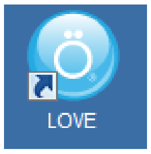
In this lesson, you'll learn to create your first program.

I put a shortcut onto your desktop, the screen that you look at after you log in.



When you double-click on this shortcut, a folder will open. This folder is located in your **Documents** folder and is called something like **LovePrograms**. This is the folder where you will create your programs.

Each new **Love** program will have its own folder. The **LovePrograms** folder already has a few program folders in it. We'll ignore these for now.

Step 1:

Create a folder for your first program by right-clicking over an empty area of the **LovePrograms** folder and selecting **New** and then **Folder** from the popup menu. This will create a new folder that you can rename to Project01 (or anything else that you would like). Double-click the Project01 folder to open it.

Step 2:

Love programs always have a file called **main.lua**. This is the place that **Love** looks for its first instructions.

Create the **main.lua** file by right-clicking over an empty area of the Project01 folder and selecting **New** and then **Text Document**. This will create a file that you can rename to **main.lua**. When you change the extension from .txt to .lua Windows will warn you, but that's OK.

Step 3:

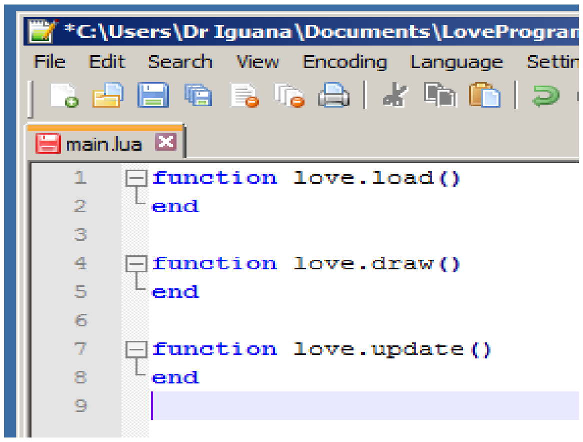
To write the instructions that Love will execute we will use an editor called **NotePad++**. An editor is like a simple word processor but geared towards writing simple text documents**. NotePad++** is a nice programming editor because it understands a bit about how Lua programs should look. This is helpful to us.

Double-click the **main.lua** file to open it in **NotePad++**.

These are the three steps you'll follow each time that you want to create a new **Love** program. If you want to modify an existing program, you'll skip to step 3 and open the existing **main.lua** file.

A Love Program

Now that we have **main.lua** open in **NotePad++**, let's type in a bare bones **Love** program. Type the text on lines 1 through 8**.** When you are done, select **File**, **Save** from the menu or type **Ctrl+S** (hold down the Ctrl key and type S).


*C:\Users\Dr Iguana\Documents\LoveProgran
File Edit Search View Encoding Language Settir
main.lua

```
1 function love.load()
2 end
3
4 function love.draw()
5 end
6
7 function love.update()
8 end
9
```

Here are a few things you should notice.

First, each line is numbered. If we make a mistake in our **Lua** instructions, **Love** will tell us the line where the error occurred.

Second, **NotePad++** has colored the words **function**, **end** and the brackets ( ) blue. This is because **NotePad++** understands **Lua** key words. Key words have specific meanings in a language. This would be like **girl**, **run** or **house** in English.

Next, NotePad++ has connected each occurrence of **function** to the nearest following **end** keyword with a gray line to the left of the code. In **Lua** the keywords **function** and **end** enclose a group of instructions that can be executed as a group. This group is naturally called a function.

We use functions in real life.

```
Function fred.washclothes()
   gather clothes
   put clothes in washing machine
   put detergent in washing machine
   start washing machine
   wait for washing machine to finish
   remove clothes from washing machine
end
```

When someone says "fred.washclothes()" they expect that Fred will execute the instructions inside the function. This is more efficient that telling Fred the individual steps each time.

In **Lua**, as well as most languages, functions are used to create groups of instructions that perform common tasks.

Understanding the Bare Bones Love Program

We've learned that functions exist to perform common tasks. The bare bones **Love** program contains three functions.

When you run **Love** and point it at your program's folder, **Love** opens the **main.lua** file and calls the function love.load() once and only once.

The purpose of the function love.load() is to do those tasks that only need to be done once. These might include setting the size of the **Love** program window. Or you might load a fancy font and set this as the default for your entire program.

When the function love.load() finishes, it returns control to the **Love** program.

Next, **Love** calls the function love.draw(). As you might imagine, the purpose of this function is to draw everything that you want onto the **Love** program window. This might be text like "Fred is naughty" or pictures, or shapes. There are lots of things that you can do.

When the function love.draw() finishes, it returns control to the **Love** program.

Finally, **Love** calls the function love.update(). When this function finishes it also returns control to the **Love** program. We'll learn more about this function later so don't worry about it for the moment.

