

## Object Properties - Changing BrickColor

**Required Previous Lessons/Knowledge:** *Creating Scripts, Output Window, Testing Code, Strings*

**Lesson Running Time:** *approx 15 - 20 minutes*

**Optional Handouts and Example Files:** *Intro to Lua Handout*

**Learning Objectives and Outcomes:** *Create a script with comments that uses string variables and dot notation to change a part's BrickColor property*

### Learning About Properties

#### Teaching Exercise

Take any available object and have your class describe it to you. Ask them to describe the color, size, taste, purpose. Whatever you can think of.

Apple Example:

- Color: red-green
- Size: 3 inches
- Taste: pretty good
- Number of seeds: 5
- Explosive: No

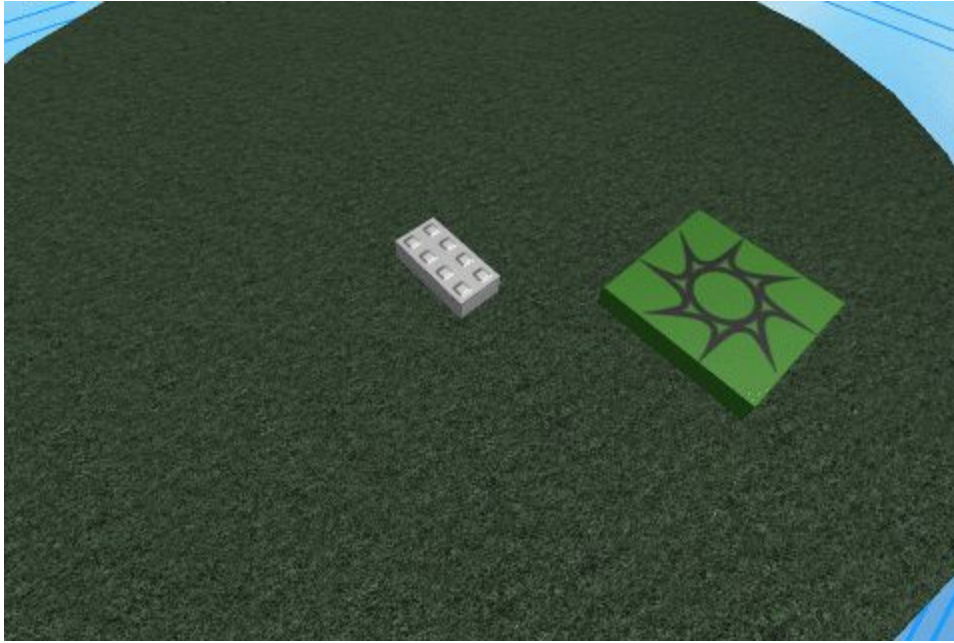
All of the things you use to describe an object are examples of properties. In a game, properties of an object might include speed, health, color, or visibility.

1. Click on any part in your game.
2. Scroll through the properties window on the right-hand side.

We can change any of the properties on a part by using code. For example, we can make a part change colors or even turn invisible.

## Changing the BrickColor Property Using Code

You're going to use code to change the BrickColor property of a part when the game starts. This can be an existing part in your game, or you can create a part to practice with.



## Setting up

Decide which part you want to change colors.

1. Select an existing part or create a new one.
2. Rename the part. In my example, my part is called *PracticePart*.
3. Create a new script (ServerScriptService > Insert Object > Script).
4. Rename the script *ChangeBrickColor*.
5. Delete the print statement at the top.

## Creating Comments

Have you ever tried to read code written by somebody else?

It can take a little time to figure out what's happening - even for experienced coders. Some coders will even completely forget what parts of their own code does what. To prevent you from forgetting what piece of code does what, always leave notes in your code. Notes that aren't actual code are called **comments**.

### Create Your First Comment

To create a comment, type `--` at the beginning of each line of your notes. Commented code will turn **green**.

- Type `--` and a note about what this script does.



