Do It Yourself Academy
D·I·Y·A
Learn | Achieve | Become

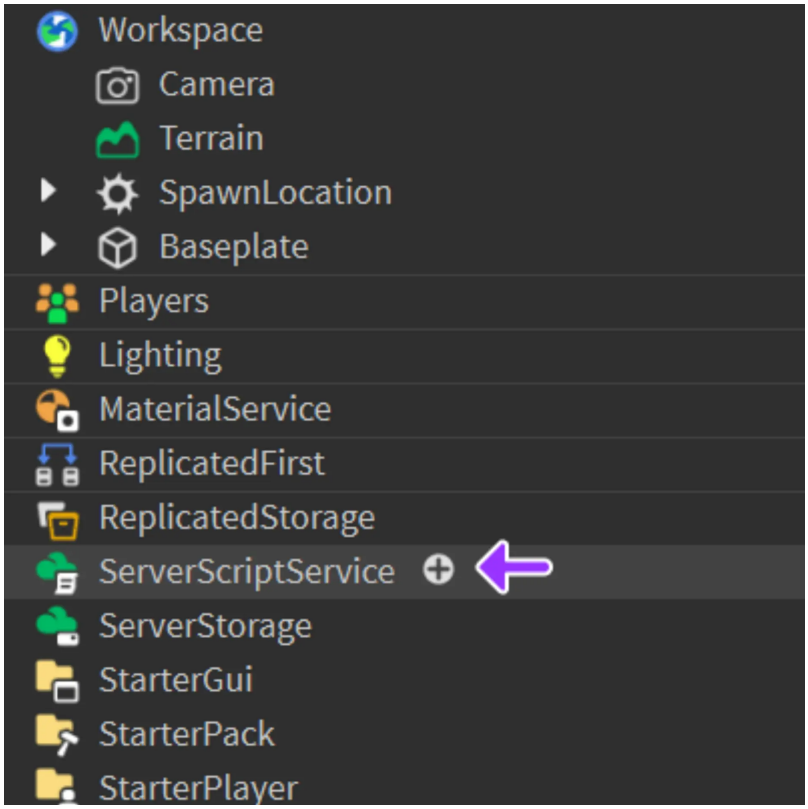
Roblox 1 : Lua Scripting Fundamentals

Table of Contents

Creating a Script

Scripts are commonly created in ServerScriptService, a special folder made just for holding scripts.

