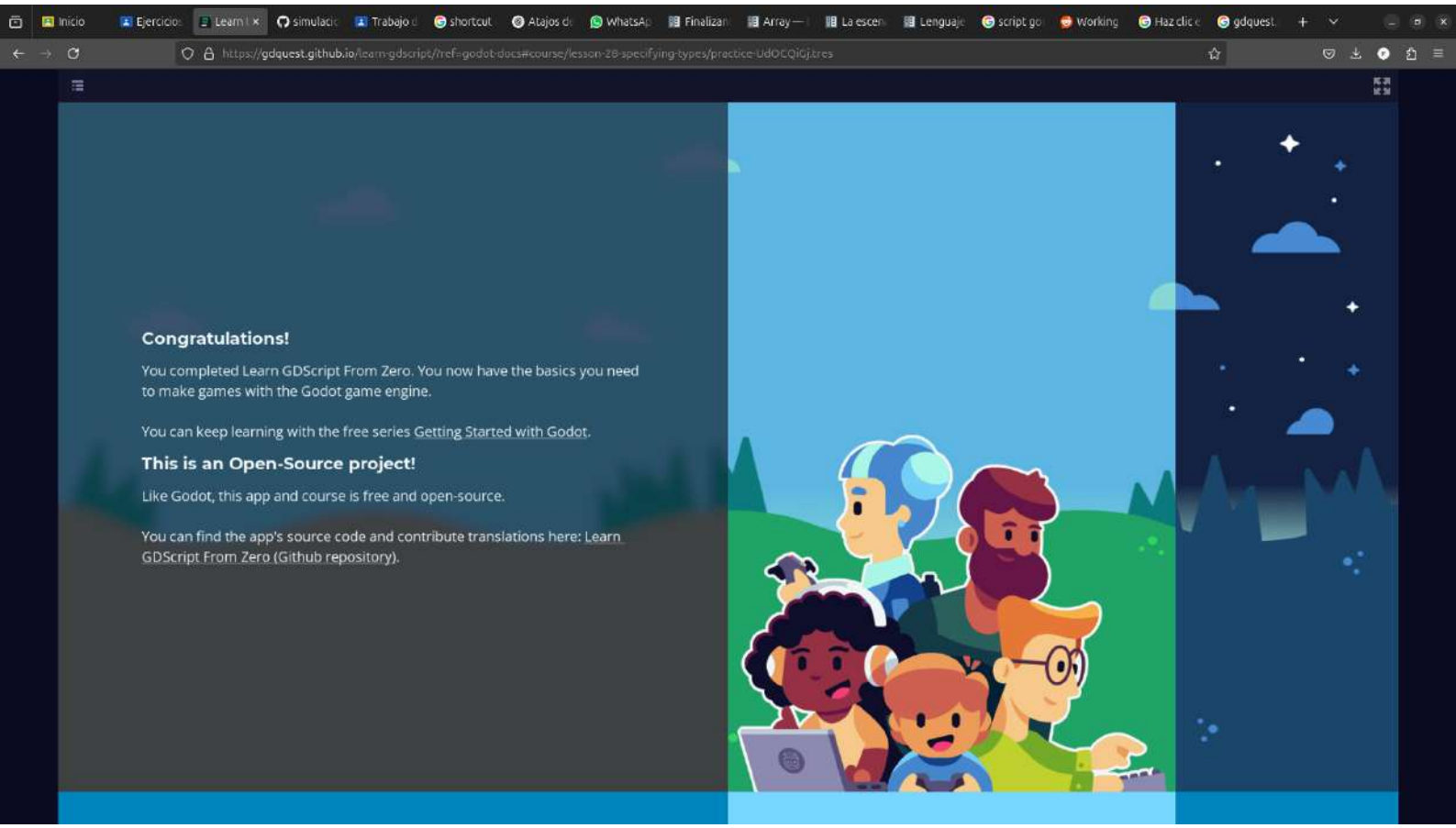
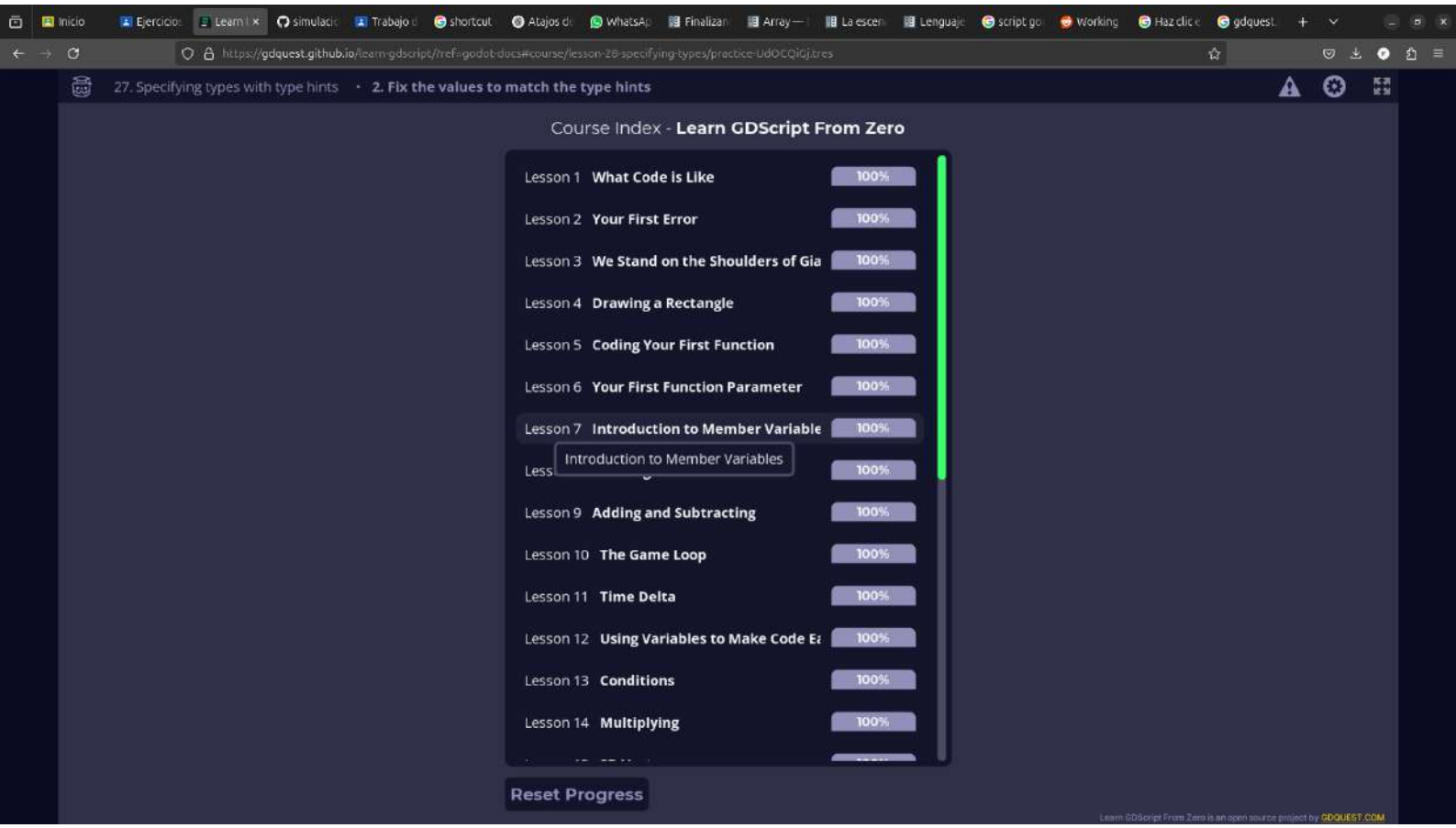
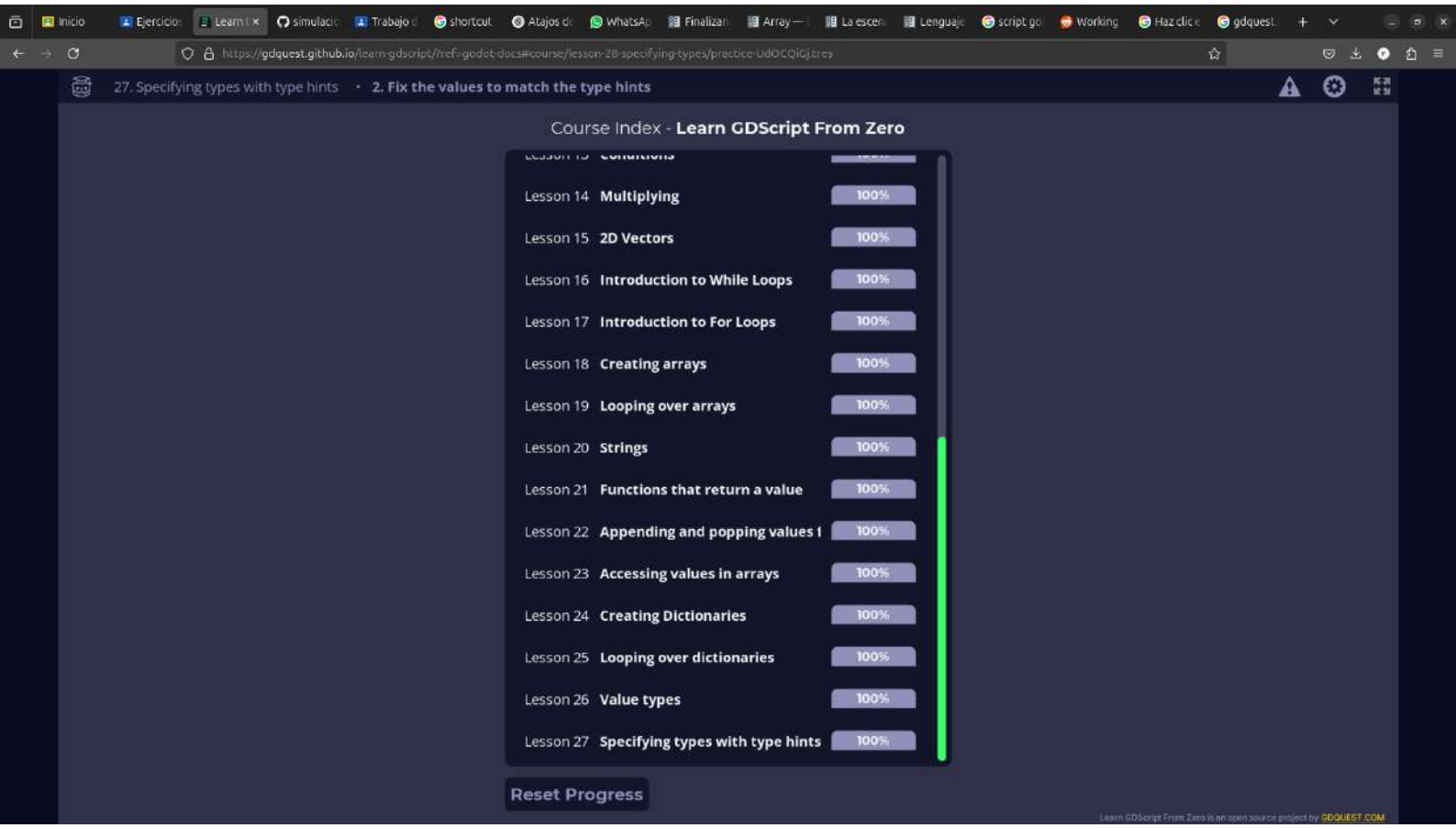
