

ROBLOX PROGRAMMING

RoboCatz

AGENDA

Introduction

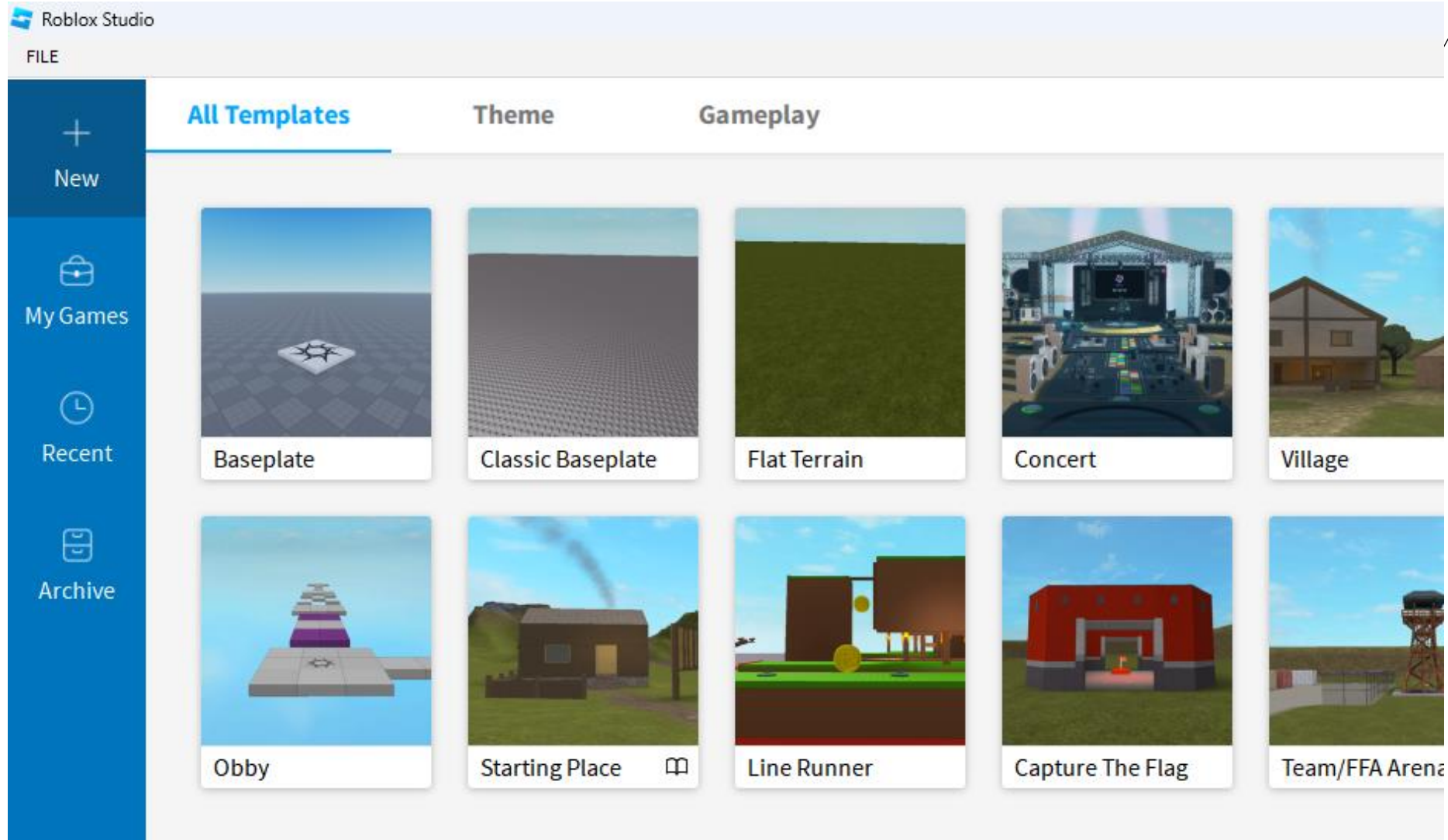
Primary goals

Areas of growth

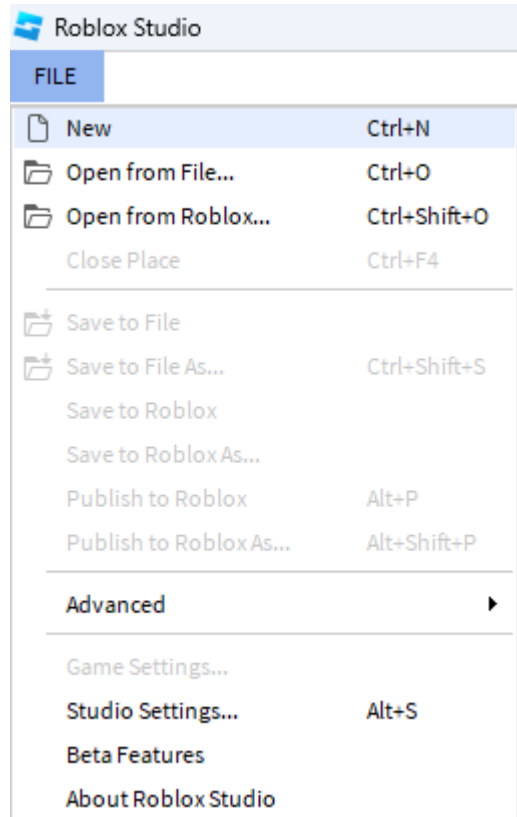
Timeline

Summary

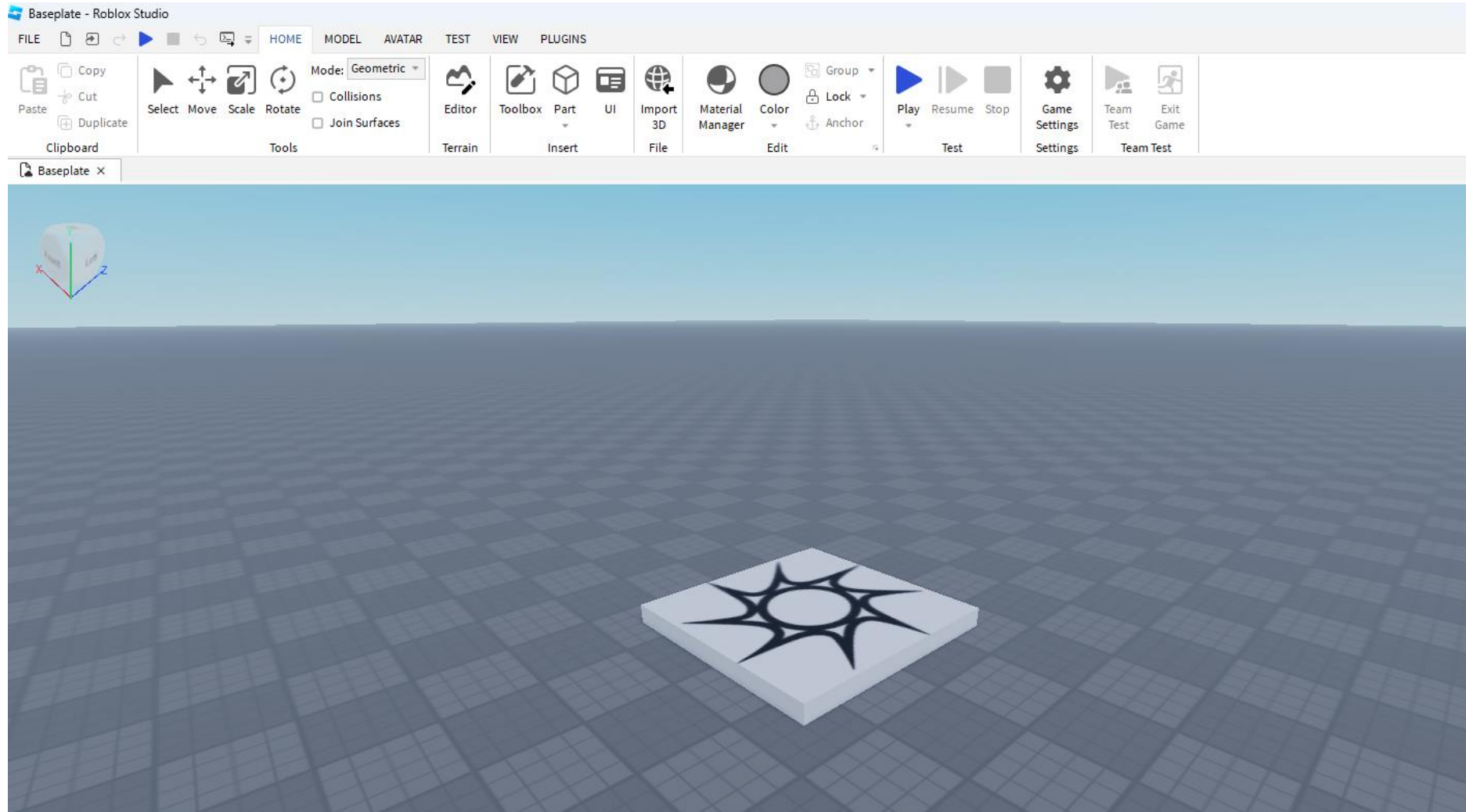
Roblox Studio



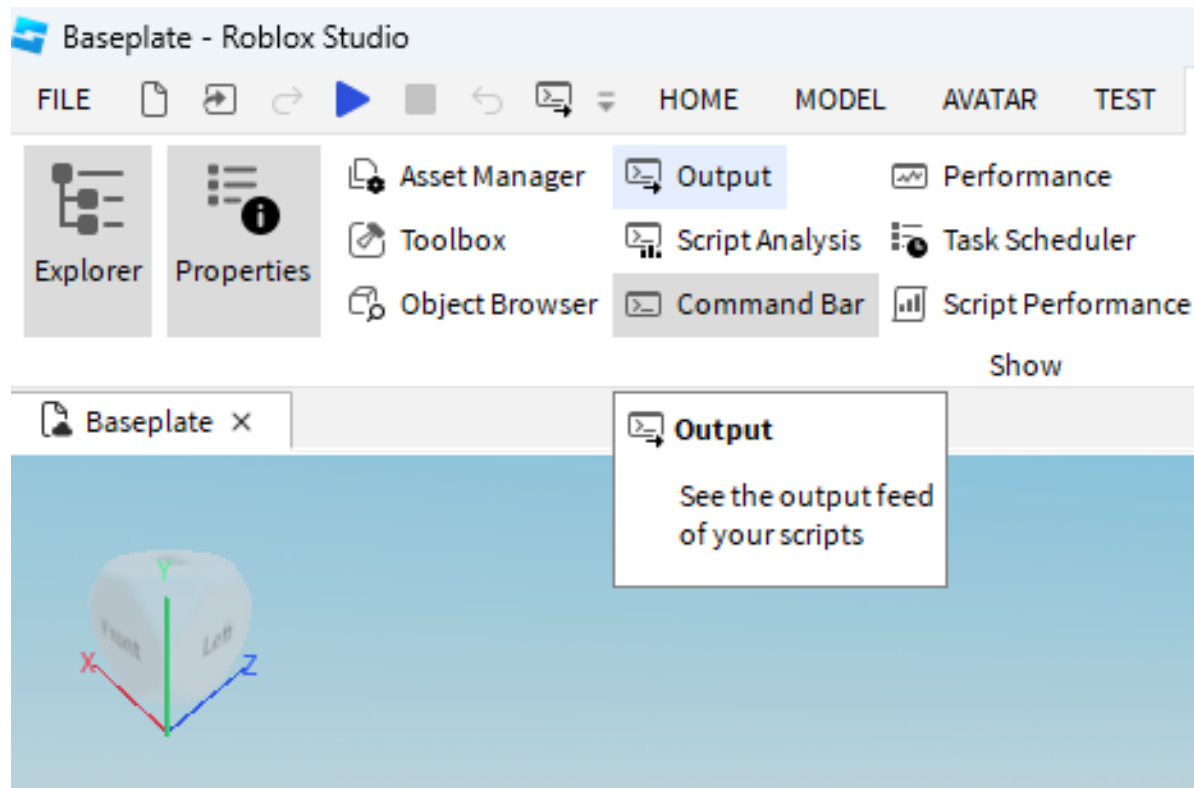
Roblox Studio - File Menu



Roblox Studio - Interface



ROBLOX STUDIO - ACTIVATE THE OUTPUT PANEL

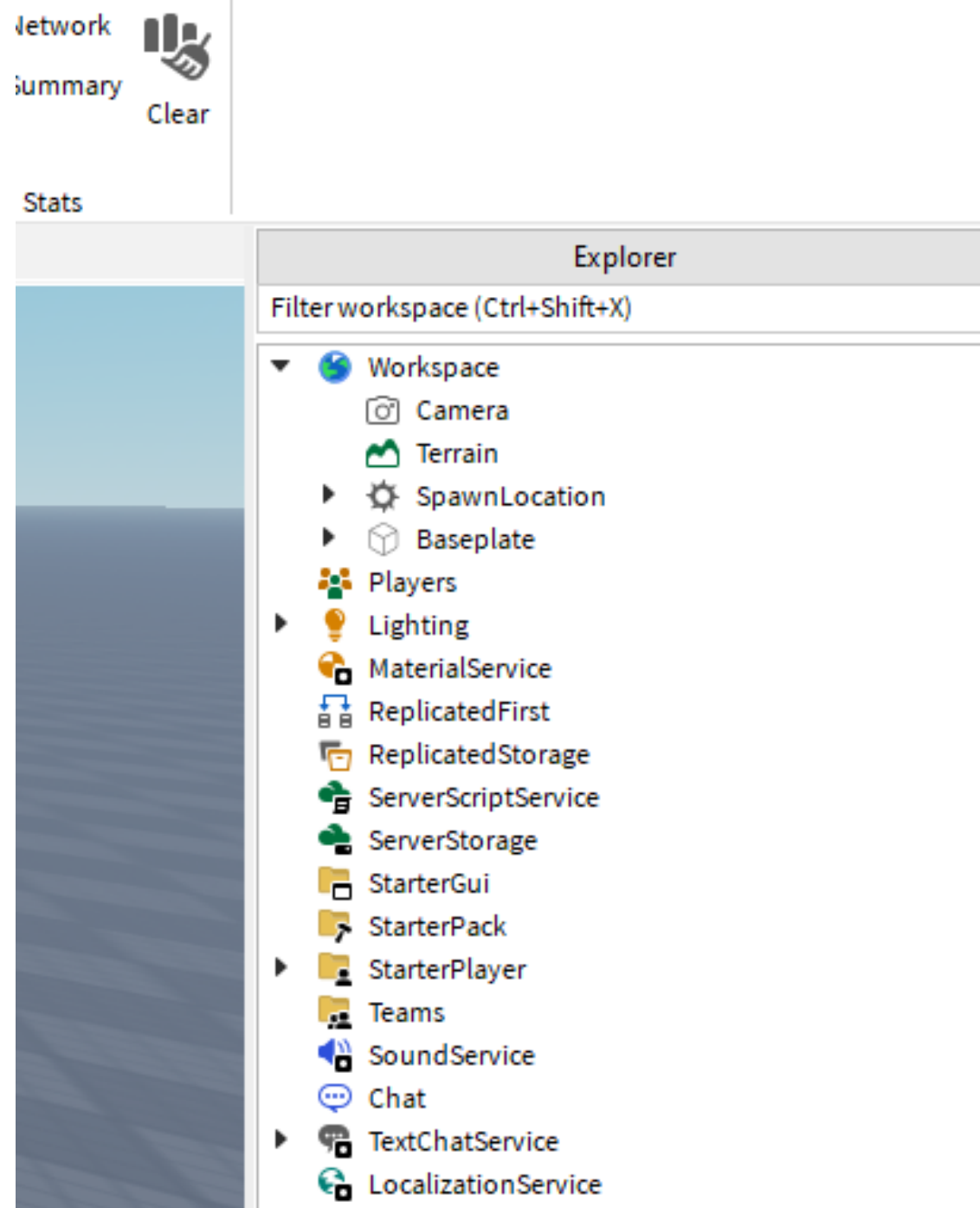


Roblox Studio - Output panel is at the bottom

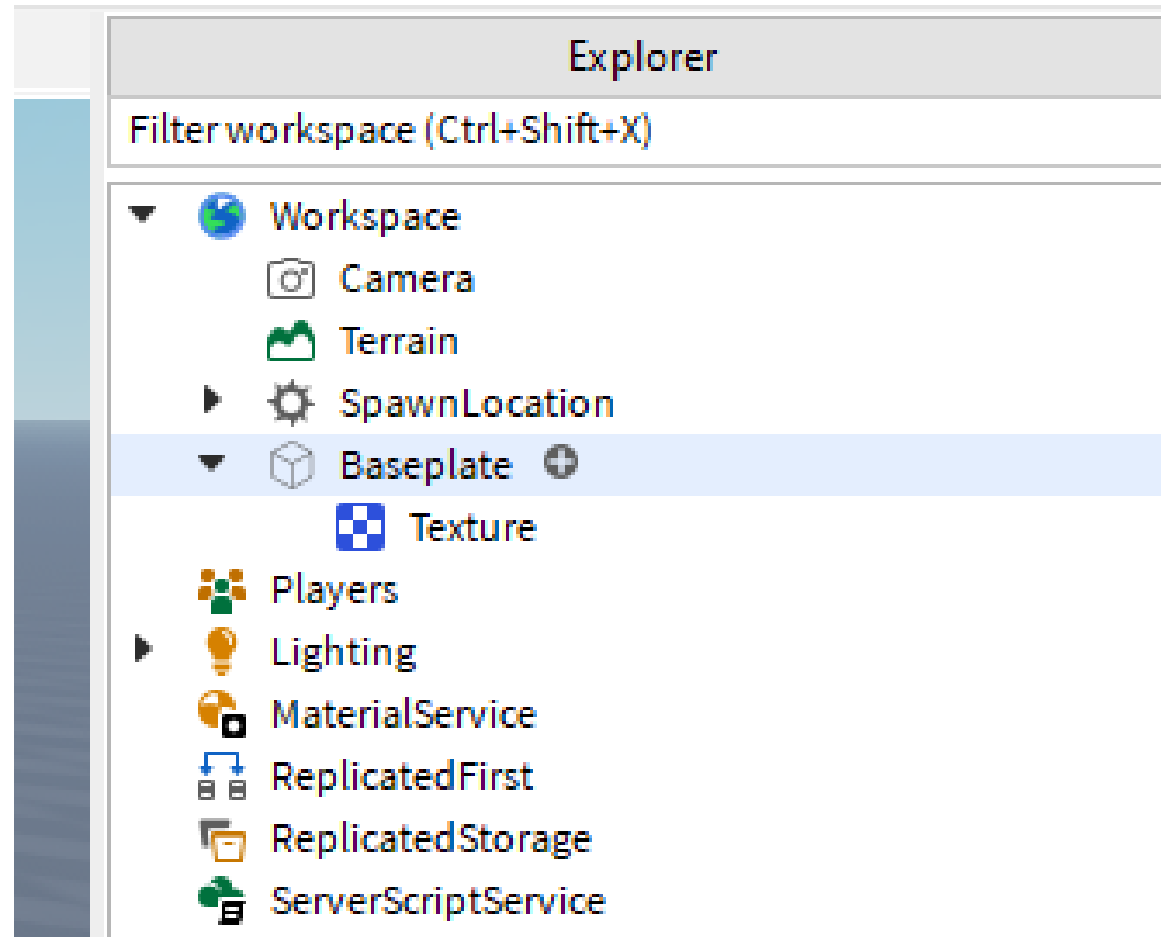


Roblox Studio

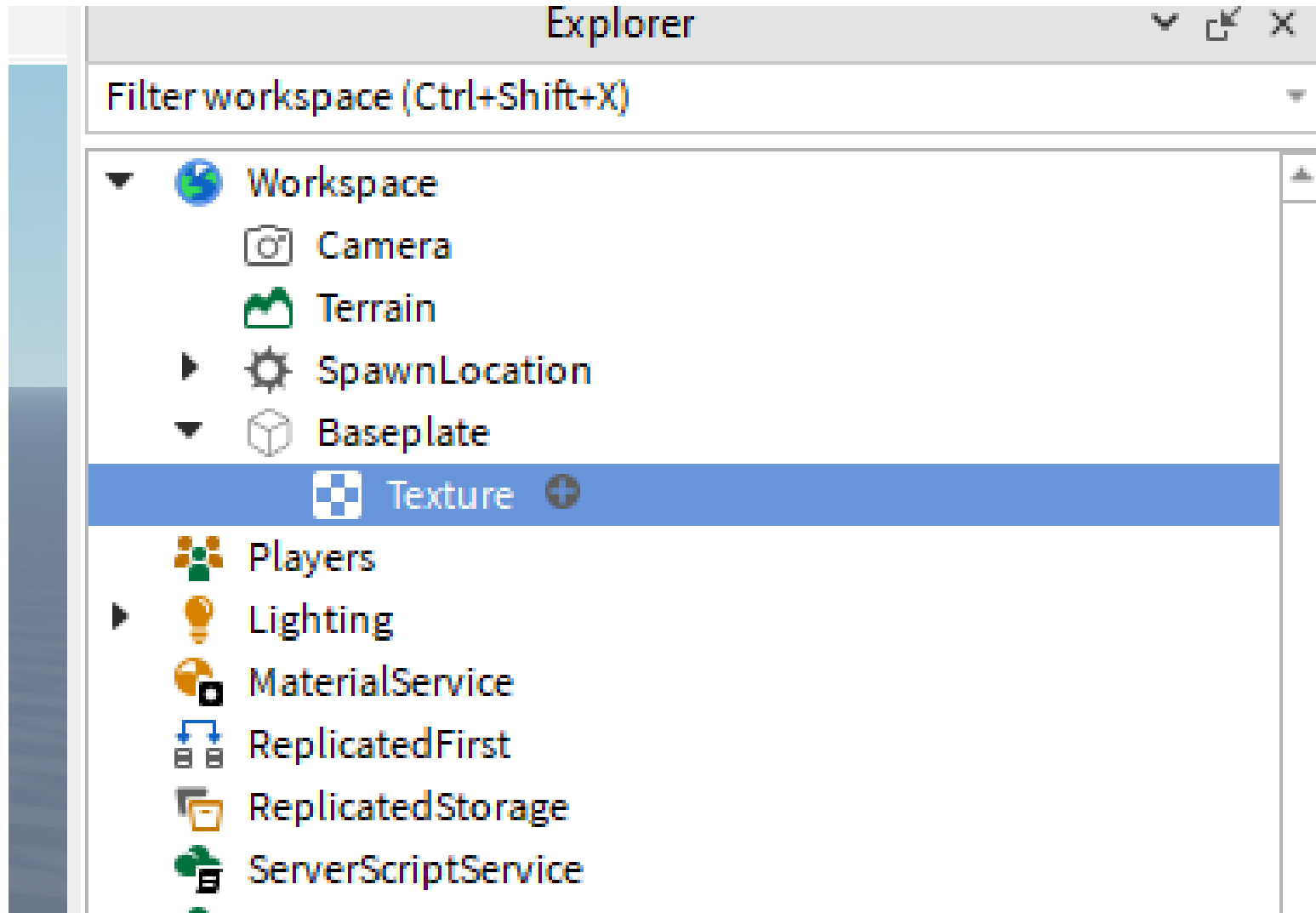
Use explorer
- Workspace



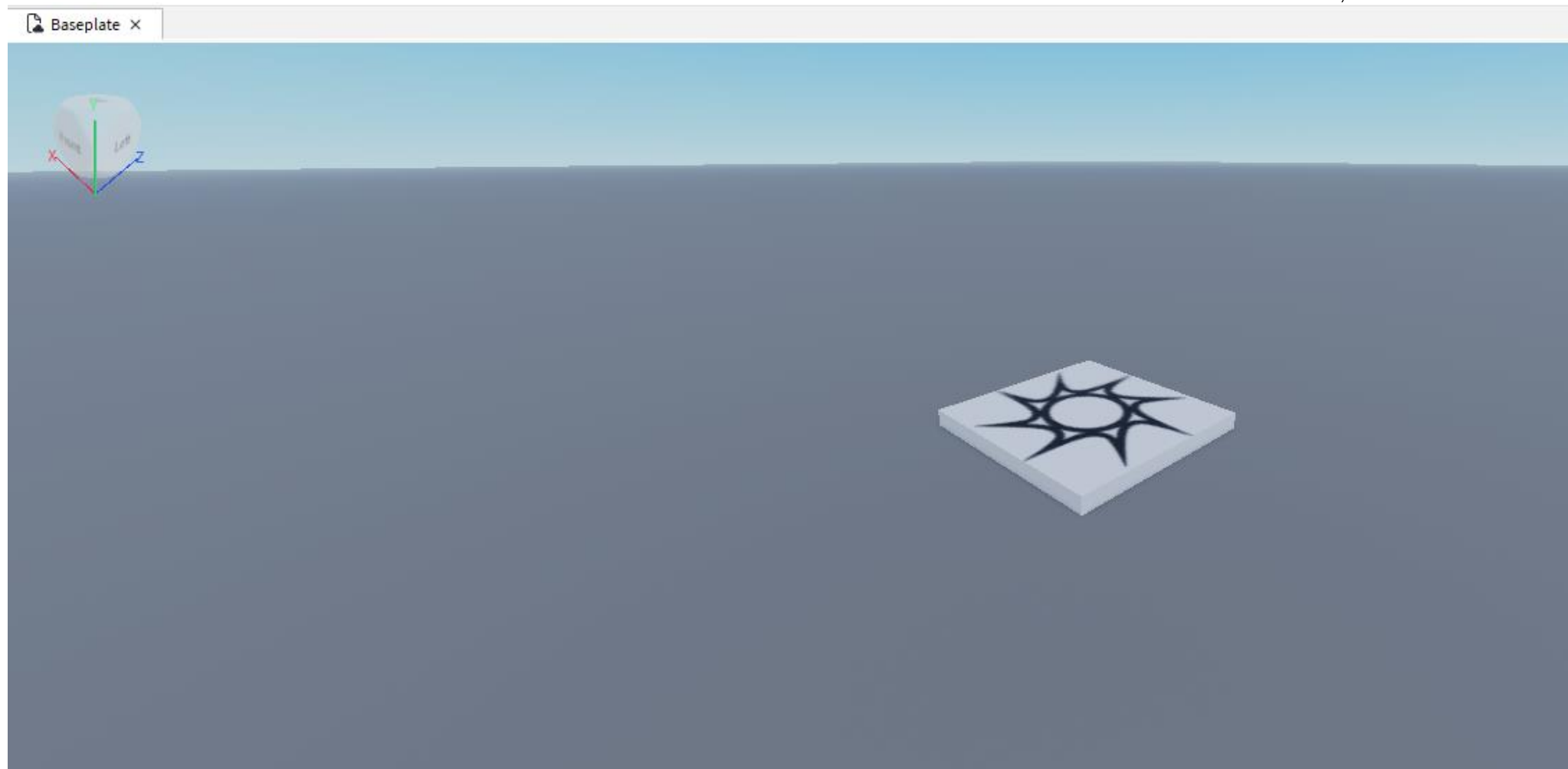
Roblox Studio - Open Baseplate object



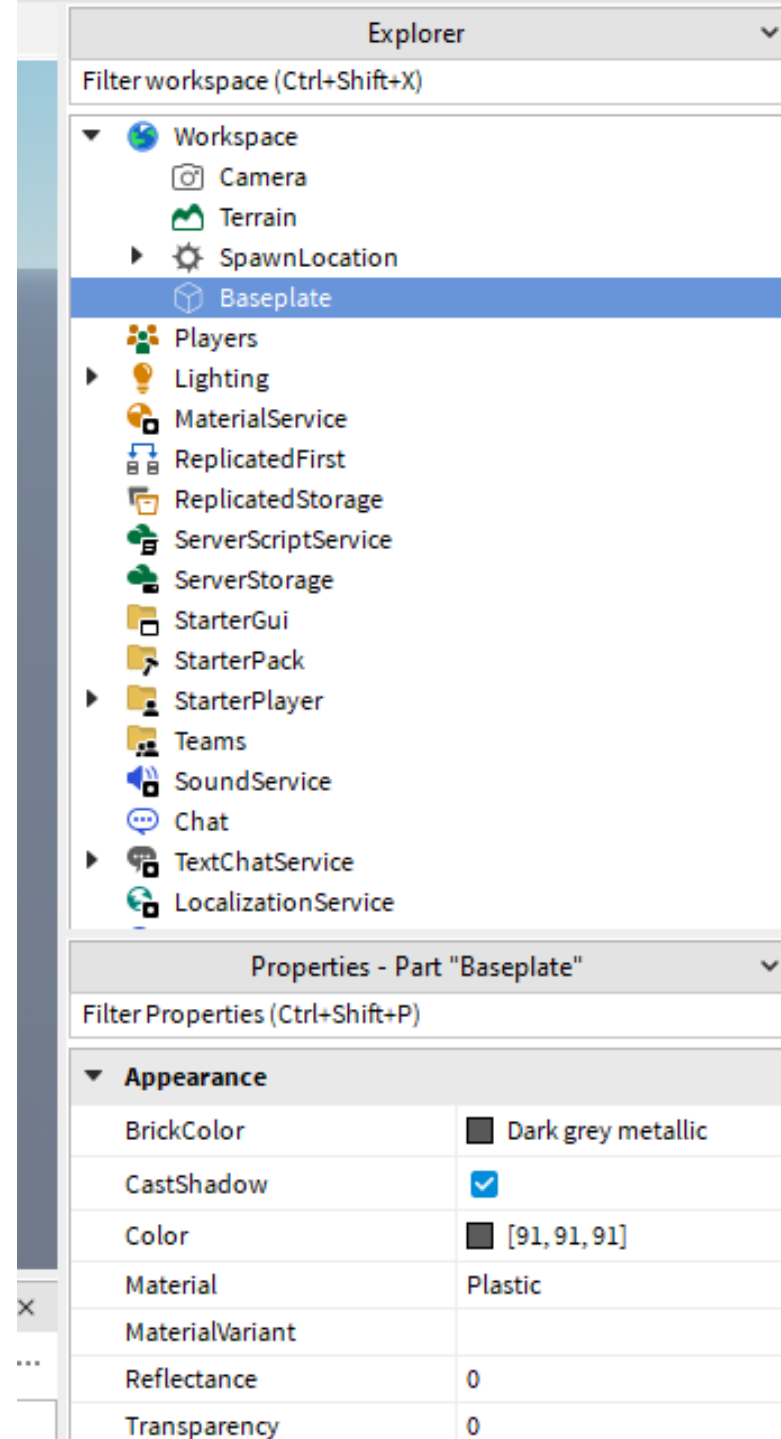
Roblox Studio - Select texture and delete it



Roblox Studio - Baseplate With no texture



