

# Minetest Mods

## Contents

|  |          |
|--|----------|
| <b>1 Skills/Concepts</b>                   | <b>1</b> |
| 1.1 Console Use . . . . .                  | 1        |
| <b>2 Overview</b>                          | <b>1</b> |
| <b>3 Interpreted vs Compiled Languages</b> | <b>2</b> |
| <b>4 Process</b>                           | <b>2</b> |
| 4.1 Install Lua Interpreter . . . . .      | 2        |
| 4.2 Printing . . . . .                     | 2        |
| 4.3 Variables . . . . .                    | 3        |
| 4.4 Types . . . . .                        | 3        |

## 1 Skills/Concepts

In this module, we'll start learning some programming terminology and concepts. Up to this point, we've done several copy-paste examples for our Minetest mods (which are written in Lua) but haven't really explained how it works in detail.

In this lesson, we'll take a step back and explain some basic programming concepts and terminology so you're better equipped to write mods and other programs.

The following concepts will be touched upon in this module:

1. Console use
2. Installing Lua 5.1 interpreter
3. Basic Lua and Console Interaction
  - Printing in Lua
  - Variables and Types
  - Mathematical Operators
  - User Input

### 1.1 Console Use

This module relies heavily on the command-line. You will be using a terminal emulator to execute the following commands.

| Command | Description                            |
|---------|--|
| cd      | Change directory                       |
| ls      | List directory contents (whats inside) |
| subl    | Sublime text editor                    |
| lua     | Lua interpreter                        |

## 2 Overview

First let's watch a video about why we should learn programming:

Video: Why we should learn programming

Programming continues to become one of the most valuable skill sets of the future. Learning how to program will give you an advantage, even if you decide to not actually become a software developer in the future.

For example, say you went into a law enforcement career in the future. For each of your cases, you may be required provide a summary for each of the cases you work on. Now, lets say your supervisor asks you to provide all the case summaries you've been working on during the year, by the end of the day. Over a year, it's possible you've worked on hundreds of cases! If you know how to program, you could write a script that grabs the summaries from all those reports and copies them into a single document for your boss. Without programming, you might have to spend a day or two copying and pasting.

In this module, we'll start with some of the basics of programming using Lua scripting language.

## 3 Interpreted vs Compiled Languages

In the world of programming, there are two basic types of programming languages; interpreted and compiled.

Compiled Languages: A language where, when the program is compiled, is expressed in the instructions of the target machine. Usually, compiled languages are used when computing resources are a factor (memory, execution time, etc...)

Benefits:

- Faster performance by directly using the native code of the target machine
- Opportunity to apply quite powerful optimizations during the compile stage

Interpreted Languages: The program instructions are not directly executed by the target machine, but instead read and executed by some other program (which normally is written in the language of the native machine). Usually, interpreted languages are used for proto-typing and quick development and where computing resources are not as much of a factor.

Benefits:

- Scripts can be run cross-platform without needing to recompile
- Can make on-the-fly changes to scripts.

Lua is an interpreted language. In this lesson, we will provide instructions to the interactive interpreter to be executed.

## 4 Process

### 4.1 Install Lua Interpreter

First, we'll install the Lua 5.1 Interpreter on your computer. The Lua interpreter will execute the instructions we write later in the module.

0. Open a terminal.
1. Enter the following command:

```
sudo apt-get install lua5.1 lua5.1-doc
```

2. Enter password when prompted and wait for Lua to install.

### 4.2 Printing

One of the first things you'll be taught when learning how to program is how to print. Printing is usually used for debugging but is also useful when first learning a language.

0. In a terminal, start the interactive Lua interpreter by entering the following command:

```
lua
```

This should drop you into the interactive interpreter.

1. Type the following command:

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

2. After pressing Enter, "Hello World" should have been printed to the screen.
3. Try the following command:

```
print(1+2)
```

4. The mathematical statement should be resolved to print 3.

### 4.3 Variables

In programming, variables are used to store information that can be referenced and used in other parts of the program.

0. Lets set a variable. Enter the following command in the Lua interactive interpreter:

```
a = "Hello World"
```

1. We assigned the string "Hello World" to the variable a. We can now use the variable name to access the string. In your interpreter, enter the following command:

```
print(a)
```

The string "Hello World" should have been printed to the screen.

2. Try reassigning the variable a to another string and printing it. What happens?

### 4.4 Types

There are eight basic types in Lua: nil, boolean, number, string, userdata, function, thread, and table. The type function gives the type name of a given value:

```
print(type("Hello world")) --> string
print(type(10.4*3))        --> number
print(type(print))         --> function
print(type(type))          --> function
print(type(true))          --> boolean
print(type(nil))           --> nil
print(type(type(X)))       --> string
```

0. Try entering the commands in the table above to see their output.