The screenshot shows a Roblox script editor window with two tabs: 'IntroToCoding\_Obby\_Properties\_Start.rbxl' and 'ChangeBrickColor'. The first tab is active, showing a script with a single line of commented code. The line number '1' is in the left margin. The code is `--This script changes the color of PracticeBlock`, where the comment text is green. The cursor is at the end of the line.

```
1  --This script changes the color of PracticeBlock
```

## Coding Which Part to Change

You now know what part you want to change, but the program doesn't have a clue how to find it. Programs don't know anything you haven't told them. Give the program directions on how to get to the part you want it to work with, just like you would give friends directions on how to get to your house for the first time.

### Start at the Top

Tell the program it is looking for a part in the game by starting at, well, `game`.

- Type `game` below the comment.



The screenshot shows a Roblox Studio script editor window. The title bar contains two tabs: 'IntroToCoding\_Obby\_Properties\_Start.rbxl' and 'ChangeBrickColor'. The script content is as follows:

```
1  --This script changes the color of PracticeBlock
2  game|
```

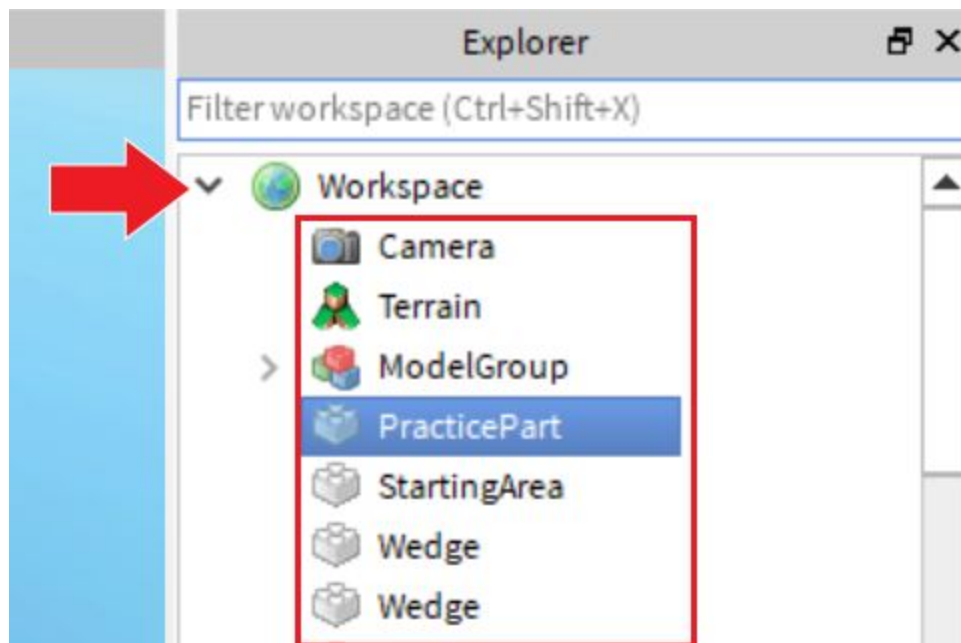
## Look in Explorer

**The Explorer window** lists all of the objects in your game and shows how they are related to each other. You can use Explorer to determine the next step in the directions to your part.

**Question:** Look at your Explorer window. Which of the following objects is *PracticePart* under?

- Workspace
- ServerScriptStorage
- Teams

**Answer:** The Workspace!



### Children & Parents

Parts nested under an object the way *PracticePart* is nested under *Workspace*, are referred to as children of the part at the top.

## Use Dot Notation

We have to tell the program to go to Workspace next, but we don't want all the words to run together like a massive hashtag. Separate the names of objects and properties from each other using a dot. Always type the names exactly as you see them and don't use any spaces.

1. On the same line as `game`, type `.Workspace`

### Troubleshooting Tip

Make sure you capitalize `Workspace`

The next set of directions is the part you want to work with.

2. Still on the same line, type `.` followed by the `NameOfYourPart` that you are working with.

## Check the Code

If your part is named *PracticePart*, your final code is `game.Workspace.PracticePart`

## Use Autocomplete

You'll notice as you type that Roblox will autocomplete some words for you. If you see the object you want, select it with your mouse or arrow buttons and press **Enter** on the keyboard.