Roblox Studio - Select Baseplate



The image shows a screenshot of the Roblox Studio interface. The top panel is the Explorer, which displays a tree view of the workspace. The 'Baseplate' is selected and highlighted in blue. Below the Explorer is the Properties panel, which shows the properties for the selected 'Part "Baseplate"'. The Properties panel is divided into sections, with 'Appearance' expanded to show various settings.

Explorer

Filter workspace (Ctrl+Shift+X)

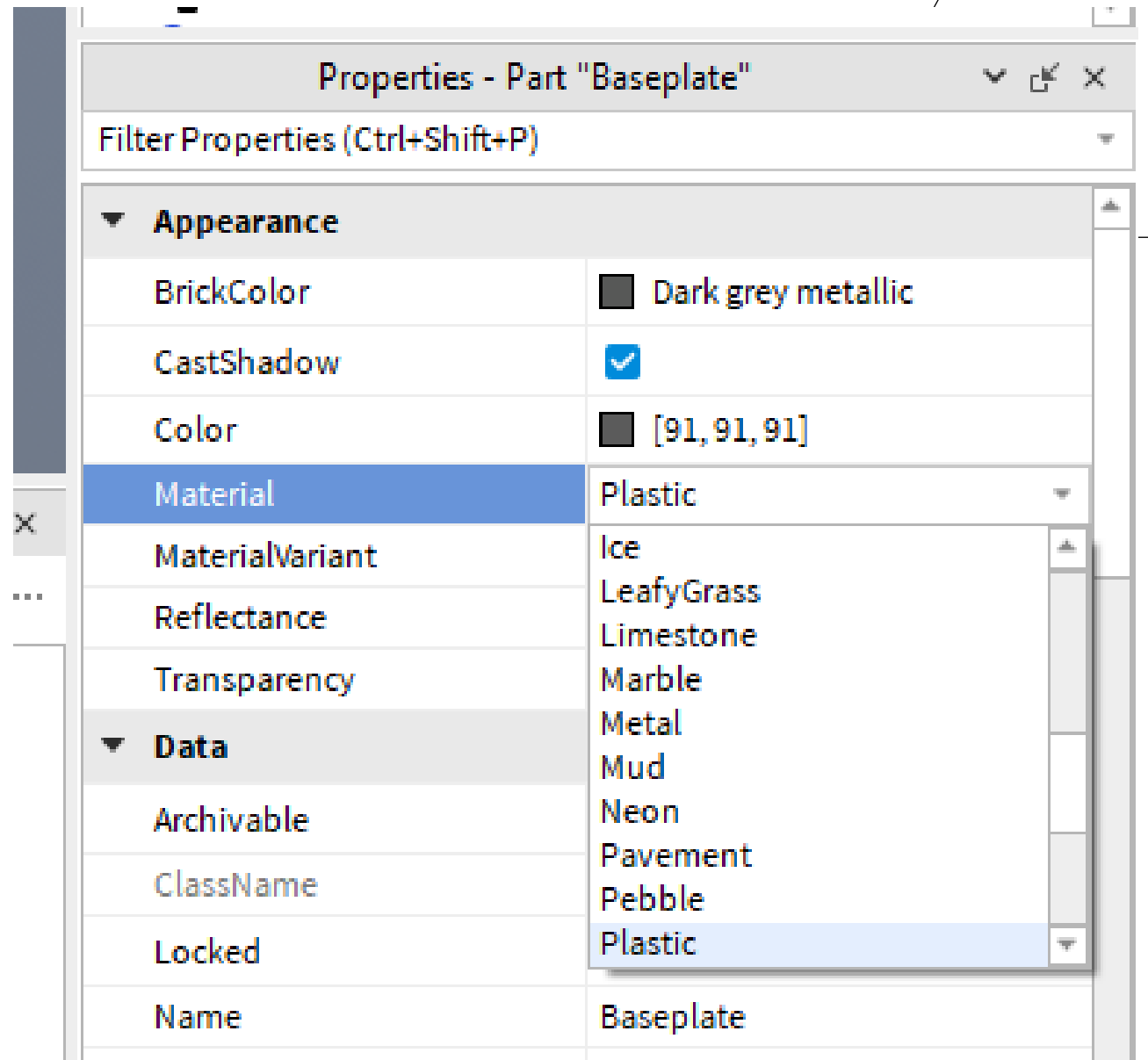
- Workspace
 - Camera
 - Terrain
 - SpawnLocation
 - Baseplate**
 - Players
- Lighting
- MaterialService
- ReplicatedFirst
- ReplicatedStorage
- ServerScriptService
- ServerStorage
- StarterGui
- StarterPack
- StarterPlayer
- Teams
- SoundService
- Chat
- TextChatService
- LocalizationService

