

# Functions in Roblox Exploration

Download [this Roblox file](#) and open it in Roblox Studio. Then explore with these guiding questions:

## Part 1: Playtest the Program: Button

1. Make sure that the Output Window is open.
2. Playtest the program.
3. Go to the rose patch (right in front of you).
4. What does the button (floating rectangle part) look like?

A neon orange box.

5. Touch the button. What happens? Make sure to include what happens in the game and the output window!

It prints "Touched the rose garden button" in the output window. The box turns gray and is no longer neon.

6. Try touching the button again. What happens? Make sure to include what happens in the game and the output window!

Nothing!

## Part 2: Investigate the PressButton Script

Stop playtesting. Find the **PressRoseButton** script that is attached to the **RoseButton** game object.

1. Look at line 6. Try changing the color that is assigned to the button in the `pressButton` function. Click into the parentheses and use the color pallet to choose any color other than red.
  - a. How do you think this change will affect the program?

Sample Response: This should change the color of the rose garden button.

- b. Playtest the program to see if you are correct. What happened? Were you correct or incorrect?

Sample Response: I was almost correct. The box changed after it was pressed. !

2. Where do you see the `pressButton` function being called (not defined)? Note: In this scenario, the function is called *without* adding parentheses at the end.
  - a. Line: 12
  - b. How do you think this line of code works? You don't need to be technical here, just make an educated guess!

Sample Response: This makes the `pressButton` function work when the button is touched.

### Bugs and Debugging

A **bug** in a program is an error that causes a program to not run properly. **Debugging** refers to finding and fixing bugs in programs. Notice on Line 8 there is a comment that says “DEBUG print statement.” The comment explains that the purpose of the print statement on Line 9 is to check that the function is working like it should. We can use print statements to help us determine if a function is running like we want it to.

3. Try changing Line 4 `button.CanTouch = false` to `button.CanTouch = true`.
  - a. Playtest the program. How does this change impact the program? Make sure to check the Output Window in addition to the game window.

The message that the rose button was touched is duplicated every time you touch the button.

- b. How do you think the `CanTouch` property works? You don't need to be technical here, just make an educated guess!

This property allows the part to be touched. It can be turned on (true) or off (false).

## Part 3: Collecting the Coin

Make sure you have stopped playtesting. Delete the code in the `PressRoseButton` script. Then, copy and paste the following code in the `PressRoseButton` script. This code is almost the same as the current code, except lines 2 and 12 have been added.

```
local button = script.Parent
local coin = workspace.RoseCoin

local function pressButton()
    button.CanTouch = false
    button.Material = "Plastic"
    button.Color = Color3.fromRGB(151, 43, 54)

    --DEBUG print statement
    print("Touched the rose garden button!")

    coin.Transparency = 0
end

button.Touched:Connect(pressButton)
```

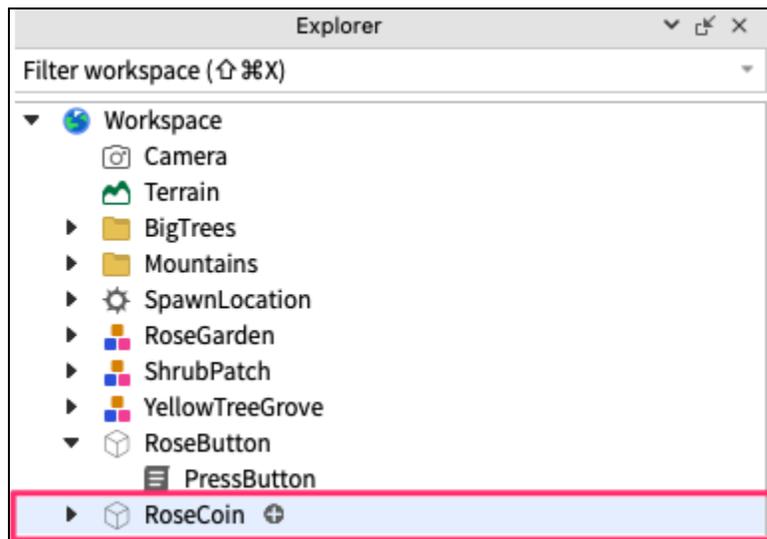
1. Playtest the program. Go to the rose patch again and press the button. What happens this time?

A coin shows up! You can collect the coin.

## Part 4: Investigate the Code for Collecting the Coin

### PressRoseButton Script

Make sure you have stopped playtesting. Look at the PressRoseButton script: Two lines were added to the PressRoseButton script. Line 2 and line 12. Line 2 is accessing the RoseCoin game object, which you can see in the Explorer Window.



1. What does line 12 (`coin.Transparency = 0`) do as part of the `pressButton` function? Check the [Roblox docs](#) if you need help remembering how the Transparency property works.

This line of code changed the transparency of the coin to 0 (invisible).

### CollectRoseCoin Script

In the Explorer Window, use the dropdown by the RoseCoin object to find the CollectRoseCoin script. Double click on the script to open it.



1. Look at lines 11 and 12. What two properties does the script initially set when the program runs?

The script sets the `CanCollide` property of the coin to false and the transparency property of the coin to 1.

2. Where do you see the `collectCoin` function being called? Note: In this scenario, the function is called *without* adding parentheses at the end.
  - a. Line: 14

- b. How do you think this line of code works? You don't need to be technical here, just make an educated guess!

Sample Response: This makes the collectCoin function work when the coin is touched.

3. Try changing Line 4 `coin.CanTouch = false` to `coin.CanTouch = true`. Run the program.
  - a. How does this change impact the program? Make sure to check the console!

It allows you to continue to touch the coin even though it has "disappeared."

- b. Why do you think this is the case?

The coin is still technically there. It's just fully transparent.

## Part 5: Synthesize

1. How do you think the `Touched` method works?

Sample Response: The `Touched` method connects a function to a part and fires the function when the part is touched.

2. How do you think the `CanTouch` property works?

Sample Response: The `CanTouch` method allows the player to touch a part.

## What's Next?

You may have noticed that there is a ladder and a coin in a tree. Right now, the ladder is transparent and you cannot climb it. In the next activity, you will add the functionality to make the ladder climbable to collect the coin in the tree.