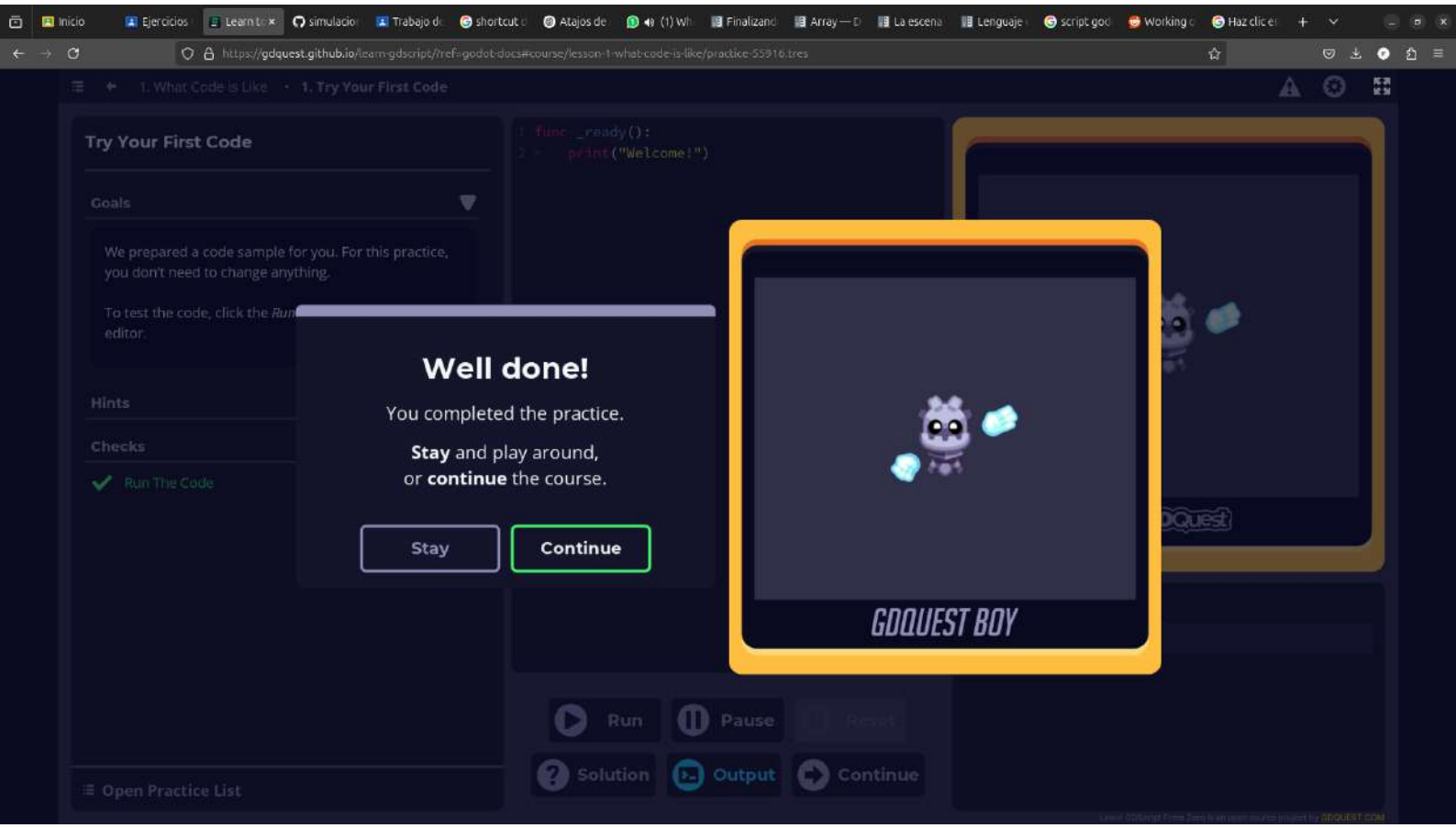
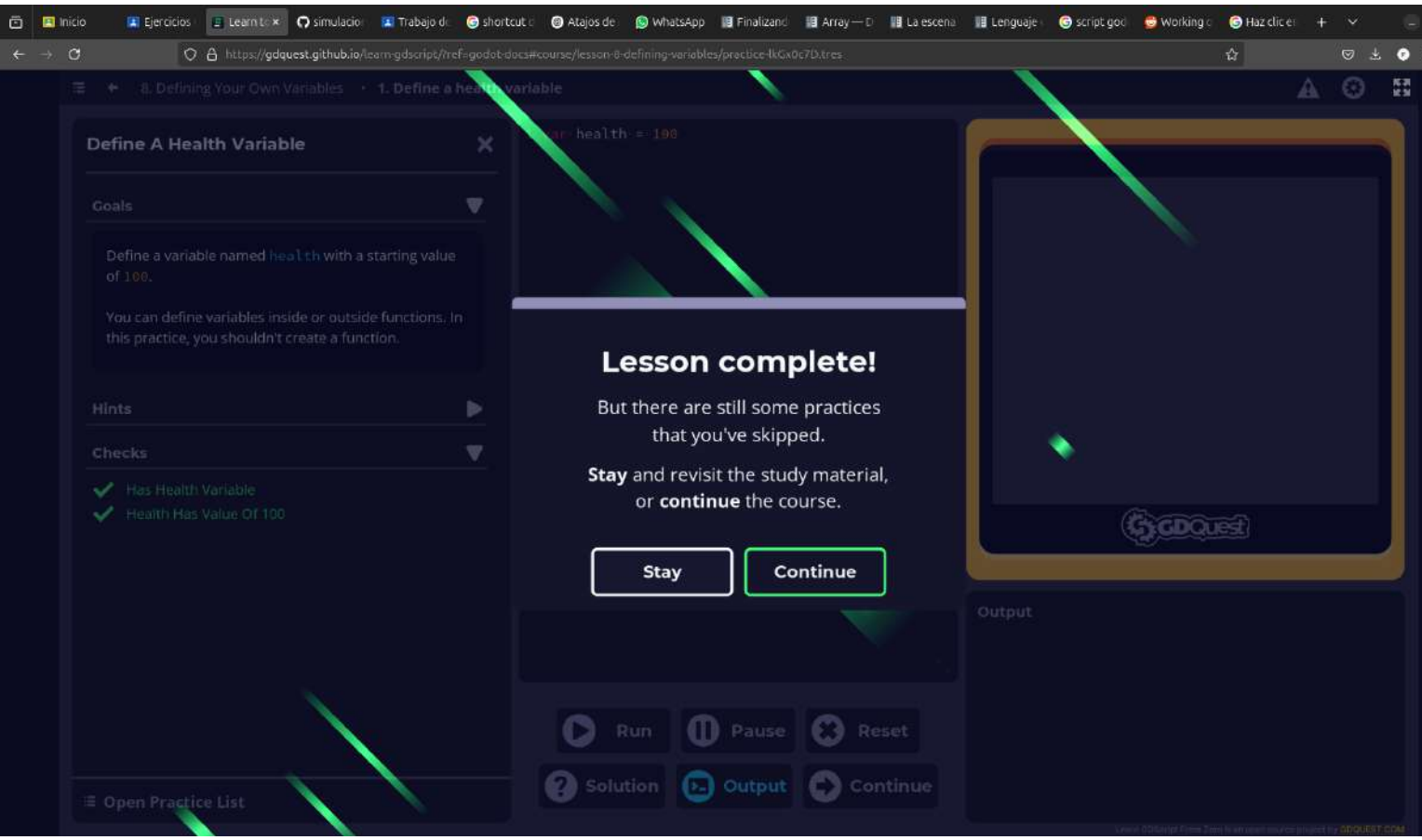