Properties - Part "Baseplate"

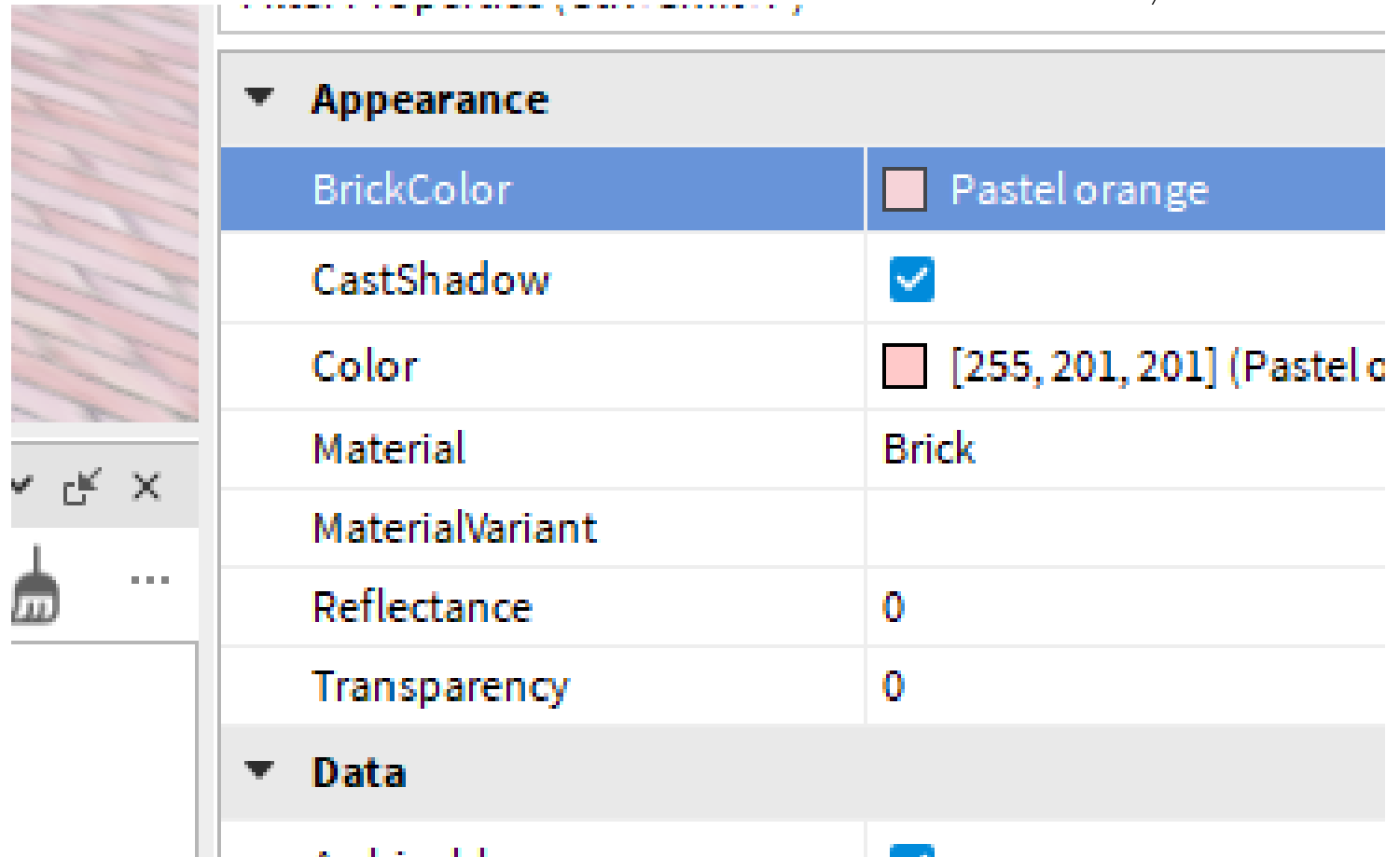
Filter Properties (Ctrl+Shift+P)

Appearance	
BrickColor	Dark grey metallic
CastShadow	<input checked="" type="checkbox"/>
Color	[91, 91, 91]
Material	Plastic
MaterialVariant	
Reflectance	0
Transparency	0

Roblox Studio - Baseplate material



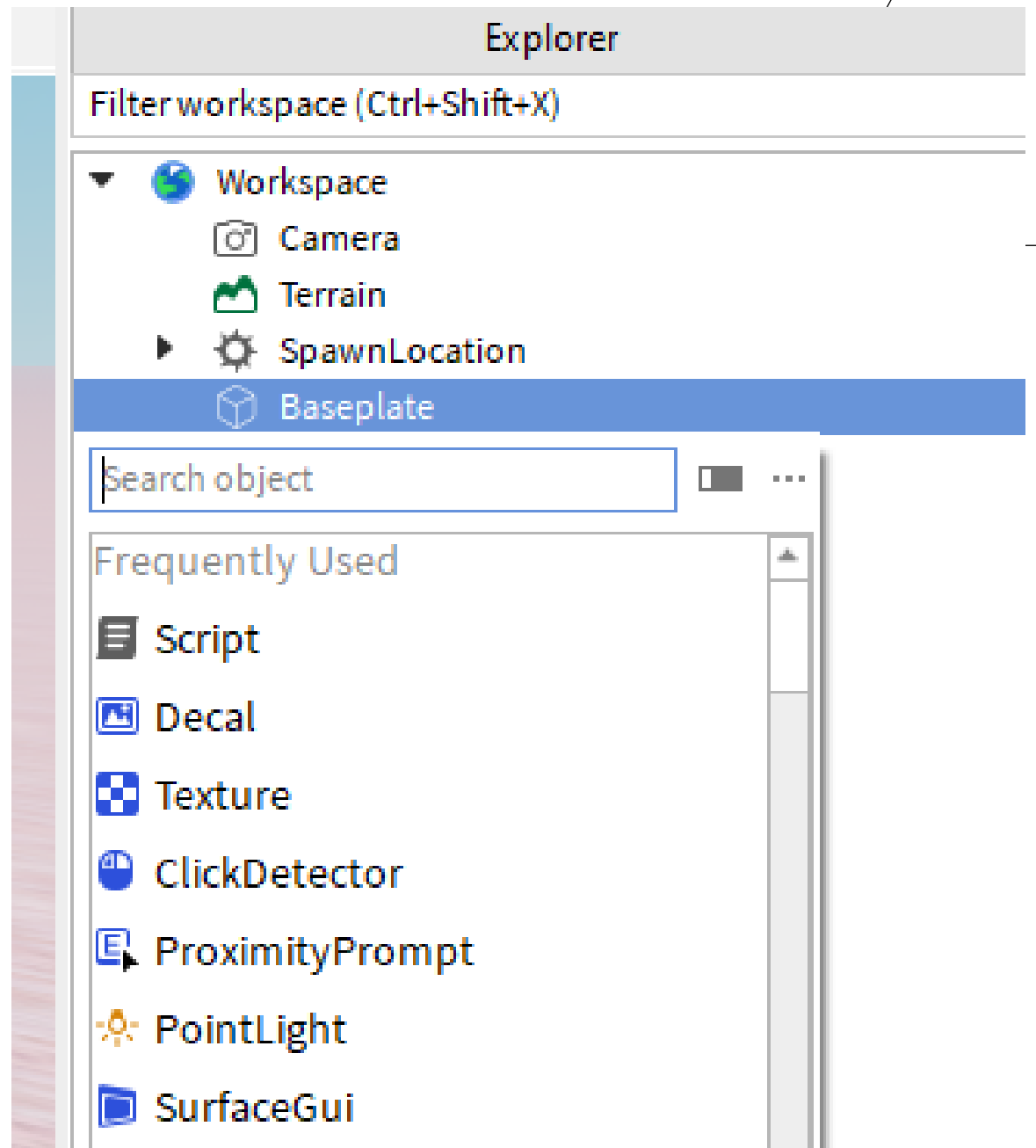
Roblox Studio - Change the material and color



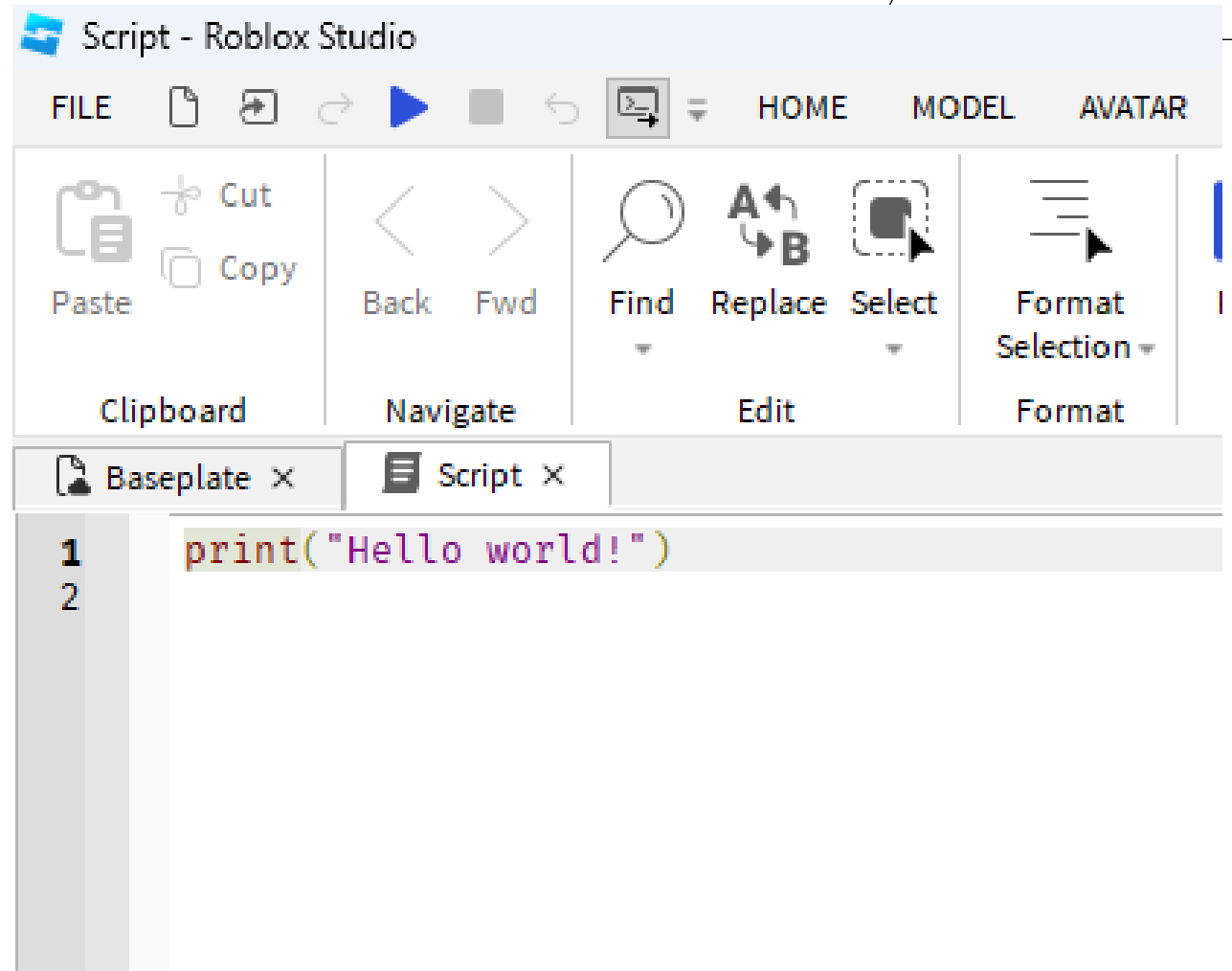
Roblox Studio - Adding a Script to the Baseplate

Click + next to
Baseplate

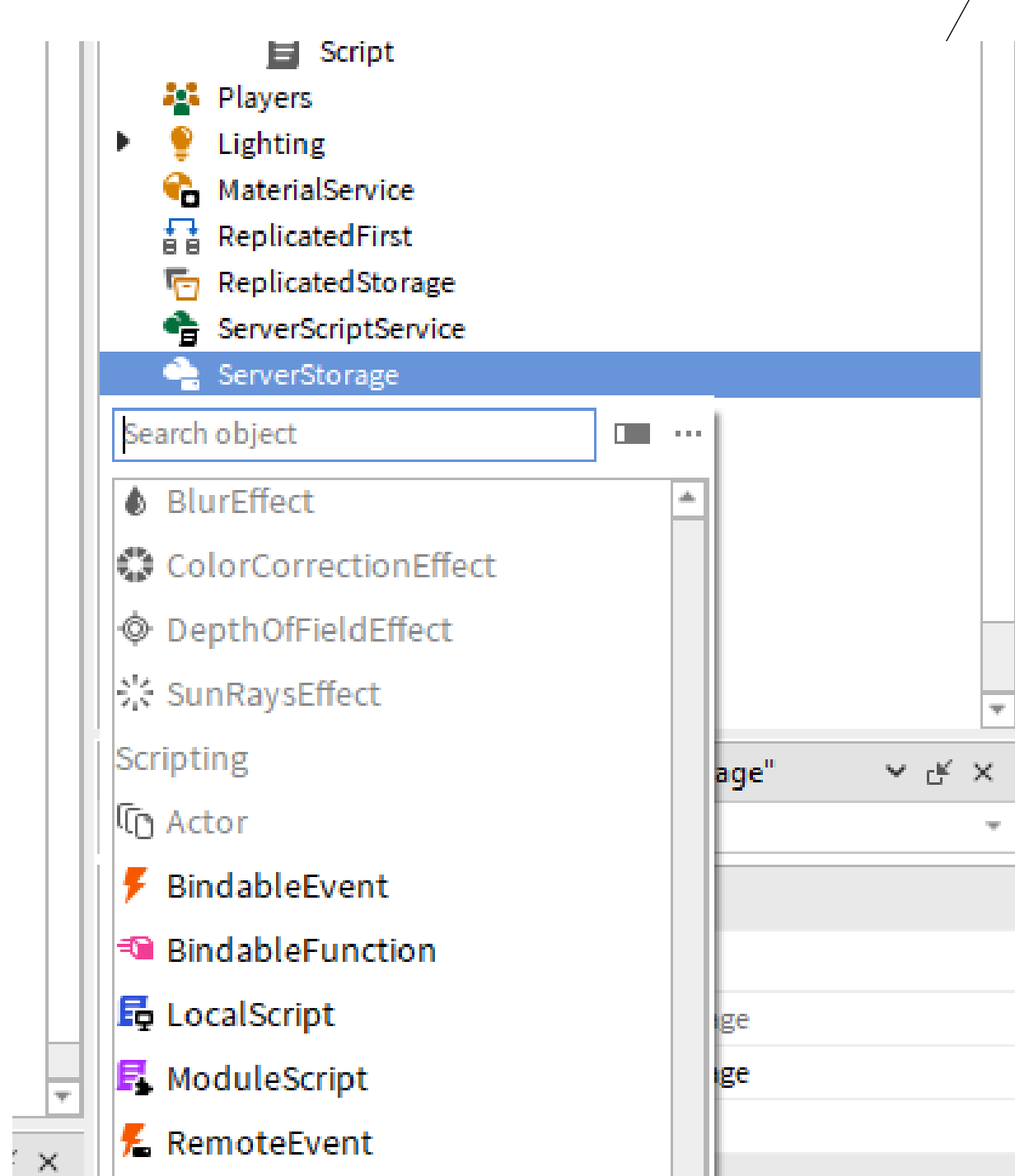
Click on "Script"



Roblox Studio -
"Hello World"
added automatically
to the script



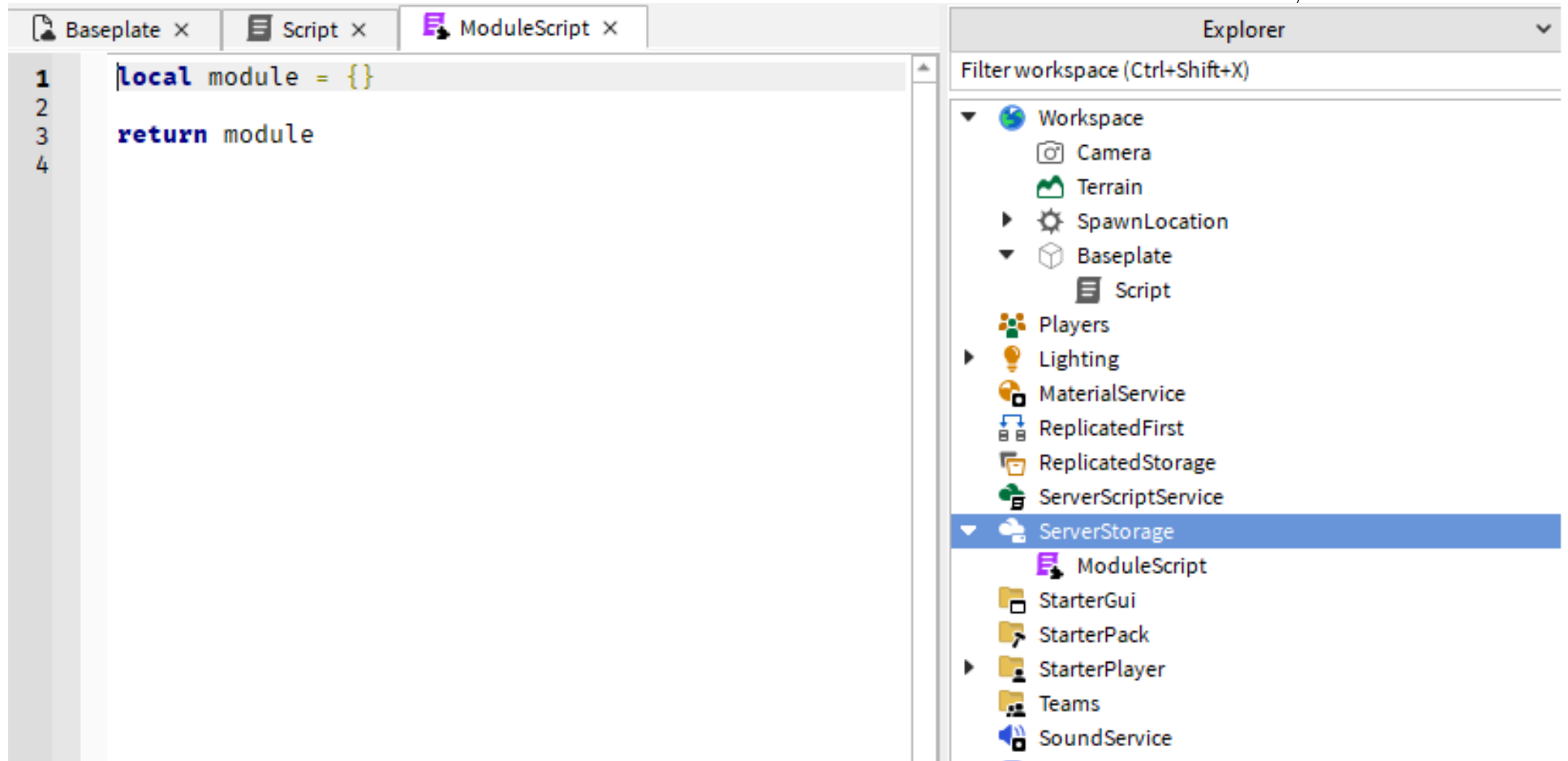
Roblox Studio -
Click the + sign
for ServerStorage
to Add a
ModuleScript



