Roblox Coding Lesson 4: Variables

For 10-11 Year Old Beginners

What Are Variables?

Variables are like labeled containers that store information. They let us:

- Save information to use later
- Change information during gameplay
- Remember things between events

Mini-Challenge 1: Growing Cube

What you'll learn: How to use variables to track and change values

Building Steps:

1. Insert a Part into your game (make it a cube)
2. Change its color to bright red
3. Add a Script inside the part
4. Add a ClickDetector to the part

The Code (only 9 lines!):

```
local cube = script.Parent
-- TODO: Create a variable called 'size' and set it to 1

-- Make sure our cube starts at the right size
cube.Size = Vector3.new(size, size, size)

local function onCubeClicked()
    -- TODO: Update the 'size' variable by adding 0.5 to it

    -- TODO: Use your updated 'size' variable to change the cube's size
end
cube.ClickDetector.MouseClick:Connect(onCubeClicked)
```

What to Explain:

- The variable `size` remembers how big the cube should be

- Each time we click, we change the variable's value
- We use the variable to set the actual size of the cube
- Without variables, the cube wouldn't "remember" its current size

Setup Tip:

- Make sure the cube is in a place where it has room to grow!

Mini-Challenge 2: Speed Booster

What you'll learn: How to use variables to control player movement

Building Steps:

1. Insert a Part into your game (make it look like a pad)
2. Make it flat and color it bright blue
3. Add a Script inside the part

The Code (only 10 lines!):

```
local speedPad = script.Parent
-- TODO: Create a variable called 'normalSpeed' and set it to 16
-- TODO: Create a variable called 'boostSpeed' and set it to 50
-- TODO: Create a variable called 'boostTime' and set it to 3

local function onPlayerTouch(otherPart)
    local character = otherPart.Parent
    local humanoid = character:FindFirstChildOfClass("Humanoid")
    if humanoid then
        -- TODO: Set the humanoid's WalkSpeed to your boostSpeed variable

        wait(boostTime)  -- Wait for boost time

        -- TODO: Set the humanoid's WalkSpeed back to normalSpeed
    end
end
speedPad.Touched:Connect(onPlayerTouch)
```

What to Explain:

- We use variables to store different speed values
- We use a variable to control how long the boost lasts
- Variables make it easy to adjust values without rewriting code
- Try changing the variable values to see how the game feels different

Setup Tip:

- Place the speed pad where players can easily run over it
- Make it a different color so players know it's special

Mini-Challenge 3: Color Cycling Sign

What you'll learn: How to use variables to cycle through options

Building Steps:

1. Insert a Part into your game (make it flat like a sign)
2. Insert a SurfaceGui into the part
3. Insert a TextLabel into the SurfaceGui
4. Add a Script inside the part
5. Add a ClickDetector to the part

The Code (only 13 lines!):

```
local sign = script.Parent
local label = sign.SurfaceGui.TextLabel
-- TODO: Create a variable called 'colorIndex' and set it to 1

-- List of colors to cycle through
local colors = {"Really red", "Bright blue", "Lime green", "New Yeller"}
local messages = {"RED", "BLUE", "GREEN", "YELLOW"}

label.TextScaled = true

local function onSignClicked()
    -- TODO: Update colorIndex to cycle to the next color
    -- Hint: Use (colorIndex % #colors) + 1

    sign.BrickColor = BrickColor.new(colors[colorIndex])
    label.Text = messages[colorIndex]
```

```
end
sign.ClickDetector.MouseClick:Connect(onSignClicked)
onSignClicked()  -- Set initial color and message
```

What to Explain:

- The variable `colorIndex` keeps track of which color we're showing
- We use the variable to pick from our list of colors
- The `%` (modulo) operator helps us cycle back to the beginning
- Variables help us remember where we are in a sequence

Setup Tip:

- Make the sign big enough to read easily
- Make sure the TextLabel fills the whole SurfaceGui

Mini-Challenge 4: Counting Door

What you'll learn: How to use variables to create a puzzle

Building Steps:

1. Insert a Part into your game (make it look like a door)
2. Insert another Part to be a button
3. Add a Script inside the button
4. Add a ClickDetector to the button

The Code (only 15 lines!):

```
local button = script.Parent
local door = workspace.Door  -- Change this to your door's name
-- TODO: Create a variable called 'clicksNeeded' and set it to 5
-- TODO: Create a variable called 'clickCount' and set it to 0

-- Create a BillboardGui to show progress
local countDisplay = Instance.new("BillboardGui", button)
```

```
local textLabel = Instance.new("TextLabel", countDisplay)
countDisplay.Size = UDim2.new(0, 50, 0, 20)
countDisplay.StudsOffset = Vector3.new(0, 3, 0)
textLabel.Size = UDim2.new(1, 0, 1, 0)
textLabel.TextScaled = true

local function onButtonClicked()
    -- TODO: Increase the clickCount variable by 1

    textLabel.Text = clickCount .. "/" .. clicksNeeded

    -- TODO: Check if clickCount is greater than or equal to clicksNeeded
    -- If it is, make the door non-solid and transparent
end
button.ClickDetector.MouseClick:Connect(onButtonClicked)
textLabel.Text = clickCount .. "/" .. clicksNeeded  -- Initial display
```

What to Explain:

- The variable `clickCount` remembers how many times we've clicked
- We compare our variable to `clicksNeeded` to decide when to open the door
- Variables let us create puzzles that require multiple steps
- Try changing the `clicksNeeded` variable to make it easier or harder

Setup Tip:

- Name your door "Door" or change the code to match your door's name
- Position the button near the door so players understand they're connected

Teaching This Lesson

Classroom Setup (5 minutes)

1. Prepare a basic Roblox place with parts already inserted
2. Have the challenges written on a board or handout
3. Make sure all students have Roblox Studio open

Introduction (5 minutes)

- Show examples of variables in popular Roblox games:
  - Health bars that remember damage
  - Scores that increase over time
  - Doors that open after collecting items
- Connect to previous lesson: "Last time we learned about data types (boolean, string, number). Today we'll learn how to store and change these values using variables!"

Demonstration (10 minutes)

- Build the Growing Cube challenge while students watch
- Explain each line as you type it:
  - "First we create a variable called 'size' and give it a starting value"
  - "When we click, we change the variable's value"
  - "Then we use the variable to set the cube's actual size"
- Show how changing the variable affects the game

Guided Practice (15 minutes)

- Have students build their first challenge (Growing Cube)
- Walk around and help as needed
- Ask questions like:
  - "What happens if you change the starting size?"
  - "What happens if you add more each time?"
  - "What would happen without the variable?"

Independent Practice (20 minutes)

- Let students try the other challenges
- Encourage them to customize their projects
- Have early finishers help others or try the extensions

Wrap-Up (5 minutes)

- Ask students to share their creations
- Review key concepts:
  - Variables store information

    - Variables can change during the game
    - Variables help things "remember" information
- Preview next lesson: "Next time we'll learn about if-statements to make decisions in our code!"

Extensions for Advanced Students

- Make the cube change color based on its size
- Add a maximum size to the cube
- Make the speed pad give different boosts based on which side you touch
- Create a sign that changes both color and shape
- Make a door that requires a specific pattern of clicks (like a combination lock)