1. In Explorer, hover over ***ServerScriptService*** to see the ***+*** button.

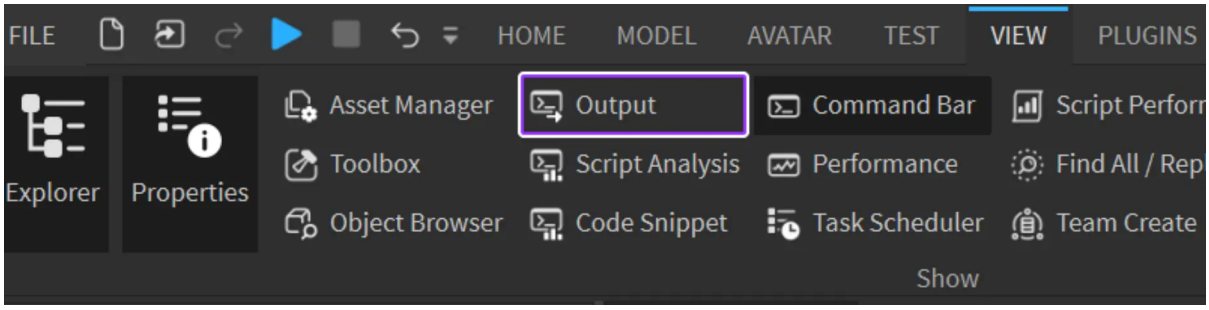


2. Click the ***+*** button and select ***Script***. A new script will be created and the script editor will open.

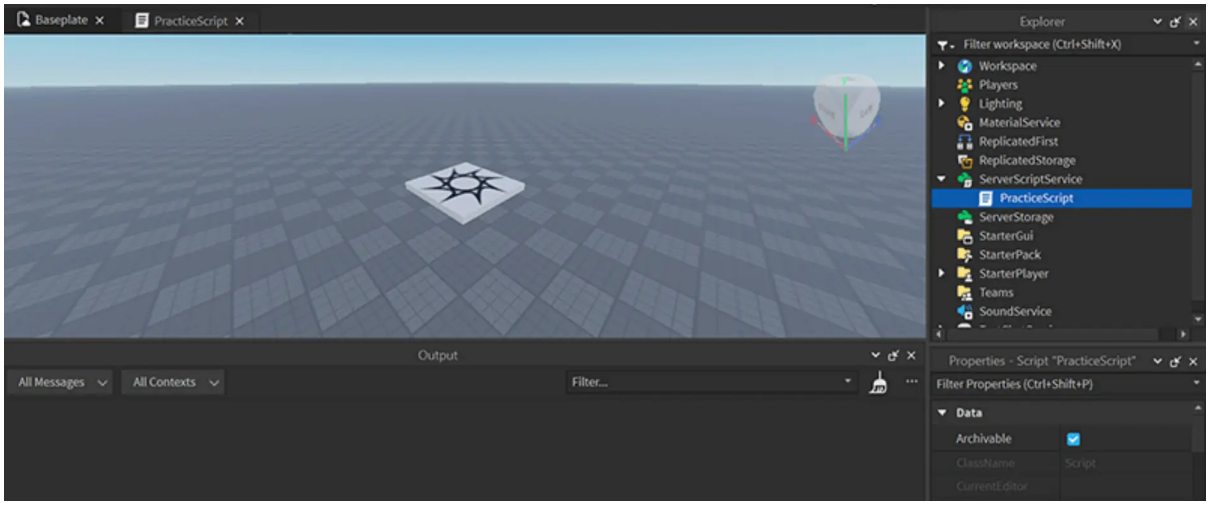
3. Right-click ***Script*** and select ***Rename***. Name the script ***PracticeScript***.

Testing Output

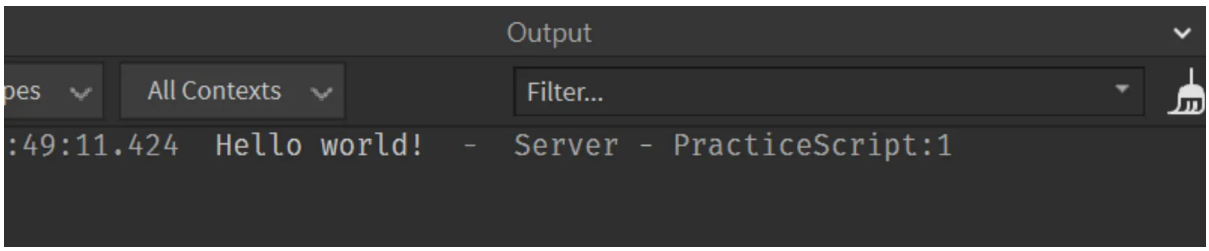
1. Select the ***View*** menu tab.
2. Click ***Output***.



The window will appear at the bottom of Roblox Studio.



3. To test the script, click ***Play***. `Hello world!` will show up in the Output window.



4. Click Stop to end the playtest. You can now return to the Script tab.

Variables

In Lua, to declare a new variable, type `local`, then type the name for the new variable. A variable that can hold a player name might look like: `local playerName`

New variables are empty. To assign it a value, or put something inside its container, use the `=` symbol. In this case, assign the variable the name of your favorite animal.