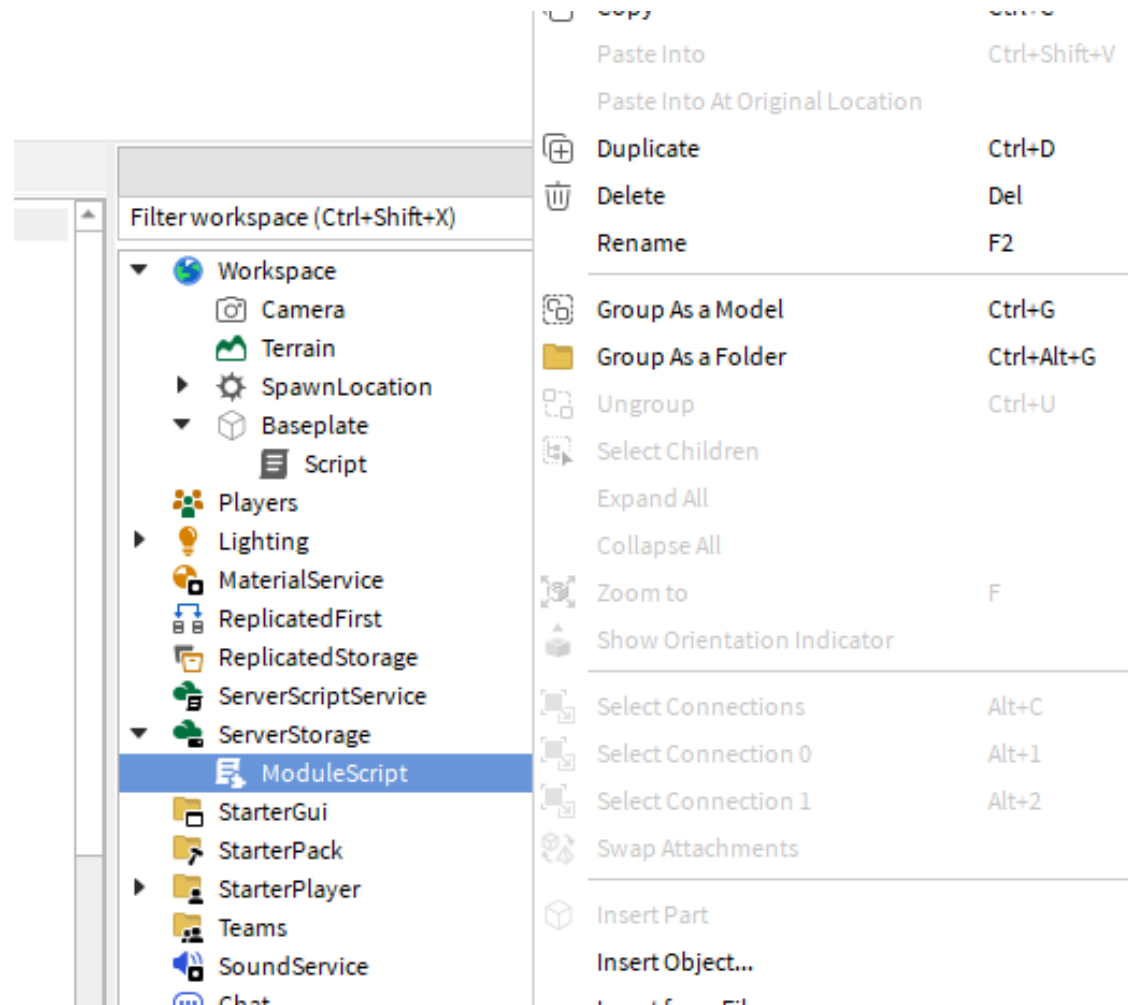
Roblox Studio -

Module with start with 2 lines of code

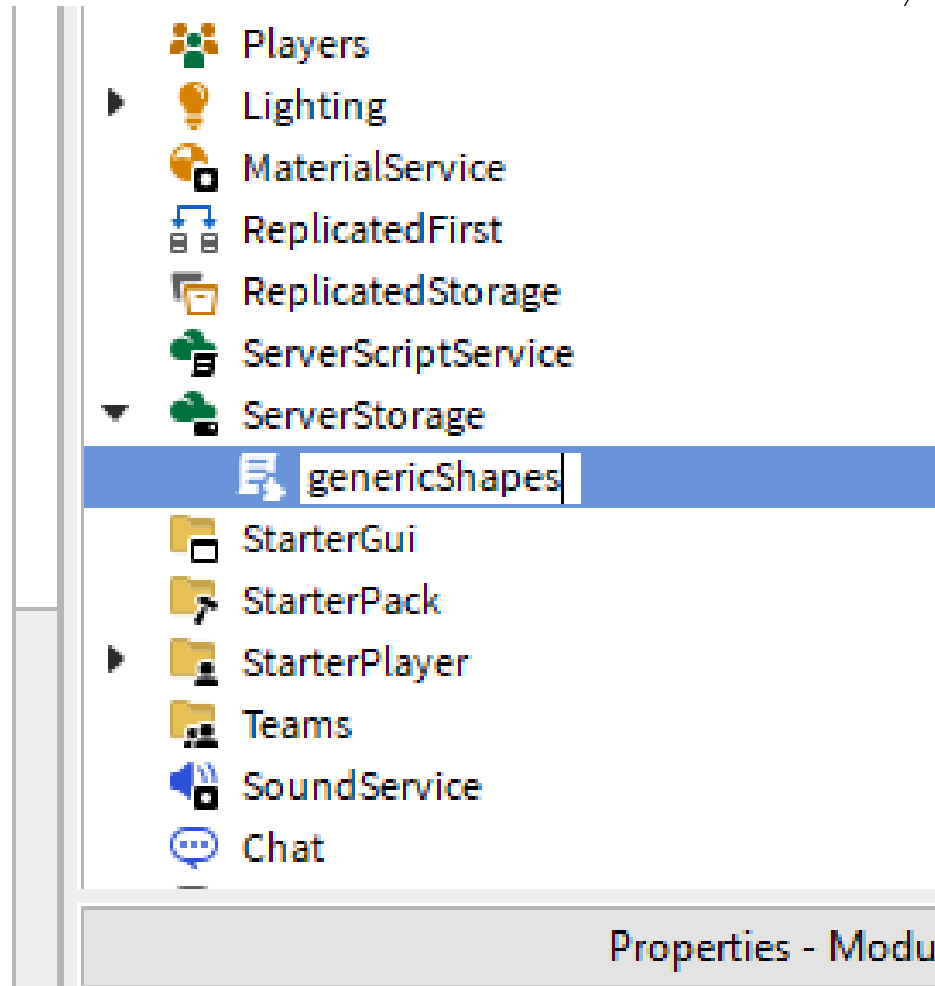
First line is an assignment of a "table" to a variable
Last line returns the "table"



Roblox Studio - Right-click on ModuleScript to rename it

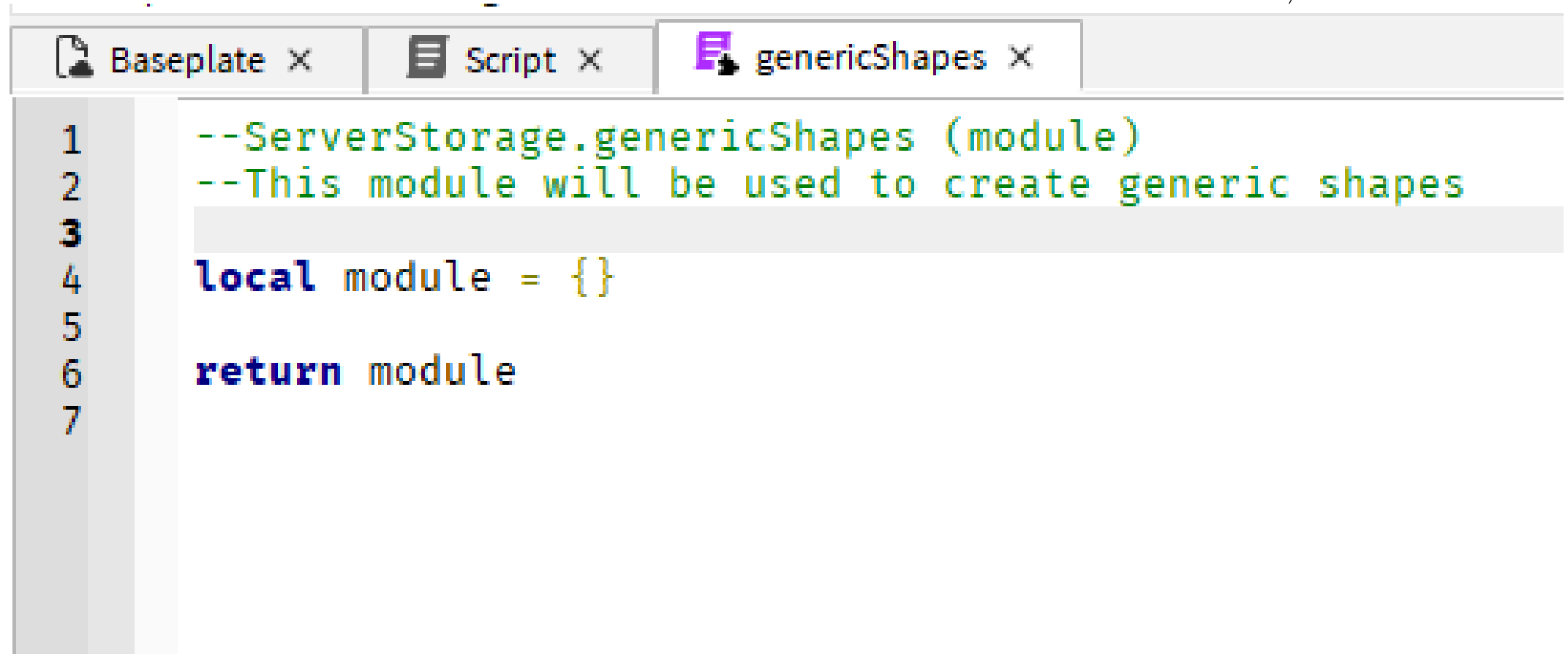


Roblox Studio - Rename it: genericShapes



Roblox Studio -

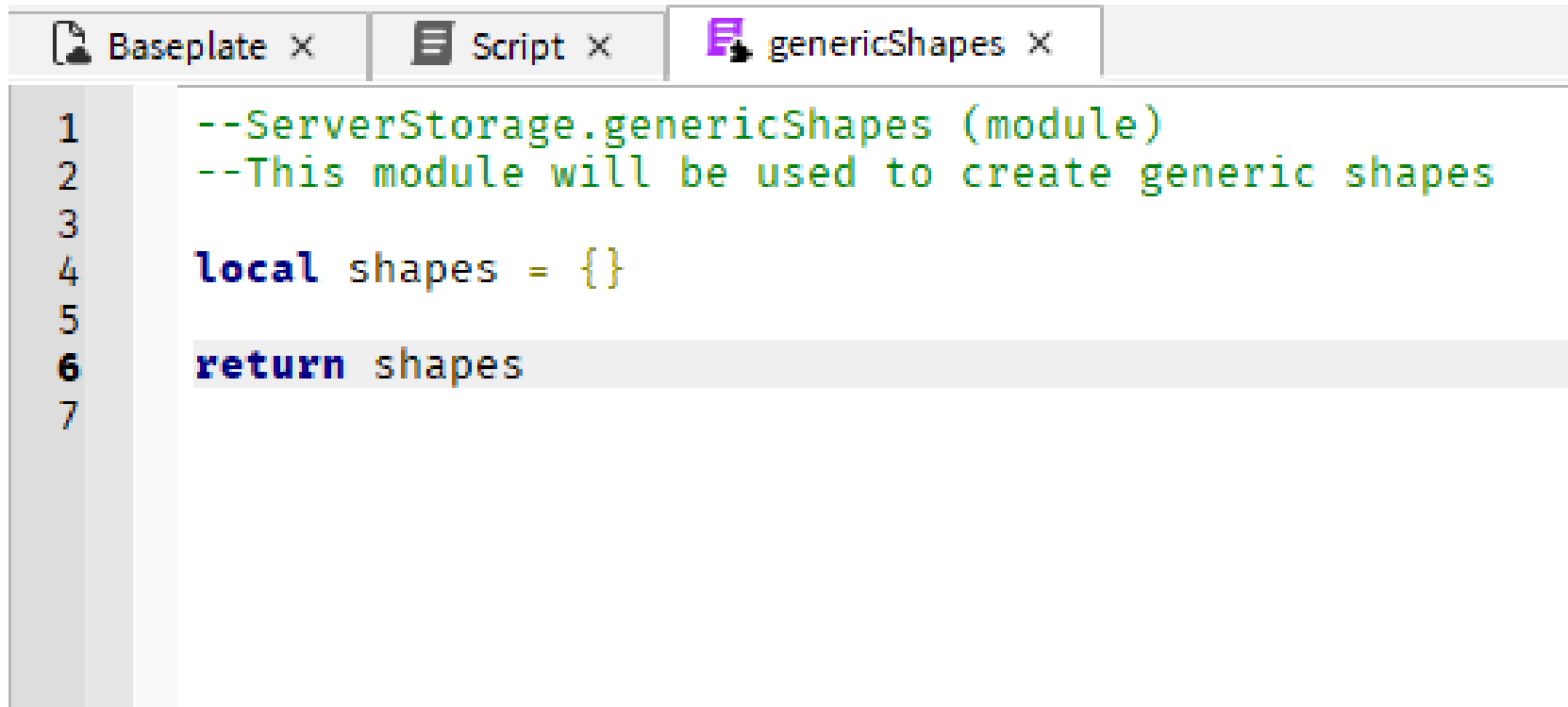
Add 2 comments to be beginning of the script



```
Baseplate x | Script x | genericShapes x  
1  --ServerStorage.genericShapes (module)  
2  --This module will be used to create generic shapes  
3  
4  local module = {}  
5  
6  return module  
7
```

Roblox Studio -

Change the class name from "module" to "shapes"



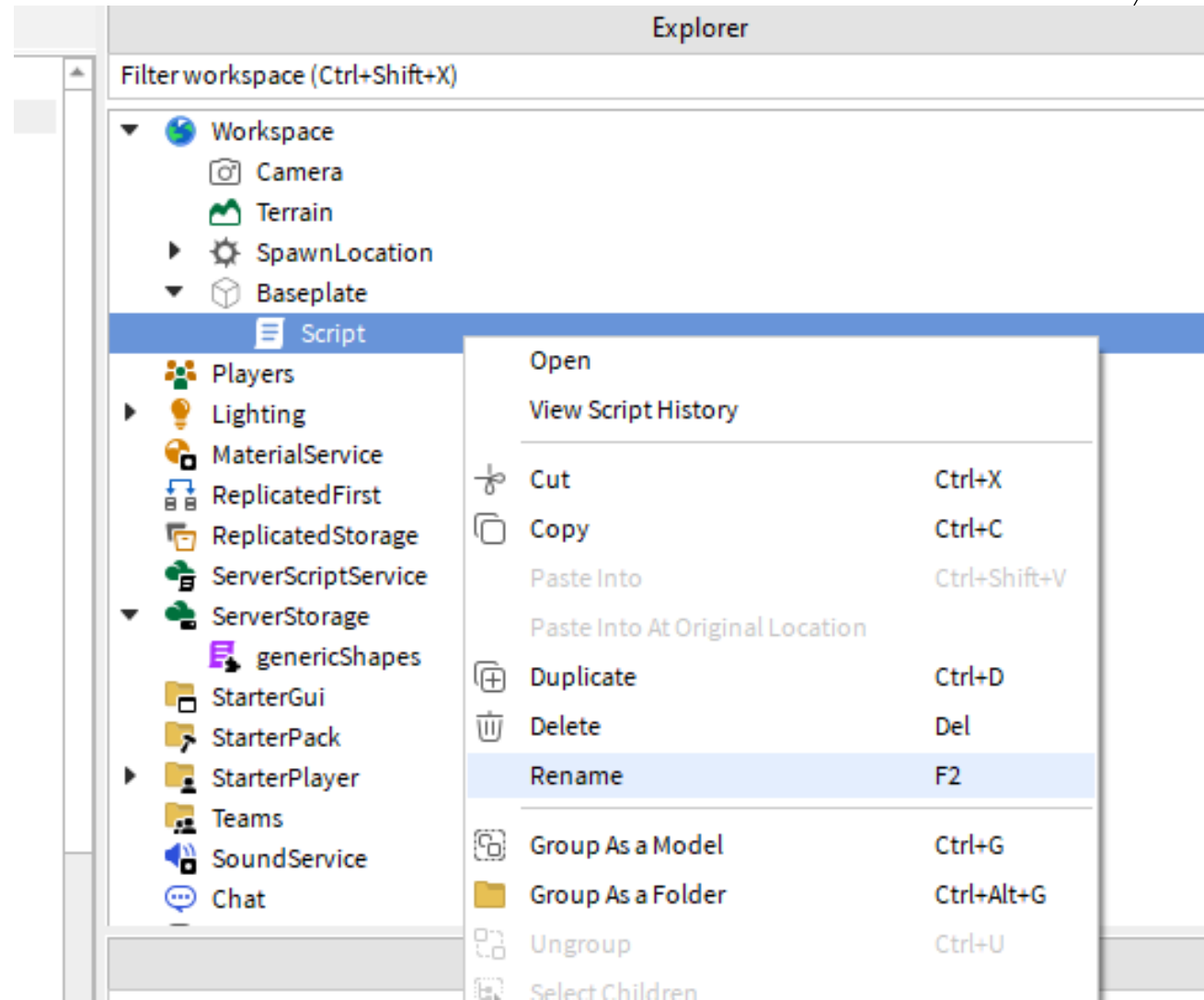
```
Baseplate x | Script x | genericShapes x  
1  --ServerStorage.genericShapes (module)  
2  --This module will be used to create generic shapes  
3  
4  local shapes = {}  
5  
6  return shapes  
7
```

