Lua

Syntax

Invoking the interpreter with a Lua file name parameter begins execution of the file and continues until the script is finished. When the script is finished, the interpreter is no longer active.

Let us write a simple Lua program. All Lua files will have extension `.lua`. So put the following source code in a `test.lua` file.

```
print("test")
```

Assuming, lua environment is setup correctly, let's run the program using the following code:

```
$ lua test.lua
```

Result:

```
test
```

Let's try another way to execute a Lua program. Below is the modified `test.lua` file

```
#!/usr/bin/env lua5.4
print("test")
```

Now, try to run this program as follows:

```
$ chmod +x test.lua
$ ./test.lua
```

Result:

```
test
```

Lua also provides a mode called interactive mode. In this mode, you can type in instructions one after the other and get instant results. This can be invoked in the shell by using the `lua -i` or just the `lua` command. Once you type in this, press Enter and the interactive mode will be started as shown below.

```
$ lua
Lua 5.4.4  Copyright (C) 1994-2022 Lua.org, PUC-Rio
>
```

You can print something using the following statement:

```
print("test")
```

Result:

```
$ lua
Lua 5.4.4  Copyright (C) 1994-2022 Lua.org, PUC-Rio
> print("test")
test
>
```

Tokens in Lua

A Lua program consists of various tokens and a token is either a keyword, an identifier, a constant, a string literal, or a symbol. For example, the following Lua statement consists of three tokens:

```
io.write("Hello world, from ", _VERSION, "!\n")
```

Result:

```
Hello world, from Lua 5.4!
file (0x7fb1521ef5a0)
```