After the variable name, type `=` to assign the value to the variable.

```
local myAnimal = "Tiger"
```

To print the variable, use the `print()` function

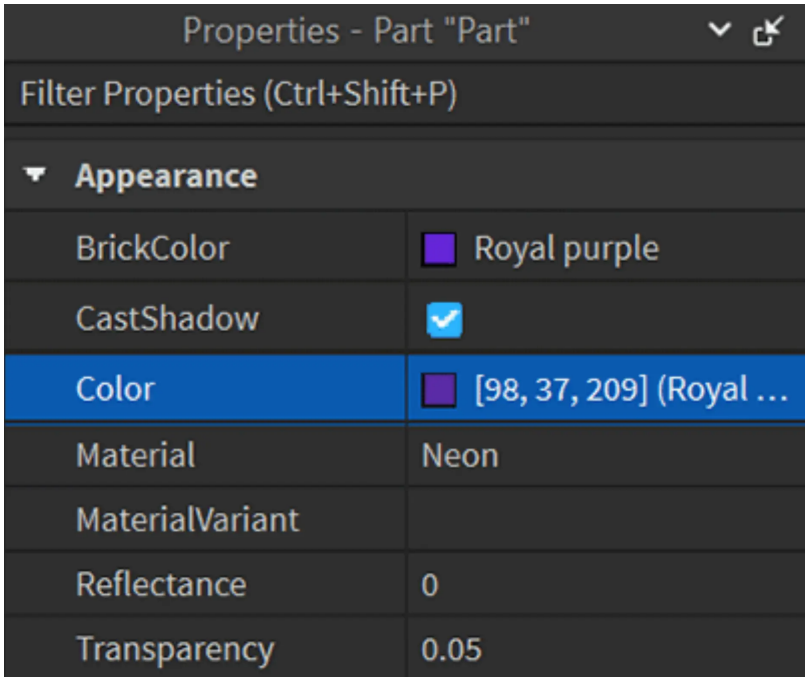
```
local myAnimal = "Tiger"
print(myAnimal)
```

Object Properties

The Properties window can be used to learn about an object's properties. Use it to take a look at a part's properties.

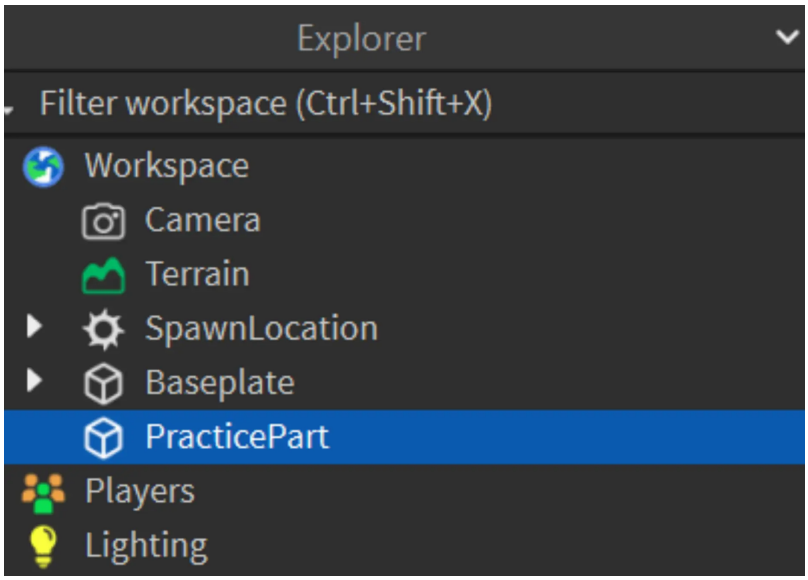
1.Select a part.

1. In the ***Properties*** Window on the bottom-right, look at the different properties that can be changed, like color, size, material and transparency. You can also change most properties in this window from within a script



Changing Part's Property

To make changes to a part, you must be able to describe the part's location. The ***Explorer*** is an excellent tool for referencing locations. In this case, ***PracticePart*** is under ***Workspace***.



Now that you know where the part is, the part's location needs to be translated into something a script can understand.