Roblox Studio - Add a function in shapes called: helloWorld()

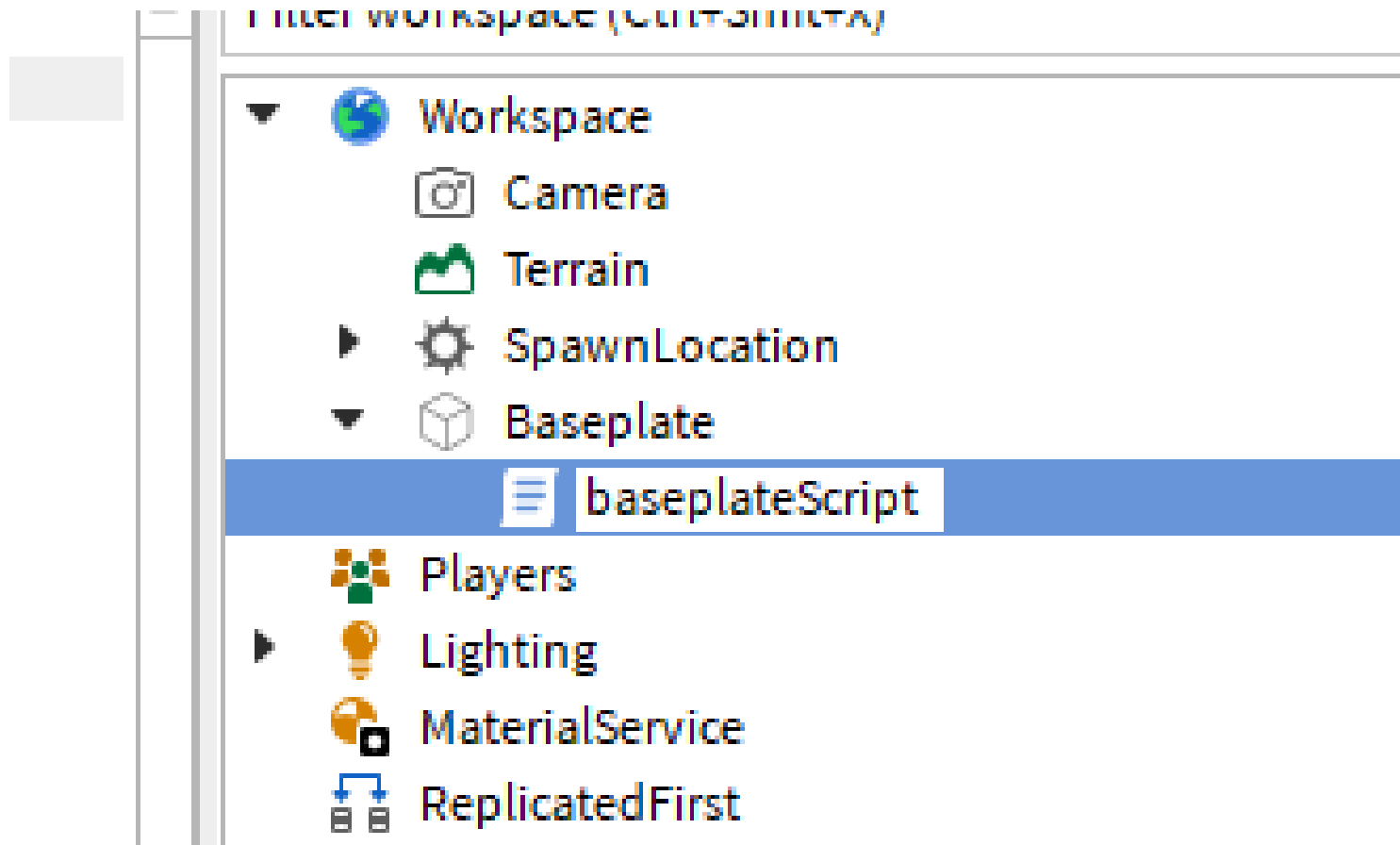
```
Baseplate x | Script x | genericShapes x |
1  --ServerStorage.genericShapes (module)
2  --This module will be used to create generic shapes
3
4  local shapes = {}
5
6  --This function returns a string value
7  function shapes.helloWorld()
8      return "Hello World"
9  end
10
11 return shapes
12
```

Roblox Studio - Rename the Baseplate Script

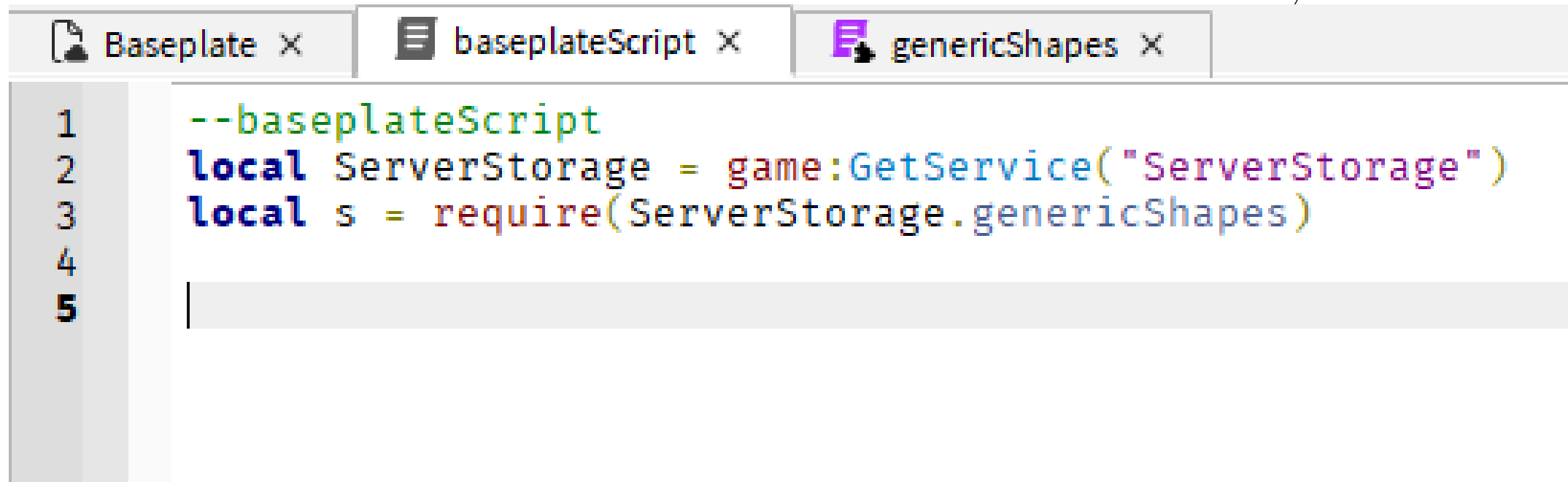
Right-click to access
the properties menu



Roblox Studio - Rename it: baseplateScript



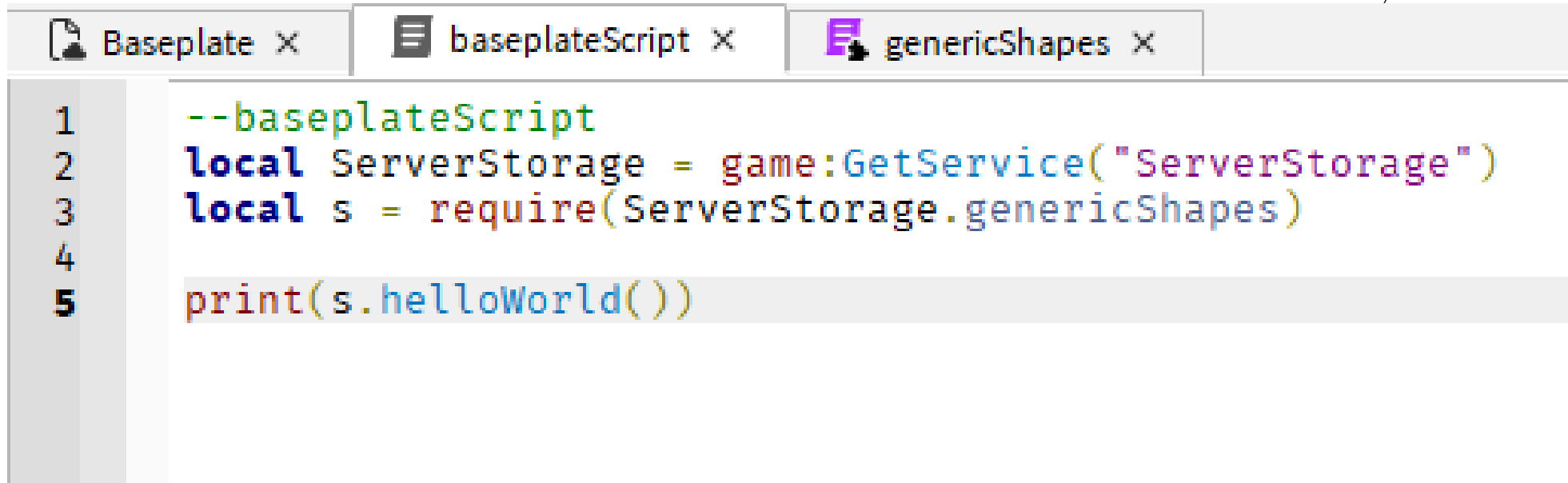
Roblox Studio - Add 2 lines of code



```
Baseplate x | baseplateScript x | genericShapes x |
1  --baseplateScript
2  local ServerStorage = game:GetService("ServerStorage")
3  local s = require(ServerStorage.genericShapes)
4
5
```

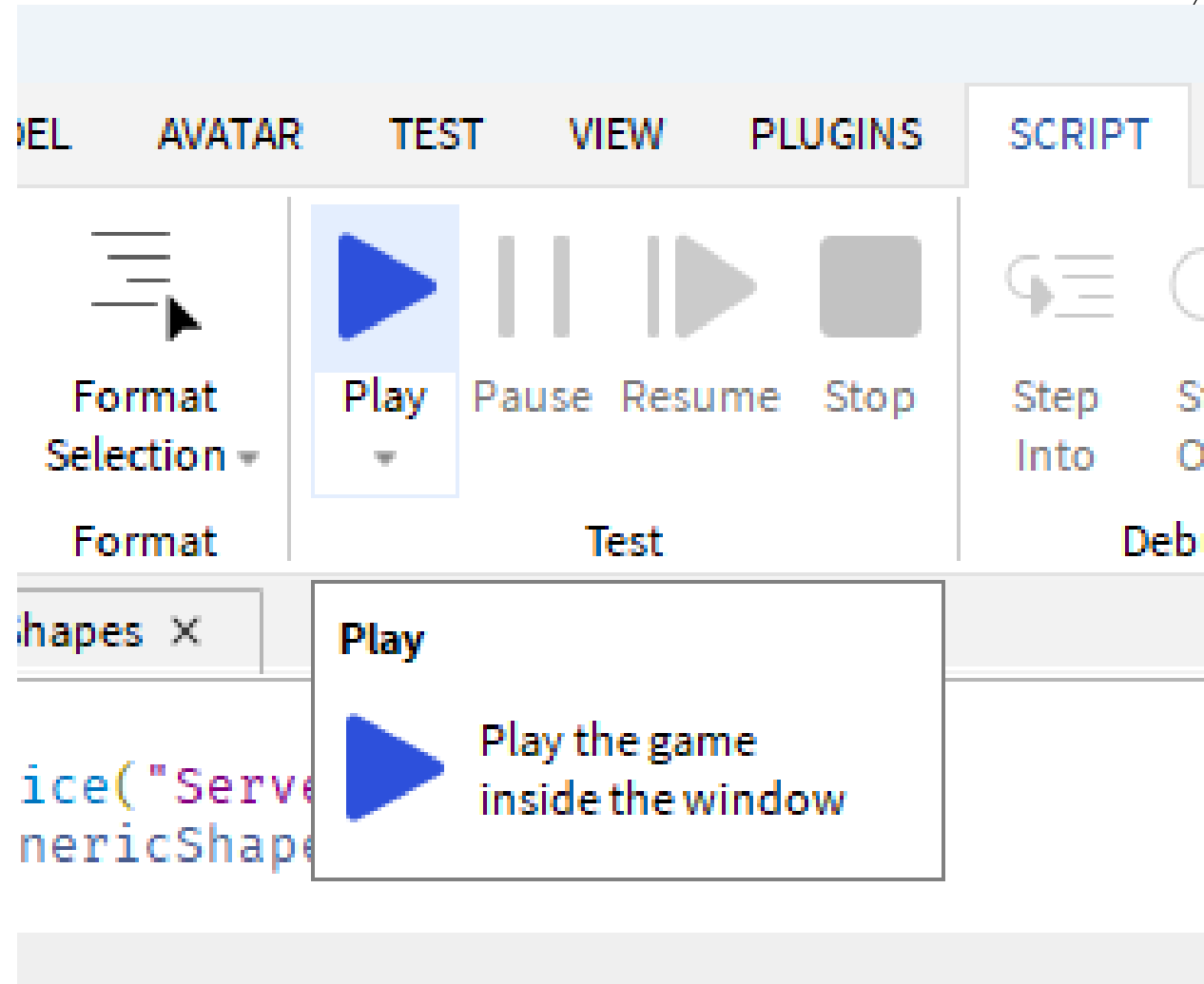
Roblox Studio -

Generic Shapes Module stored in s
Print the s.helloWorld() method.



