

Classes with Classic



Classes are code templates providing a common place to store values and behaviors (functions). Objects are variables that are created from a class.

Differently from many other languages, Lua does not have built-in classes, but several people have created Class modules.

Classic is one of the most simple ones:
<https://github.com/rxi/classic>

Creating a Point class

```
local Class = require "classic"
local Point = Class:extend()
```

```
function Point:new(x, y)
    self.x = x
    self.y = y
end
```

```
function Point:moveX(dx)
    self.x = self.x + dx
end
```

Creating an object from a class

```
local corner = Point(100, 200)
corner:moveX(50)
```

Classes with Classic



Classes are code templates providing a common place to store values and behaviors (functions). Objects are variables that are created from a class.

Differently from many other languages, Lua does not have built-in classes, but several people have created Class modules.

Classic is one of the most simple ones:
<https://github.com/rxi/classic>

Creating a Point class

```
local Class = require "classic"
local Point = Class:extend()
```

```
function Point:new(x, y)
    self.x = x
    self.y = y
end
```

```
function Point:moveX(dx)
    self.x = self.x + dx
end
```

Creating an object from a class

```
local corner = Point(100, 200)
corner:moveX(50)
```

Classes with Classic



Classes are code templates providing a common place to store values and behaviors (functions). Objects are variables that are created from a class.

Differently from many other languages, Lua does not have built-in classes, but several people have created Class modules.

Classic is one of the most simple ones:
<https://github.com/rxi/classic>

Creating a Point class

```
local Class = require "classic"
local Point = Class:extend()
```

```
function Point:new(x, y)
    self.x = x
    self.y = y
end
```

```
function Point:moveX(dx)
    self.x = self.x + dx
end
```

Creating an object from a class

```
local corner = Point(100, 200)
corner:moveX(50)
```

Classes with Classic



Classes are code templates providing a common place to store values and behaviors (functions). Objects are variables that are created from a class.

Differently from many other languages, Lua does not have built-in classes, but several people have created Class modules.

Classic is one of the most simple ones:
<https://github.com/rxi/classic>

Creating a Point class

```
local Class = require "classic"
local Point = Class:extend()
```

```
function Point:new(x, y)
    self.x = x
    self.y = y
end
```

```
function Point:moveX(dx)
    self.x = self.x + dx
end
```

Creating an object from a class

```
local corner = Point(100, 200)
corner:moveX(50)
```

The anatomy of a class

- We first require the "classic" module and locally call it Class
- We create the new Point class by extending the classic Class.
- The class must define the new() constructor.
- The constructor defines the class variables. Often by initializing them with the constructor's arguments.

The object

- The object is created by calling the create() function of Point.

Class modules

Most of the time you will put the class in a separate file: create a file containing a local definition of a class and returning the local variable at the end of the file.

To be noted

- Per convention, the name of the class starts with a capital letter;
- When using "Classic", the "constructor" is called new().
- The functions in the class are defined and called by using a semi colon and not a dot.
- The name of the variables start with a lowercase letter.

The anatomy of a class

- We first require the "classic" module and locally call it Class
- We create the new Point class by extending the classic Class.
- The class must define the new() constructor.
- The constructor defines the class variables. Often by initializing them with the constructor's arguments.

The object

- The object is created by calling the create() function of Point.

Class modules

Most of the time you will put the class in a separate file: create a file containing a local definition of a class and returning the local variable at the end of the file.

To be noted

- Per convention, the name of the class starts with a capital letter;
- When using "Classic", the "constructor" is called new().
- The functions in the class are defined and called by using a semi colon and not a dot.
- The name of the variables start with a lowercase letter.

The anatomy of a class

- We first require the "classic" module and locally call it Class
- We create the new Point class by extending the classic Class.
- The class must define the new() constructor.
- The constructor defines the class variables. Often by initializing them with the constructor's arguments.

The object

- The object is created by calling the create() function of Point.

Class modules

Most of the time you will put the class in a separate file: create a file containing a local definition of a class and returning the local variable at the end of the file.

To be noted

- Per convention, the name of the class starts with a capital letter;
- When using "Classic", the "constructor" is called new().
- The functions in the class are defined and called by using a semi colon and not a dot.
- The name of the variables start with a lowercase letter.

The anatomy of a class

- We first require the "classic" module and locally call it Class
- We create the new Point class by extending the classic Class.
- The class must define the new() constructor.
- The constructor defines the class variables. Often by initializing them with the constructor's arguments.

The object

- The object is created by calling the create() function of Point.

Class modules

Most of the time you will put the class in a separate file: create a file containing a local definition of a class and returning the local variable at the end of the file.

To be noted

- Per convention, the name of the class starts with a capital letter;
- When using "Classic", the "constructor" is called new().
- The functions in the class are defined and called by using a semi colon and not a dot.
- The name of the variables start with a lowercase letter.