1. Based on the picture, we can see that ***PracticePart*** is located under ***Workspace***. So, to do anything to affect the part, type `workspace.PracticePart`.

```
workspace.PracticePart
```

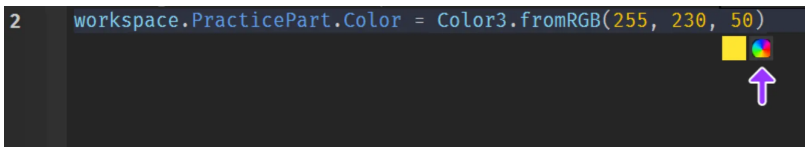
1. To change ***PracticePart***'s property, type `.Color` to access the ***Color*** property

```
workspace.PracticePart.Color
```

3. For the part, the script will change its Color property to a new Color3, a data type that stores colors. Type `= Color3.fromRGB()` This code will allow you to assign a new color.

```
workspace.PracticePart.Color = Color3.fromRGB()
```

4. RGB color values can be manually typed inside the parentheses, but using the color picker is easier. Click inside the parentheses, and then click the color wheel. Follow the popup to create a color.



Parents and Children Relationship

Instead of running scripts from ***ServerScriptService***, you may want to attach a script to the part.

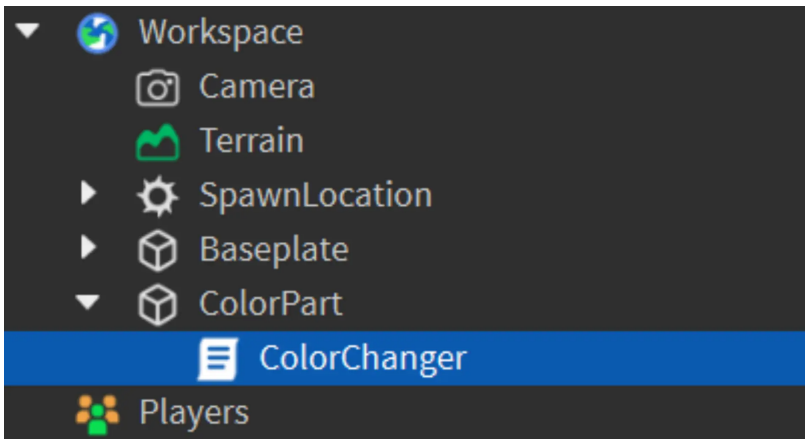
To do this, you must understand parent and child relationships.

Parents and children are ways to describe the hierarchy between different objects. Anytime you've added a new part to Workspace, Workspace has been the parent object, and the part became a child object. When you added a script to ***ServerScriptService***, ***ServerScriptService*** was the parent, and the script was a new child.

1. Create a new part and rename it. This lesson will use ColorPart.
2. Right-click the part and select ***Insert Object > New Script***. Rename the script ***ColorChanger***.
3. Copy and paste the code below into ColorChanger. This version of the code is the same that you used before. It assigns a specific part to a variable.

```
local colorPart = workspace.ColorPart
colorPart.Color = Color3.fromRGB(50, 240, 255)
```

A parent is anything with objects, like scripts or parts, attached below it. Anything under the parent is its children. In the example below, ***ColorPart*** is the parent, and ***ColorChanger*** is the child.



With the current script, you can only change the color of a single part named ***ColorPart***. To change any part's color, you can design the code to work on the script's parent object, whatever it happens to be named. The code script.Parent, will go up the hierarchy and find the object the script is attached to. To make use of this relationship, you can use change the previous code to the example below.

```
local colorPart = script.Parent
colorPart.Color = Color3.fromRGB(50, 240, 255)
```

Creating a Function

1. Create a new script in ***ServerScriptService***.
2. Rename the script ***FunctionsPractice***.
3. Type local function `printFood()`, then press Enter on your keyboard. It should autocomplete the function to look like this

```
local function printFood()

end
```

4. All of the code for your function has to be typed between the lines local function `printFood()` and `end`. Any code not between those two points won't run when the function does. For example like below

```
local function printFood()
  print("Pizza!")
end
```

5. Functions won't run until they are called. To call a function, type the function's name including the `()` at the end.

```
local function printFood()
  print("Pizza!")
end
printFood()
```

6. Run the code as shown in the previous to see the output