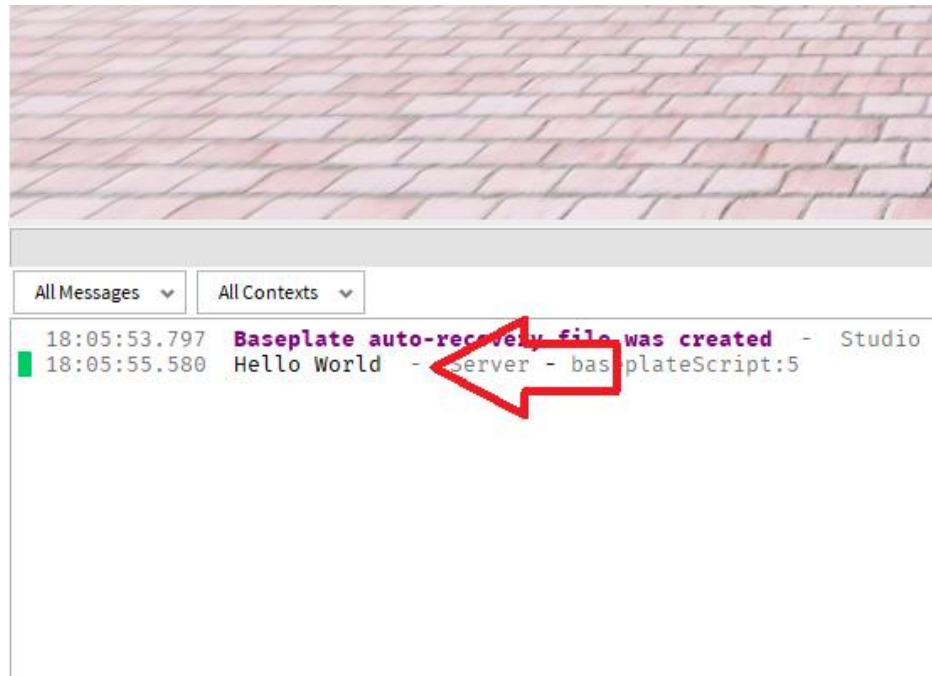
```
1  --baseplateScript
2  local ServerStorage = game:GetService("ServerStorage")
3  local s = require(ServerStorage.genericShapes)
4
5  print(s.helloWorld())
```

Roblox Studio -
Click on "Play"

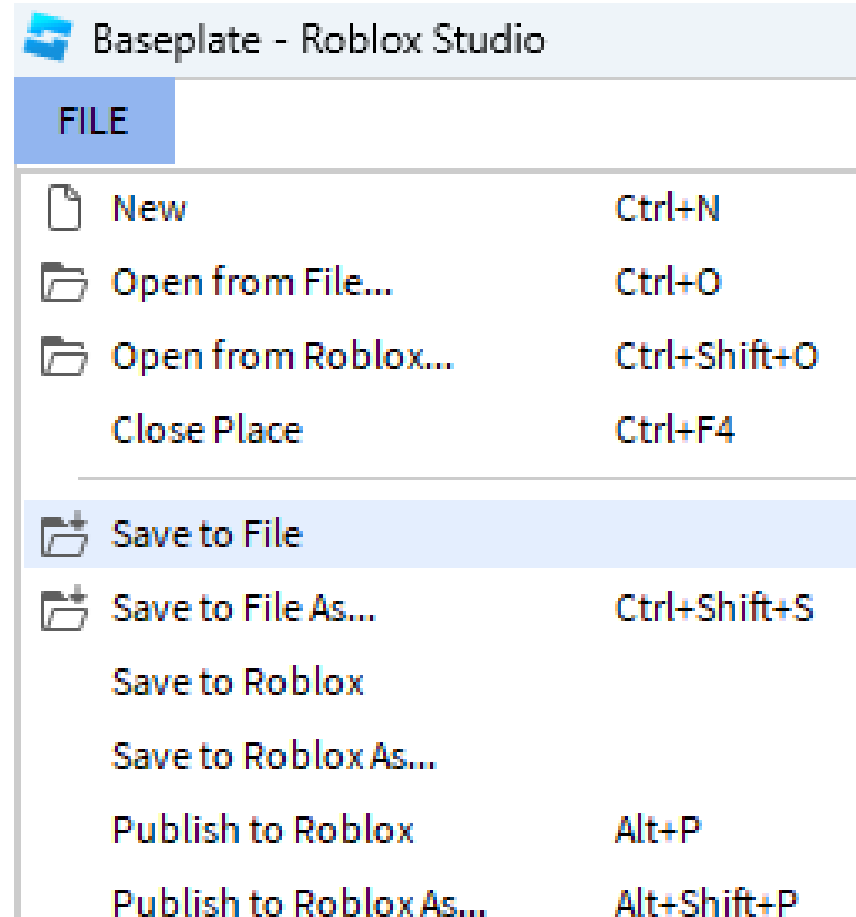


Roblox Studio - Output shows your message

Printed by the method contained in the shapes module



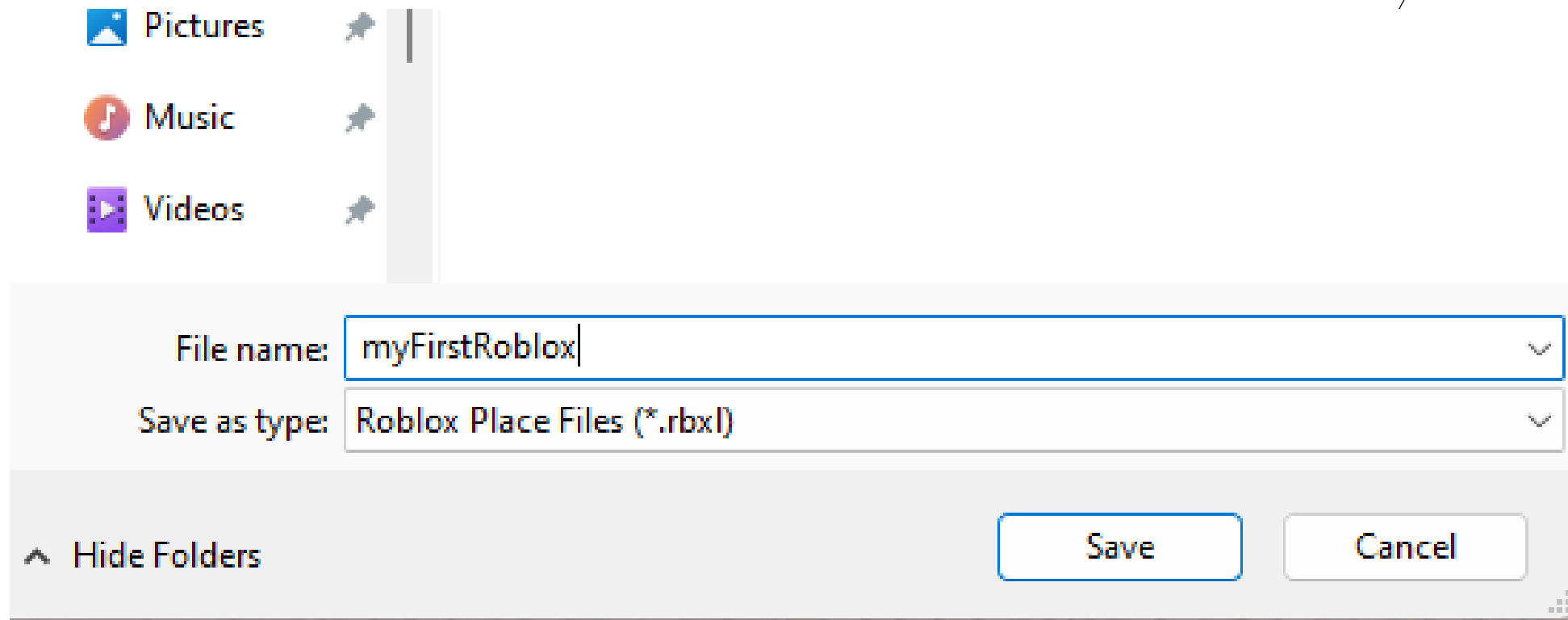
Roblox Studio - Save your program



Save to File

Give it a decent file name

Roblox Studio - I called mine: myFirstRoblox

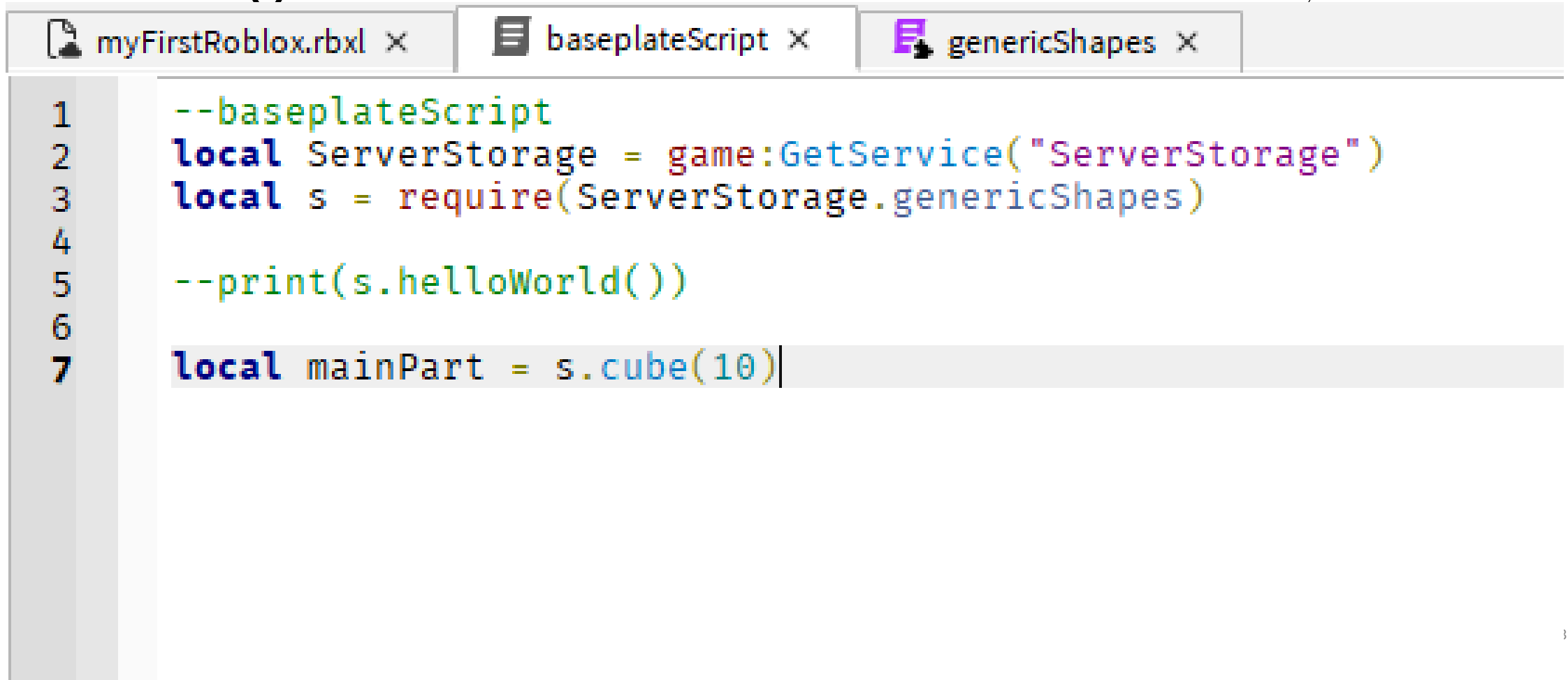


Roblox Studio -
Add the cube()
function to the
genericShapes module

```
1 --ServerStorage.genericShapes (module)
2 --This module will be used to create generic shapes
3
4 local shapes = {}
5
6 --This function returns a string value
7 ▼ function shapes.helloWorld()
8     return "Hello World"
9 end
10
11 --This function creates a cube
12 ▼ function shapes.cube(size)
13     local mainPart = Instance.new("Part")
14     mainPart.Anchored = true
15     mainPart.Shape = Enum.PartType.Block
16     mainPart.TopSurface = Enum.SurfaceType.Smooth
17     mainPart.BottomSurface = Enum.SurfaceType.Smooth
18     mainPart.Transparency = 0
19     mainPart.Reflectance = 0.6
20     mainPart.Parent = workspace
21     mainPart.Size = Vector3.new(size, size, size)
22     local zCoord = -15
23     local xCoord = 0
24     local yCoord = size/2
25     mainPart.Position += Vector3.new(xCoord, yCoord, zCoord)
26     return mainPart
27 end
28
29
30 return shapes
31
```


Roblox Studio -

Modify the baseplateScript to comment the print() and add the s.cube()



```
1  --baseplateScript
2  local ServerStorage = game:GetService("ServerStorage")
3  local s = require(ServerStorage.genericShapes)
4
5  --print(s.helloWorld())
6
7  local mainPart = s.cube(10)
```

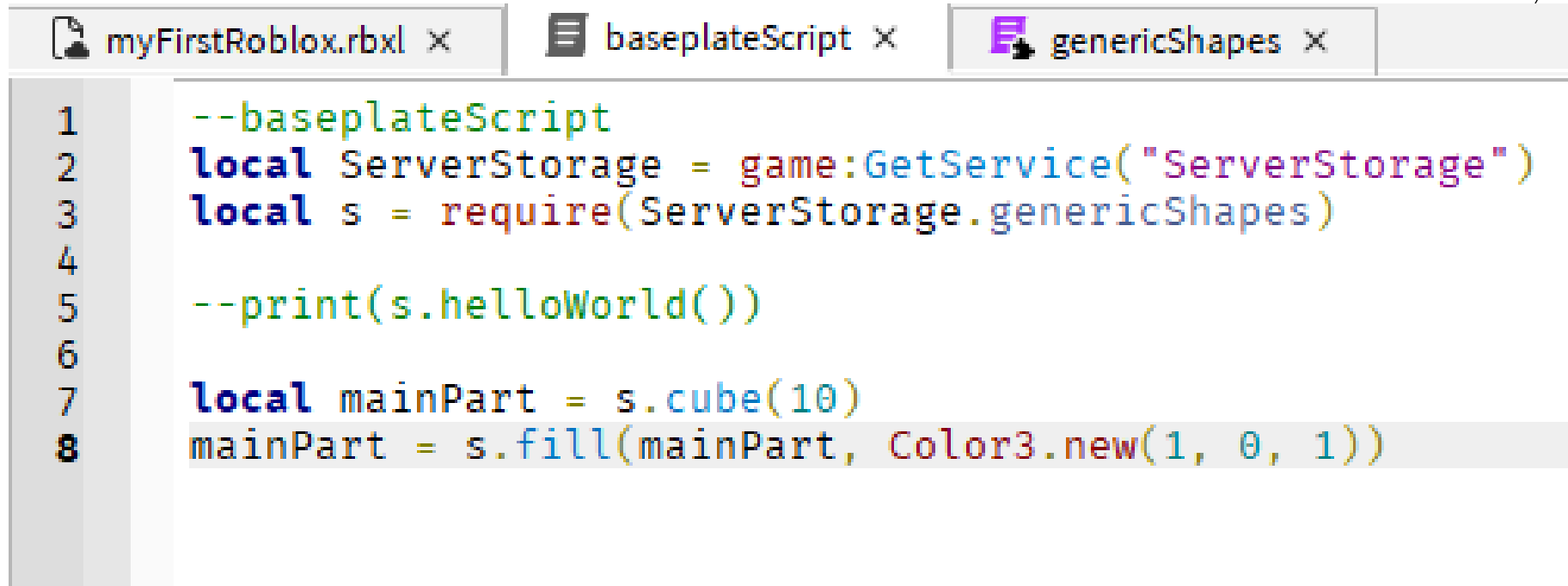
Roblox Studio - Result...



Roblox Studio - Add the .fill() method

```
28  
29 --This function changes the color of the object  
30 ▼ function shapes.fill(mainPart, colorValue)  
31     print(typeof(colorValue))  
32     mainPart.Color = colorValue  
33     return mainPart  
34 end  
35  
36  
37
```

Roblox Studio -
Edit the baseplateScript to use the new method.



```
1  --baseplateScript
2  local ServerStorage = game:GetService("ServerStorage")
3  local s = require(ServerStorage.genericShapes)
4
5  --print(s.helloWorld())
6
7  local mainPart = s.cube(10)
8  mainPart = s.fill(mainPart, Color3.new(1, 0, 1))
```

Roblox Studio

Run it...

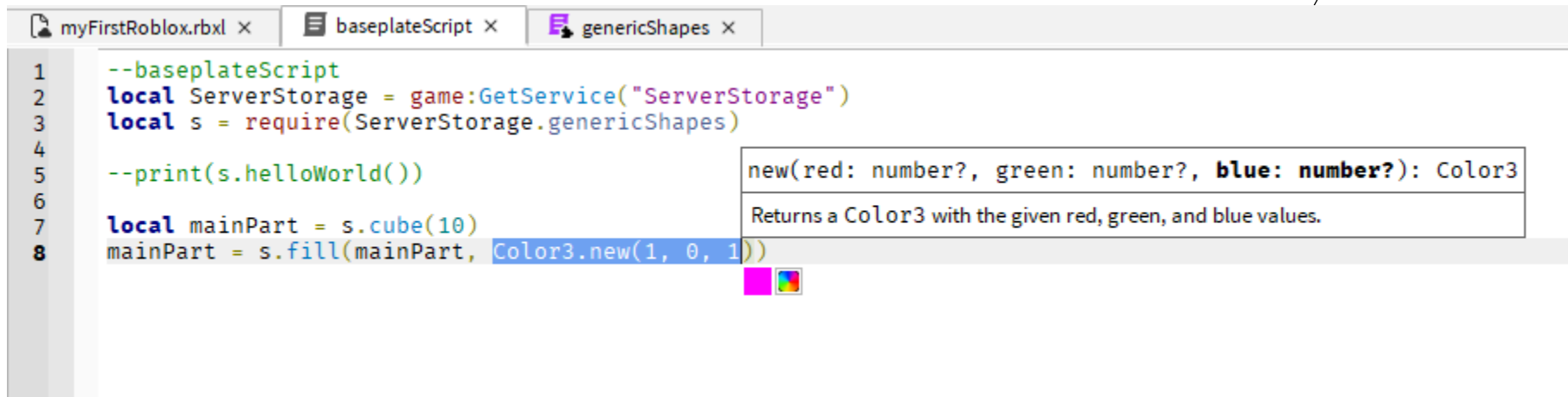


Roblox Studio - Modify the fill method to evaluate typeof(colorValue)

```
28
29 --This function changes the color of the object
30 ▼ function shapes.fill(mainPart, colorValue)
31 ▼ if typeof(colorValue) == "Color3" then
32     mainPart.Color = colorValue
33 ▼ else
34 ▼     if typeof(colorValue) == "string" then
35         local red = 1
36         local green = 0
37         local blue = 0
38         mainPart.Color = Color3.new(red, green, blue)
39 ▼     else
40         print("Error in the Fill method. You need to pass a Color3 vector or a string")
41     end
42 end
43 return mainPart
44 end
45
46
```

Roblox Studio -

If you block the `Color3.new()` you will get a color picker popup



The screenshot shows the Roblox Studio script editor with three tabs: 'myFirstRoblox.rbxl', 'baseplateScript', and 'genericShapes'. The script in the 'baseplateScript' tab is as follows:

