder containing the file using cd and then using the following command −

```
$ lua helloWorld
```

We can see the following output −

```
hello world
```

Lua IDE

As mentioned earlier, for Windows SciTE, Lua IDE is the default IDE provided by the Lua creator team. The alternate IDE available is from ZeroBrane Studio which is available across multiple platforms like Windows, Mac and Linux.

There are also plugins for eclipse that enable the Lua development. Using IDE makes it easier for development with features like code completion and is highly recommended. The IDE also provides interactive mode programming similar to the command line version of Lua.