



Programming Language

What is Lua?

Lua is:

1. Scripting language
2. Dynamic language
3. Moon in Protogués
4. Written in ANSI C
5. Highly portable
6. Extremely Fast
7. Open Source

Lua is not:

1. Best Language
2. Game Engine
3. IDE

Lua can be:

1. Library
2. Shared Object

Hello, world!

```
print("hello, world!")
```

Run your app


```
user@laptop ~$ lua ./hello_world.lua  
hello, world!  
user@laptop ~$
```

Easy!

Running lua interpreter

Commenting your code

-- Single line comment

--[[Multi-line
Comment]]

Variables

Local Variables:

local x = 30

Global Variables:

$$x = 30$$

Looping

for loop:

```
for index = 1, 5 do  
    print( index )  
end
```

```
for index = 10, 1, -1 do  
    print( index )  
end
```

While loop:

```
while cond do
  -- infinit loop
end
```


Repeat loop:

repeat

-- statements

until true

Functions

First class functions

```
local oldprint = print
function print(s)
  if s == "foo" then
    oldprint("bar")
  else
    oldprint(s)
  end
end
```

```
function addto(x)
    return function(y)
        return x + y
    end
end
fourplus = addto ( 4 )
print ( fourplus ( 3 ) ) -- prints 7
```

Tables

table = { key = value }

Examples

```
a_table = { }
```

```
-- empty table
```

```
a_table = { x = 10 }
```

```
print( a_table [ "x" ] )
```

```
-- prints 10
```

```
b_table = a_table  
b_table["x"] = 20  
print( b_table["x"] )  
print( a_table.x )  
-- both prints 20
```

Tables as namespaces

```
Point = { }
```

```
Point.new = function(x, y)  
  return{x = x, y = y}  
end
```

```
Point.set_x = function(point, x)  
  point.x = x  
end
```

```
pl = Point.new(30, 20)  
Point.set_x(pl, 10)
```

Tables as arrays

```
array = { "a", "b", "c", "d" }  
print(array[2]) -- prints b
```

```
array[0] = "z"  
--[[ illegal ]]
```

```
print(#array)  
-- prints 4
```

O.O. programming


```
Vector = { }
```

```
function Vector:new(x, y, z)  
    local object = { x = x, y = y, z = z }  
    setmetatable ( object, { __index = Vector } )  
  
    return object  
end
```

```
function Vector:print()  
    print(self.x..", "..self.y..", "..self.z);  
end
```

```
v1 = Vector:new(1, 2.5, 0.3)
```

```
v1.x = 2.2
```

```
v1:print ( )
```

Importing files

```
local lib = require( "lib" )
```

Or

```
require( "lib" )
```

Thank you