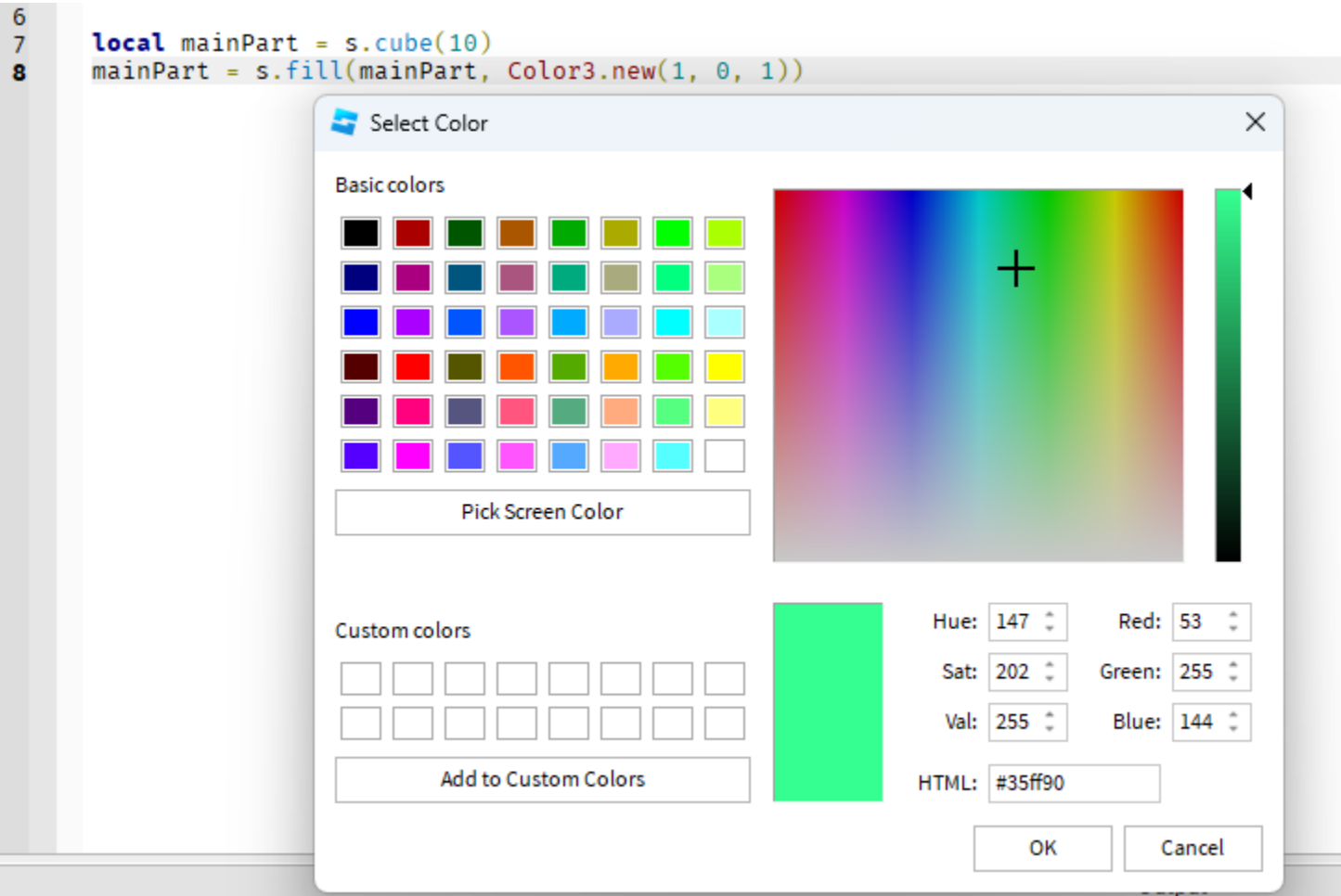
```
1 --baseplateScript
2 local ServerStorage = game:GetService("ServerStorage")
3 local s = require(ServerStorage.genericShapes)
4
5 --print(s.helloWorld())
6
7 local mainPart = s.cube(10)
8 mainPart = s.fill(mainPart, Color3.new(1, 0, 1))
```

A tooltip is displayed over the `Color3.new(1, 0, 1)` call on line 8. The tooltip contains the following text:

```
new(red: number?, green: number?, blue: number?): Color3
Returns a Color3 with the given red, green, and blue values.
```

Below the tooltip, there is a small color picker icon consisting of a pink square and a multi-colored square.

Roblox Studio - You can pick colors from this dialog



Roblox Studio -
Add the colorName table
to the top of the
genericShapes module

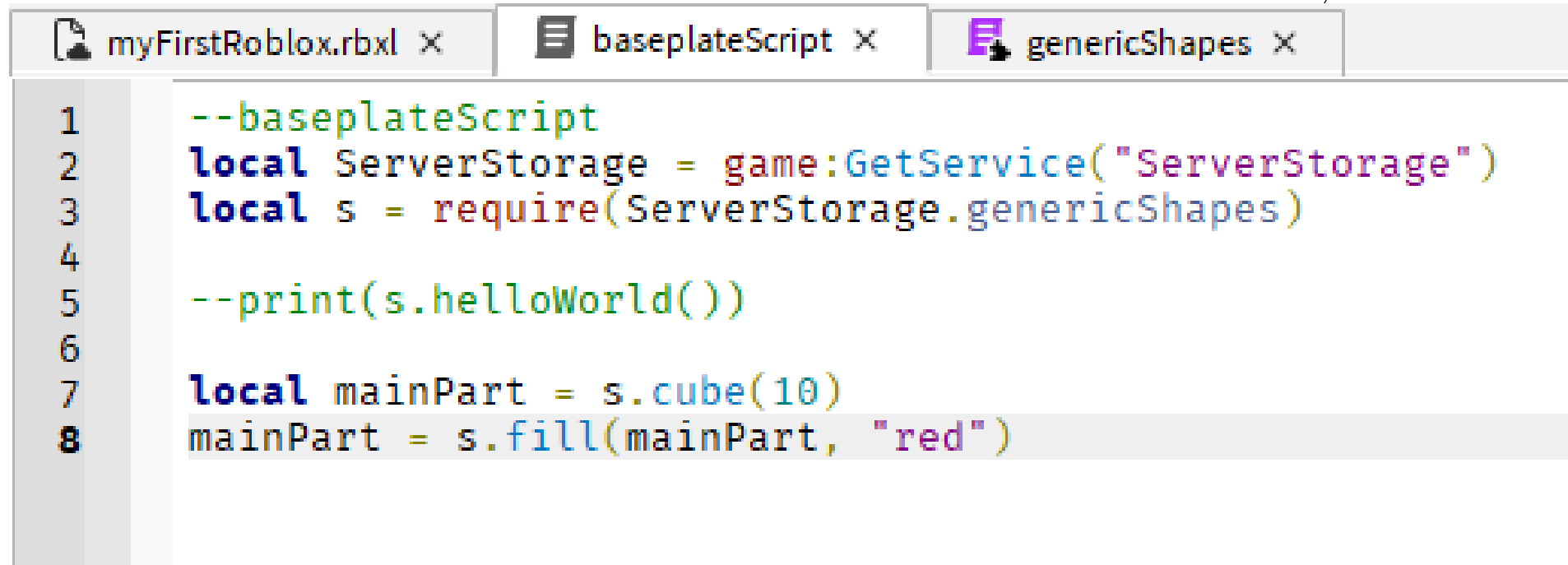
```
myFirstRoblox.rbxl x baseplateScript x genericShapes x
1 --ServerStorage.genericShapes (module)
2 --This module will be used to create generic shapes
3
4 local shapes = {}
5
6 local colorName={
7     ["pink"]={["RGB"]="FFC0CB", ["DEC"]={255,192,203}},
8     ["lightpink"]={["RGB"]="FFB6C1", ["DEC"]={255,182,193}},
9     ["hotpink"]={["RGB"]="FF69B4", ["DEC"]={255,105,180}},
10    ["deeppink"]={["RGB"]="FF1493", ["DEC"]={255,20,147}},
11    ["palevioletred"]={["RGB"]="DB7093", ["DEC"]={219,112,147}},
12    ["mediumvioletred"]={["RGB"]="C71585", ["DEC"]={199,21,133}},
13    ["lightsalmon"]={["RGB"]="FFA07A", ["DEC"]={255,160,122}},
14    ["salmon"]={["RGB"]="FA8072", ["DEC"]={250,128,114}},
15    ["darksalmon"]={["RGB"]="E9967A", ["DEC"]={233,150,122}},
16    ["lightcoral"]={["RGB"]="F08080", ["DEC"]={240,128,128}},
17    ["indianred"]={["RGB"]="CD5C5C", ["DEC"]={205,92,92}},
18    ["crimson"]={["RGB"]="DC143C", ["DEC"]={220,20,60}},
19    ["firebrick"]={["RGB"]="B22222", ["DEC"]={178,34,34}},
20    ["darkred"]={["RGB"]="8B0000", ["DEC"]={139,0,0}},
21    ["red"]={["RGB"]="FF0000", ["DEC"]={255,0,0}},
22    ["orangered"]={["RGB"]="FF4500", ["DEC"]={255,69,0}},
23    ["tomato"]={["RGB"]="FF6347", ["DEC"]={255,99,71}},
24    ["coral"]={["RGB"]="FF7F50", ["DEC"]={255,127,80}},
25    ["darkorange"]={["RGB"]="FF8C00", ["DEC"]={255,140,0}},
26    ["orange"]={["RGB"]="FFA500", ["DEC"]={255,165,0}},
27    ["yellow"]={["RGB"]="FFFF00", ["DEC"]={255,255,0}},
28    ["lightyellow"]={["RGB"]="FFFFE0", ["DEC"]={255,255,224}},
29    ["lemonchiffon"]={["RGB"]="FFFACD", ["DEC"]={255,250,205}},
30    ["lightgoldenrodyellow"]={["RGB"]="FAFAD2", ["DEC"]={250,250,210}},
31    ["papayawhip"]={["RGB"]="FFEFD5", ["DEC"]={255,239,213}},
32    ["moccasin"]={["RGB"]="FFE4B5", ["DEC"]={255,228,181}},
33    ["peachpuff"]={["RGB"]="FFDAB9", ["DEC"]={255,218,185}},
34    ["palegoldenrod"]={["RGB"]="EEE8AA", ["DEC"]={238,232,170}},
35    ["khaki"]={["RGB"]="F0E68C", ["DEC"]={240,230,140}},
36    ["darkkhaki"]={["RGB"]="8DD760", ["DEC"]={140,210,107}}
```

Roblox Studio - Modify the .fill() method.

```
171
172 --This function changes the color of the object
173 ▼ function shapes.fill(mainPart, colorValue)
174 ▼     if typeof(colorValue) == "Color3" then
175         mainPart.Color = colorValue
176 ▼     else
177 ▼         if typeof(colorValue) == "string" then
178             local red = colorName[colorValue]["DEC"][1]/255    --rescale from 0-255 to 0-1
179             local green = colorName[colorValue]["DEC"][2]/255  --rescale from 0-255 to 0-1
180             local blue = colorName[colorValue]["DEC"][3]/255  --rescale from 0-255 to 0-1
181 | mainPart.Color = Color3.new(red, green, blue)
182 ▼     else
183         print("Error in the Fill method.  You need to pass a Color3 vector or a string")
184     end
185 end
186 return mainPart
187 end
188
189
```

Roblox Studio -

Modify the baseplateScript to use a string for the colorName.



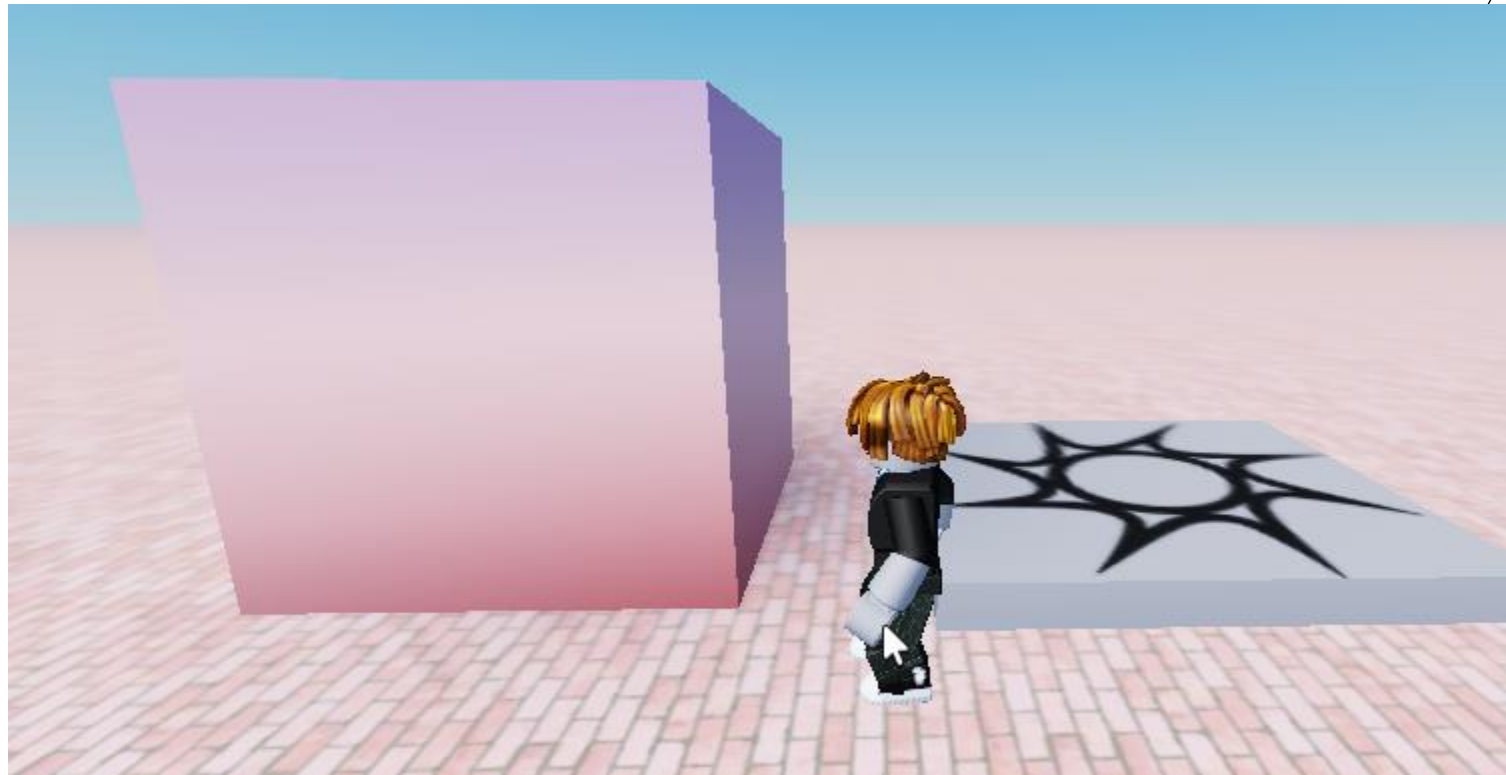
The screenshot shows the Roblox Studio interface with three tabs: 'myFirstRoblox.rbxl', 'baseplateScript', and 'genericShapes'. The 'baseplateScript' tab is active, displaying the following Lua code:

```
1  --baseplateScript
2  local ServerStorage = game:GetService("ServerStorage")
3  local s = require(ServerStorage.genericShapes)
4
5  --print(s.helloWorld())
6
7  local mainPart = s.cube(10)
8  mainPart = s.fill(mainPart, "red")
```

The line `mainPart = s.fill(mainPart, "red")` on line 8 is highlighted in grey, indicating the modification to use a string for the color name.

Roblox Studio

Run it...



Roblox Studio

Trying another color...



The screenshot shows the Roblox Studio interface with three open files: 'myFirstRoblox.rbxl', 'baseplateScript', and 'genericShapes'. The 'genericShapes' file is active, displaying a Lua script. The script contains the following code:

```
1  --baseplateScript
2  local ServerStorage = game:GetService("ServerStorage")
3  local s = require(ServerStorage.genericShapes)
4
5  --print(s.helloWorld())
6
7  local mainPart = s.cube(10)
8  mainPart = s.fill(mainPart, "darkslateblue")
```

The line `mainPart = s.fill(mainPart, "darkslateblue")` on line 8 is highlighted in a light gray background, indicating it is the current line of execution or the focus of the user's action.

Roblox Studio

Run it...



Roblox Studio - Add the method for .reflectance()

```
188
189 --This function changes the reflectance of the object
190 ▼ function shapes.reflectance(mainPart, value)
191 ▼   if typeof(value) == "number" then
192     mainPart.Reflectance = value
193 ▼   else
194     print("Error in the Reflectance method. You need to pass a number. You passed a: "..typeof(value))
195   end
196   return mainPart
197 end
198
199
200
```

Roblox Studio - Change the reflectance of the mainPart

```
myFirstRoblox.rbxl x  baseplateScript x  genericShapes x  
1  --baseplateScript  
2  local ServerStorage = game:GetService("ServerStorage")  
3  local s = require(ServerStorage.genericShapes)  
4  
5  --print(s.helloWorld())  
6  
7  local mainPart = s.cube(10)  
8  mainPart = s.fill(mainPart, "darkslateblue")  
9  s.reflectance(mainPart, 0)
```


Roblox Studio

Run it...



Roblox Studio -

Add rotation and transparency methods to the genericShapes module.

```
198
199 --This function changes the rotation of the object
200 ▼ function shapes.rotate(mainPart, rotateValue)
201 ▼   if typeof(rotateValue) == "Vector3" then
202     mainPart.Orientation = rotateValue
203 ▼   else
204     print("Error in the Rotation method. You need to pass a Vector3. You passed a: "..typeof(rotateValue))
205   end
206   return mainPart
207 end
208
209 --This function changes the transparency of the object
210 ▼ function shapes.transparency(mainPart, value)
211 ▼   if typeof(value) == "number" then
212     mainPart.Transparency = value
213 ▼   else
214     print("Error in the Transparency method. You need to pass a number. You passed a: "..typeof(value))
215   end
216   return mainPart
217 end
218
```

Roblox Studio - Edit the baseplateScript to use these methods.

Note:

- Line 8 is an assignment.
- Lines 9-11 are member dot methods.

```
myFirstRoblox.rbxl x  baseplateScript x  genericShapes x
1  --baseplateScript
2  local ServerStorage = game:GetService("ServerStorage")
3  local s = require(ServerStorage.genericShapes)
4
5  --print(s.helloWorld())
6
7  local mainPart = s.cube(10)
8  mainPart = s.fill(mainPart, "darkslateblue")
9  s.reflectance(mainPart, 0)
10 s.rotate(mainPart, Vector3.new(45, 45, 0))
11 s.transparency(mainPart, 0.4)
12
```

Roblox Studio

Run it...



