Roblox 1 : Lua Scripting Fundamentals (Continuation)

Table of Contents

Parameters and Event

Parameters are placeholders for information you want to give to the function at a later time.

This script will create a trap part that destroys whatever touches the part, including other parts. You'll have to use a parameter to set it up. Be careful to anchor the trap where it doesn't fall and destroy things unintentionally.

Create new parts

1. Create a new part that's not touching anything. If it's touching something it might go off too soon.
2. In the ***Explorer***, rename the part to ***TrapPart***.
3. Anchor the part.
4. Add a new script into the trap part. Rename the script ***TrapScript***.

Set up the script

5. Delete Hello World and add a descriptive comment.

6. Under the comment, create a new variable which finds the script's parent.

```
local trap = script.Parent
```

7. Create a local function. It can be named anything, but this lesson will use `onTouch`

```
local trap = script.Parent
local function onTouch()

end
```

8. Inside the `()`, type a name for the parameter. This lesson will use `objectTouched`.

```
local trap = script.Parent

local function onTouch(objectTouched)
```

```
end
```

9. Between local function `onTouch()` and `end`, create a `print` statement. You'll use this to check if something is touching the part in the next section.

```
local trap = script.Parent

local function onTouch(objectTouched)
    print("Something touched the trap")
end
```

Add Event Listener

Events are things that happen in the experience. Like a player touching a part or losing health. When a function is connected to an event, the function runs whenever the event happens.

The `Touched` event fires whenever one part touches another part and can be used to create buttons, traps, and other objects that players interact with.

10. Beneath the function's `end`, type `trap.Touched:Connect(onTouch)`

```
local trap = script.Parent

local function onTouch(objectTouched)
    print("Something touched the trap")
end

-- Connect the function to the Touched event
trap.Touched:Connect(onTouch)
```

11. Click ***Test*** and then touch the part. Check for your test print statement: `Something touched the trap`

12. Now that the function is correctly set up, use it to destroy whatever touches the part. Inside the function, after the string, type `objectTouched:Destroy()`

```
local trap = script.Parent

local function onTouch(objectTouched)
    print("Something touched the trap")
    -- Destroy the touching object
    objectTouched:Destroy()
end

trap.Touched:Connect(onTouch)
```

11. Test again and see what happens when the part is touched. Your avatar should end up missing feet or arms.

If Statements

Conditions can come in various forms but are often simple statements like math equations. For example, if 1+1 equals 2, then run some code. Like ordinary math equations, conditional can use operators such as plus (+) or less than (<) to evaluate statements.

1. Set up the empty conditional. In the script, type `if then`, and press Enter to autocomplete the conditional. The keyword `then` will be underlined because the code is incomplete.

```
if then
  -- empty code
end
```

2. After the keyword `if`, type a true statement such as `3 + 3 == 6`.

```
if 3 + 3 == 6 then
  -- empty code
end
```

3. Within the conditional, type `print("Hello World")`.

```
if 3 + 3 == 6 then
  print("Hello World")
end
```

4. Test your code. If three plus three is equal to six, the code will output `Hello World`