

Table of Contents



- [1. Installation](#)
- [2. Hello World](#)
- [3. Variables and Data Types](#)
- [4. Comments](#)
- [5. Control Structures](#)
 - [If-Else Statement:](#)
 - [While Loop:](#)
 - [For Loop:](#)
 - [Repeat-Until \(Do-While\):](#)
- [6. Functions](#)
- [7. Tables \(Arrays & Dictionaries\)](#)
- [8. Iterating Over Tables](#)
- [9. String Manipulation](#)
- [10. Operators](#)
 - [Arithmetic:](#)
 - [Relational:](#)
 - [Logical:](#)
- [11. Metatables and Operator Overloading](#)
- [12. Error Handling](#)
- [13. Modules \(Import/Export\)](#)
- [14. File I/O](#)
- [15. Coroutines \(Multitasking\)](#)
- [16. Object-Oriented Programming \(OOP\)](#)
- [17. Common Lua Functions](#)
 - [Example: FizzBuzz in Lua](#)

1. Installation

- Windows:
Download from <https://lua.org/download.html>
- Linux/Mac:

```
sudo apt install lua5.3 # Ubuntu
brew install lua        # macOS
```

- Check Version:

```
lua -v
```

2. Hello World

```
print("Hello, Lua!")
```

3. Variables and Data Types

```
x = 10          -- Number
y = "Lua"       -- String
z = true        -- Boolean
n = nil         -- Nil (null)
```

- Dynamic Typing: No need to declare types.
- Multi-assignment:

```
a, b, c = 1, "hello", 3.14
```

4. Comments

```
-- Single-line comment
--[[
  Multi-line comment
]]
```

5. Control Structures

If-Else Statement:

```
x = 15
if x > 10 then
    print("x is greater than 10")
elseif x == 10 then
    print("x equals 10")
else
    print("x is less than 10")
end
```

While Loop:

```
i = 1
while i <= 5 do
    print(i)
    i = i + 1
end
```

For Loop:

```
for i = 1, 5 do
    print("Loop:", i)
end
```

```
-- Reverse Loop
for i = 5, 1, -1 do
    print(i)
end
```

Repeat-Until (Do-While):

```
x = 1
repeat
    print(x)
    x = x + 1
until x > 5
```

6. Functions

```
function greet(name)
  print("Hello, " .. name)
end
```

```
greet("Lua")
```

- Returning Values:

```
function add(a, b)
  return a + b
end
print(add(5, 7))
```

- Anonymous Functions:

```
square = function(x) return x * x end
print(square(4))
```

7. Tables (Arrays & Dictionaries)

Arrays:

```
arr = {10, 20, 30, 40}
print(arr[1])  -- Lua is 1-indexed
```

Dictionaries (Key-Value):

```
dict = {name = "Lua", year = 2024}
print(dict.name)
```

Mixed Table:

```
person = {name = "John", age = 25, [1] = "Developer"}
```

```
print(person[1])
print(person.name)
```

8. Iterating Over Tables

```
for key, value in pairs(dict) do
    print(key, value)
end
```

9. String Manipulation

```
str = "Lua Programming"
print(#str)           -- String length
print(string.upper(str)) -- Uppercase
print(string.lower(str)) -- Lowercase
print(string.sub(str, 1, 3)) -- Substring
```

10. Operators

Arithmetic:

+ - * / % ^ -- Power (x^y)

Relational:

== ~= < > <= >=

Logical:

and or not

11. Metatables and Operator Overloading

```
setmetatable(t, {__add = function(a, b) return a + b end})
```

12. Error Handling

```
success, msg = pcall(function()  
    error("An error occurred!")  
end)  
if not success then  
    print("Error: " .. msg)  
end
```

13. Modules (Import/Export)

Module (math_utils.lua):

```
local math_utils = {}  
function math_utils.add(a, b)  
    return a + b  
end  
return math_utils
```

Import Module:

```
local math_utils = require("math_utils")  
print(math_utils.add(5, 10))
```

14. File I/O

Write to File:

```
file = io.open("example.txt", "w")
file:write("Hello, Lua!")
file:close()
```

Read from File:

```
file = io.open("example.txt", "r")
content = file:read("*all")
file:close()
print(content)
```

15. Coroutines (Multitasking)

```
co = coroutine.create(function()
    for i = 1, 5 do
        print(i)
        coroutine.yield()
    end
end)
```

```
coroutine.resume(co)
coroutine.resume(co)
```

16. Object-Oriented Programming (OOP)

```
Person = {}
function Person:new(name, age)
    obj = {name = name, age = age}
    setmetatable(obj, self)
    self.__index = self
    return obj
end
```

```
john = Person:new("John", 30)
```

```
print(john.name)
```

17. Common Lua Functions

Function	Description
<code>type()</code>	Returns the type of a variable
<code>tostring()</code>	Converts to string
<code>tonumber()</code>	Converts to number
<code>table.insert()</code>	Inserts element into a table
<code>table.remove()</code>	Removes element from a table
<code>ipairs()</code>	Iterates over array-like tables
<code>pairs()</code>	Iterates over key-value tables

Example: FizzBuzz in Lua

```
for i = 1, 20 do
  if i % 3 == 0 and i % 5 == 0 then
    print("FizzBuzz")
  elseif i % 3 == 0 then
    print("Fizz")
  elseif i % 5 == 0 then
    print("Buzz")
  else
    print(i)
  end
end
```