Roblox Studio -
Add a materialNames array to the
genericShapes module.

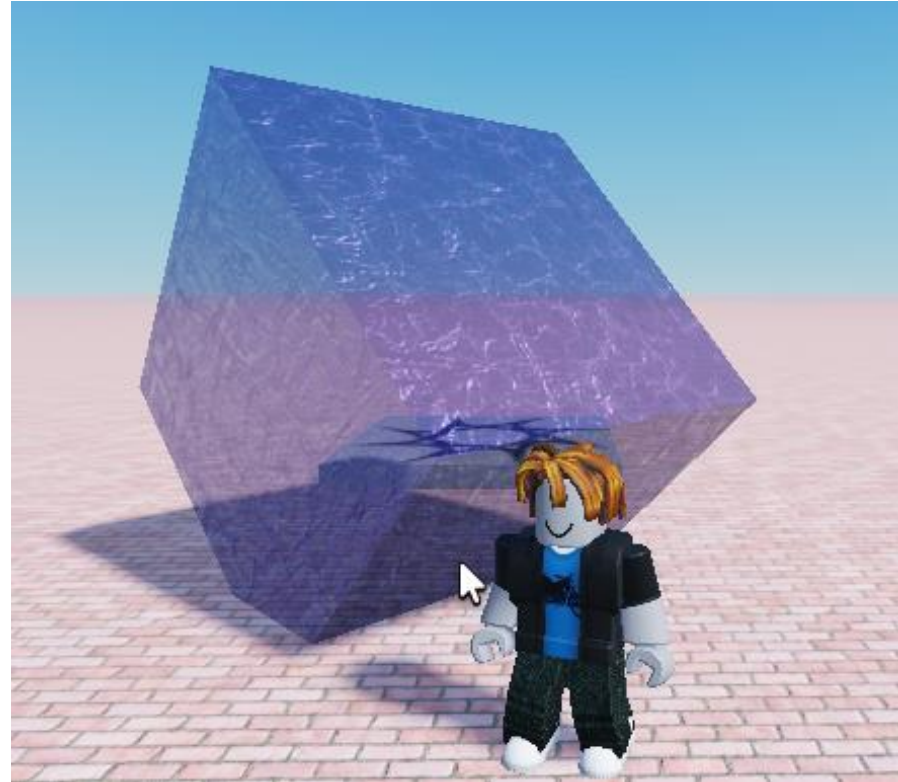
```
148  
149 ▼ local materialNames = {  
150     "Plastic",  
151     "Wood",  
152     "Slate",  
153     "Concrete",  
154     "CorrodedMetal",  
155     "DiamondPlate",  
156     "Foil",  
157     "Grass",  
158     "Ice",  
159     "Marble",  
160     "Granite",  
161     "Brick",  
162     "Pebble",  
163     "Sand",  
164     "Fabric",  
165     "SmoothPlastic",  
166     "Metal",  
167     "WoodPlanks",  
168     "Cobblestone",  
169     "Air",  
170     "Water",  
171     "Rock",  
172     "Glacier",  
173     "Snow",  
174     "Sandstone",  
175     "Mud",  
176     "Basalt",  
177     "Ground",  
178     "CrackedLava",  
179     "Neon",
```

Roblox Studio - Add .material() method to the genericShapes module

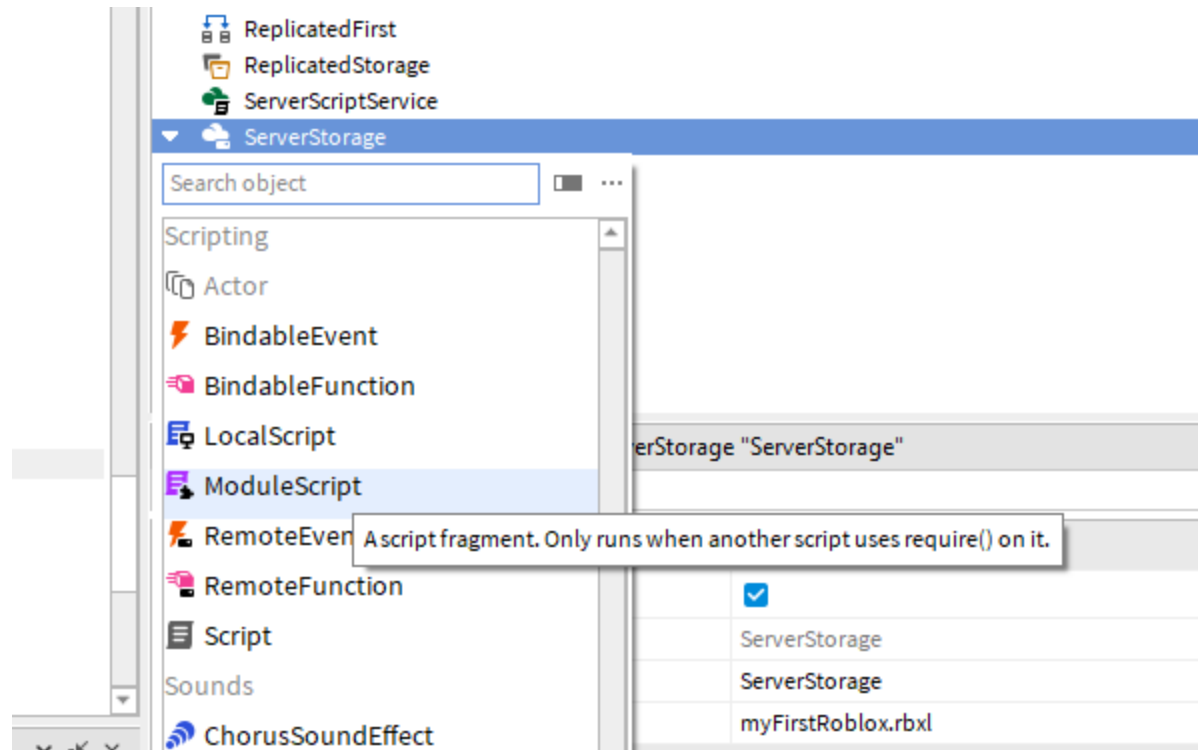
```
259
260 --This function changes the material of the object
261 ▼ function shapes.material(mainPart, value)
262     --local Materials = Enum.Material:GetEnumItems()
263     --if you want to pass an enumerated material to this function, do something like: s.material(mainPart, Enum.Material.Wood)
264 ▼ if typeof(value) == "string" or typeof(value) == "EnumItem" then
265     mainPart.Material = value
266 ▼ else
267     print("Error in the Material method. You need to pass a string. You passed a: "..typeof(value))
268 end
269 return mainPart
270 end
271
```

Roblox Studio

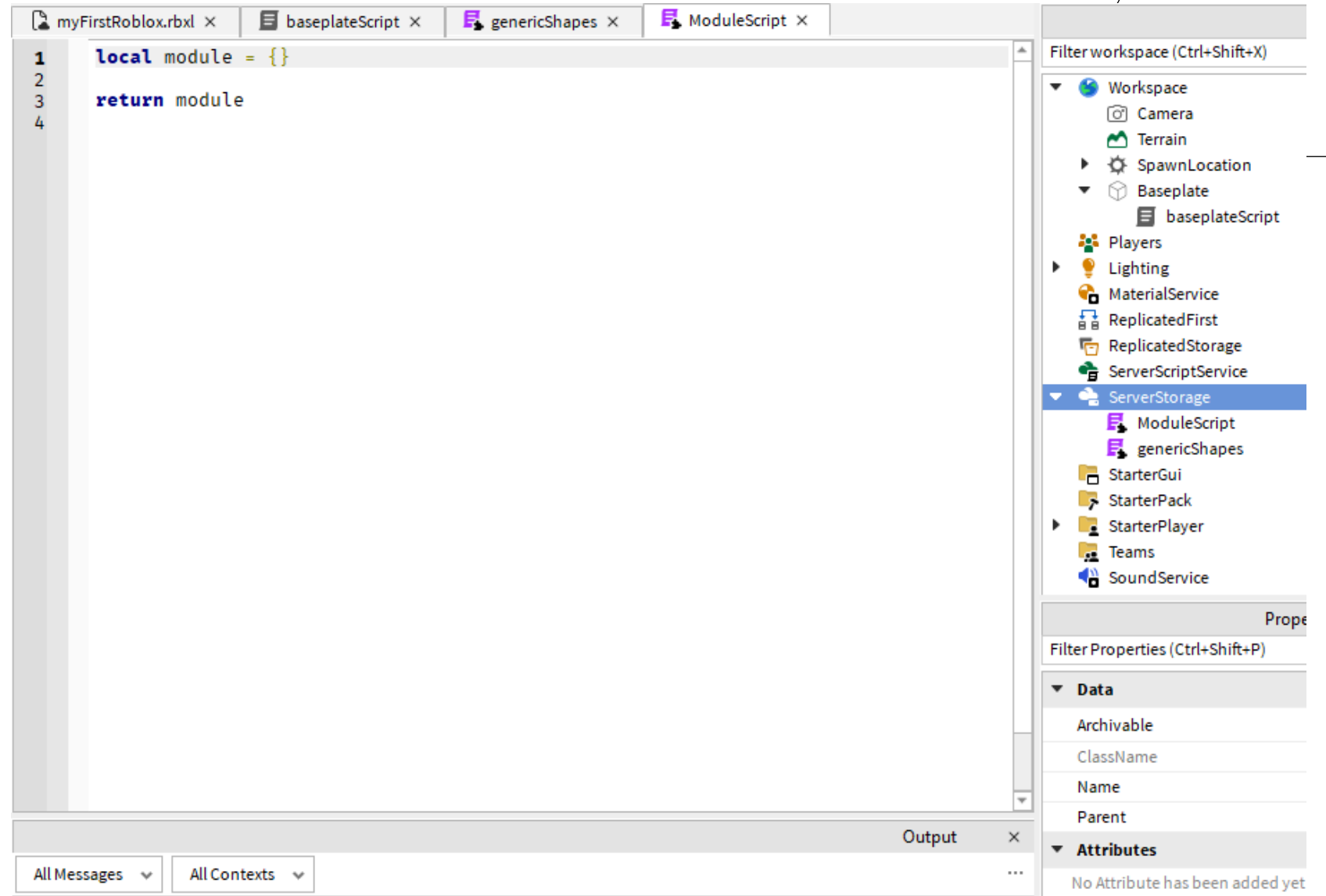
Run it...



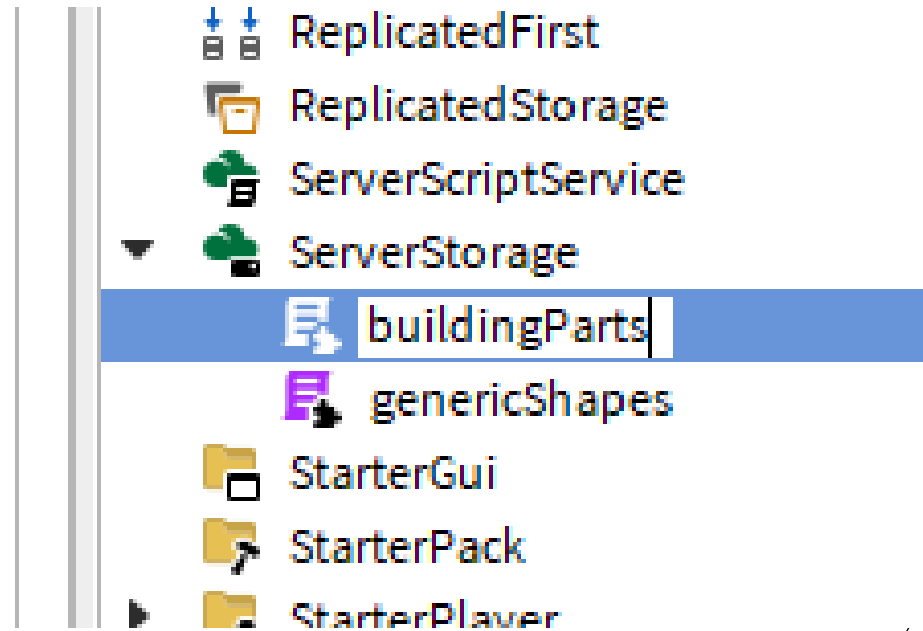
Roblox Studio - Add another ModuleScript to ServerStorage



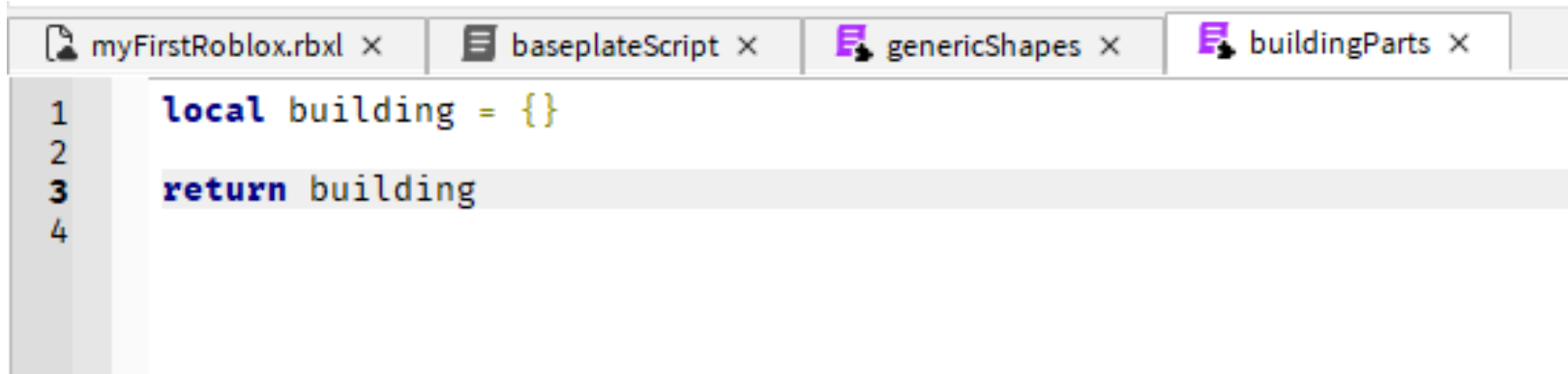
Roblox Studio -



Roblox Studio - Rename the module: buildingParts



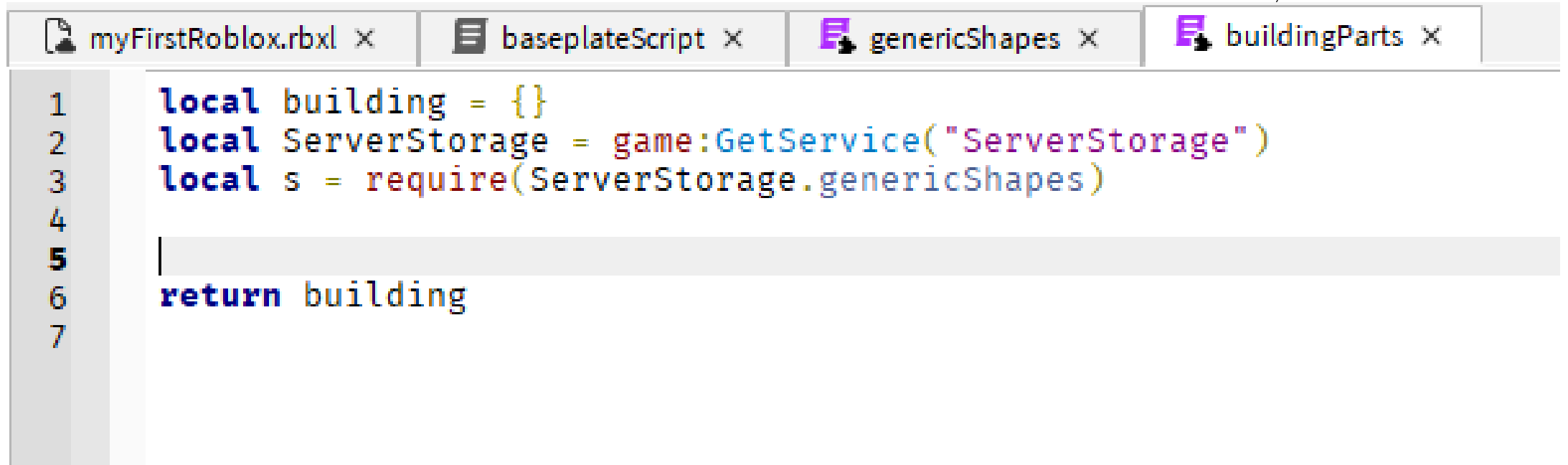
Roblox Studio - Edit the module script to use the class: building



```
myFirstRoblox.rbxl x | baseplateScript x | genericShapes x | buildingParts x  
1 local building = {}  
2  
3 return building  
4
```

Roblox Studio -

Add to lines to the buildingParts module to get the ServerStorage and access the genericShapes module.



```
myFirstRoblox.rbxl x | baseplateScript x | genericShapes x | buildingParts x
1  local building = {}
2  local ServerStorage = game:GetService("ServerStorage")
3  local s = require(ServerStorage.genericShapes)
4
5  |
6  return building
7
```

Roblox Studio -
activate the output panel

Roblox Studio -
activate the output panel

Roblox Studio -
activate the output panel

Roblox Studio -
activate the output panel

Roblox Studio -
activate the output panel

Roblox Studio -
activate the output panel

Roblox Studio -
activate the output panel

Roblox Studio -
activate the output panel

Roblox Studio -
activate the output panel