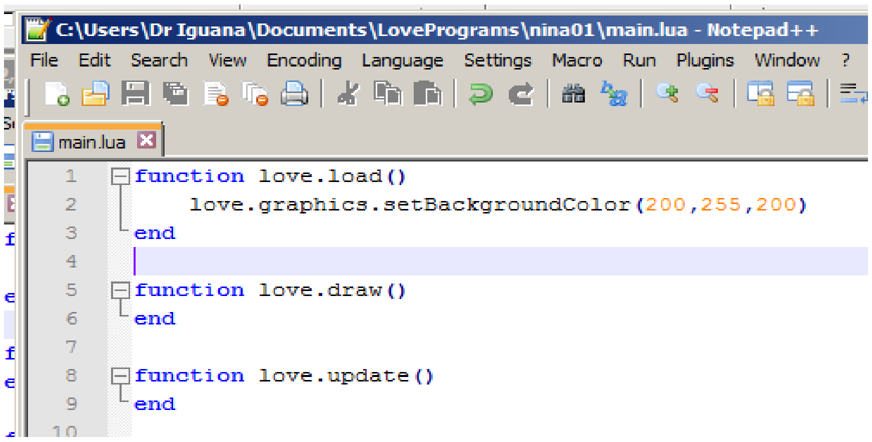
The main thing is that after **Love** calls love.update() it then loops back and calls love.draw() again. It keeps calling these two functions one after the other until you end the program.

Expanding Your First Program

Let's just run the bare bones program to see what happens. You can do this in **NotePad++** by choosing **Run** and then **Launch in Love 2D** from the menu, or holding down **Shift** and **Ctrl**, and the typing **L**

You should see a black window with the caption 'Untitled'; not very impressive.

Let's add an instruction to love.load() line 2. Remember to save your addition using Ctrl+S.


C:\Users\Dr Iguana\Documents\LovePrograms\nina01\main.lua - Notepad++
File Edit Search View Encoding Language Settings Macro Run Plugins Window ?
main.lua

```
1  function love.load()
2      love.graphics.setBackgroundColor(200,255,200)
3  end
4  
5  function love.draw()
6  end
7  
8  function love.update()
9  end
10
```

Note that **Love** and **Lua** are case sensitive. That means that UPPERCASE and lowercase letters are different. The following are all different and only the last one correct. I've marked the letters that are wrong.

Love.graphics.setBackgroundColor(200,255,200)	Wrong
love.graphics.setbackgroundcolor(200,255,200)	Wrong
love.graphics.setBackgroundColor(200,255,200)	Correct

The function love.graphics.setBackgroundColor() changes the color of the **Love** window. Try running the program using Shift+Ctrl+L. The background should be a pale green. The numbers inside the brackets are called arguments and affect the function's behavior.

We could modify the function fred.washclothes() to fred.washclothes(who).

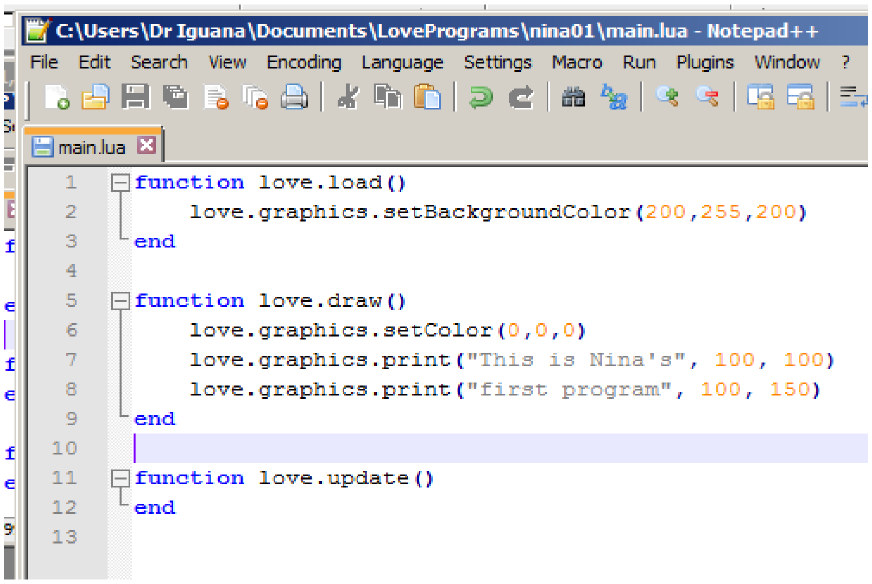
Then someone could call fred.washclothes("Jack") or fred.washclothes("Jill"). This way one function can do many similar but different things.

In the love.graphics.setBackgroundColor function the three arguments specify the desired color. Colors in many programs are specified in something called RGB or Red, Green, Blue. In **Lua**, 0 means none of a color and 255 means the most of a color.

0,0,0 means no Red, Green or Blue which is black.
255,255,255 means the most of Red, Green, and Blue which is white.
255,0,0 means the most of Red, and no Green or Blue which is pure red.

Let's finish off the first lesson by printing something into the **Love** window.

Add the instructions on lines 6, 7 and 8 to the love.draw() function.


C:\Users\Dr Iguana\Documents\LovePrograms\nina01\main.lua - Notepad++
File Edit Search View Encoding Language Settings Macro Run Plugins Window ?
main.lua

```
1  function love.load()
2      love.graphics.setBackgroundColor(200,255,200)
3  end
4
5  function love.draw()
6      love.graphics.setColor(0,0,0)
7      love.graphics.print("This is Nina's", 100, 100)
8      love.graphics.print("first program", 100, 150)
9  end
10
11 function love.update()
12 end
13
```

The first instruction calls love.graphics.setColor() and specifies the color of the next thing that you will print or draw. Remember that 0, 0, 0 means black.

The next instruction calls love.graphics.print(). The three arguments are “what to print”, the horizontal location, and the vertical location. Text is enclosed in quotes in most languages. For the horizontal position zero is at the left. For the vertical position zero is at the top. You can play with the arguments and see what happens. You can change the numbers to print in different locations. You can change the text inside the “quotes“ to print a different message.