## Coding Which Property to Change

Great success! You've directed the script to which part it should work with. Now use code to tell it which property to change. To change the part color, we're going to replace a string variable in the **BrickColor** property.

To work with the BrickColor property:

- Type `.BrickColor` after the name of your part.

### Code Example

```
--Changes BrickColor of NameOfYourPart  
game.Workspace.NameOfYourPart.BrickColor
```

## Replace the Old BrickColor String

Like the print function, the **BrickColor** property also uses a string variable to store information. By default, the string is "Medium stone grey". That's a boring color. Tell the program to replace "Medium stone grey" with a new, more exciting color.

- On the same line, type `= BrickColor.new()`

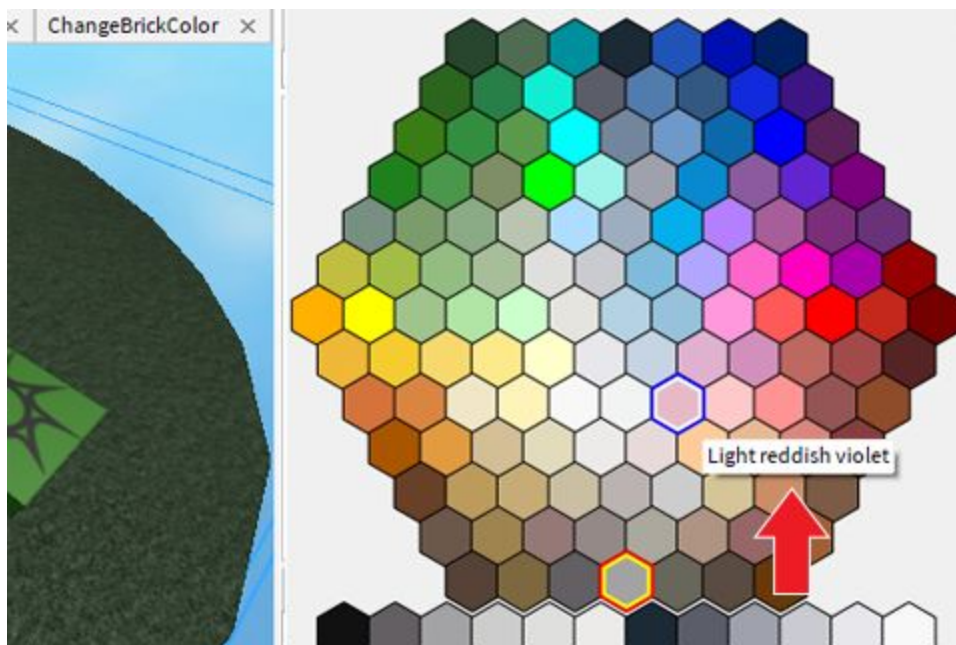
### Code Example

```
--Changes BrickColor of NameOfYourPart  
game.Workspace.NameOfYourPart.BrickColor = BrickColor.new()
```

## Pick a New Color

When you hover over a name in the color picker, the name of the color will appear.

1. Bring up the **BrickColor** color picker.
2. Hover over the color you want to use.
3. Remember the name exactly. It might help to write it down with matching capitalization and spacing.



## Change the String

- In between ( and ) type the name of the material you chose as a string. Remember to match the capitalization.
  - `game.Workspace.NameOfYourPart.BrickColor = BrickColor.new("Color name")`

## Tutorial Example

```
--Changes BrickColor of NameOfYourPart  
game.Workspace.PracticePart.BrickColor = BrickColor.new("Light  
reddish violet")
```

## Troubleshooting Your Code

- `NameOfYourPart` needs to be replaced with the exact name of your part. Match the capitalization exactly
- **Strings can have spaces.**
- **Match the capitalization** and spacing of the color you selected.
- **If your part turns grey**, you most likely have not put in the name of a real color or your capitalization doesn't match. Pro tip, Roblox capitalizes the first word in a string. Other words are lowercase.