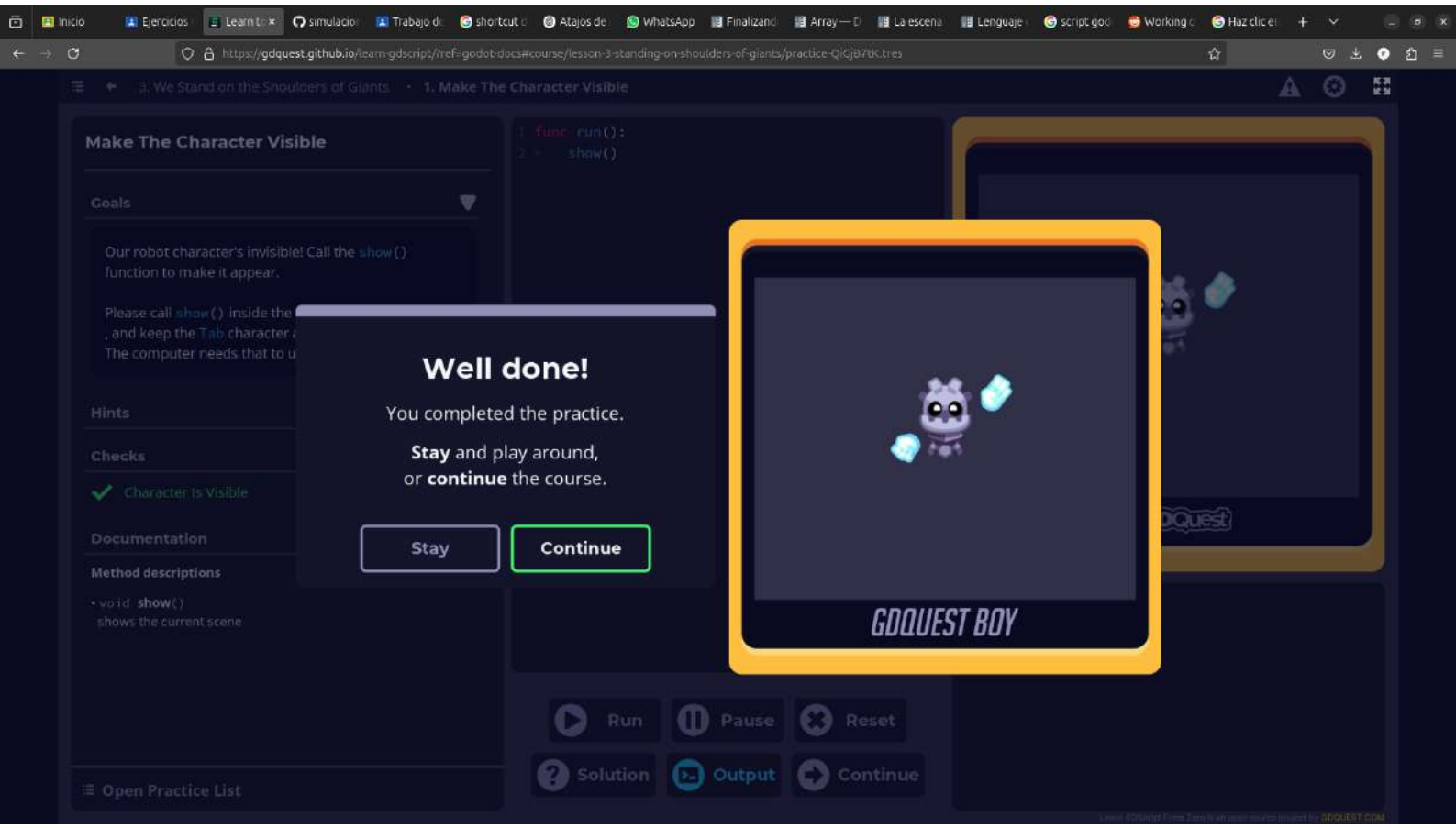

Congratulations!
You completed Learn GDScript From Zero. You now have the basics you need to make games with the Godot game engine.
You can keep learning with the free series Getting Started with Godot.
This is an Open-Source project!
Like Godot, this app and course is free and open-source.
You can find the app's source code and contribute translations here: Learn GDScript From Zero (Github repository).

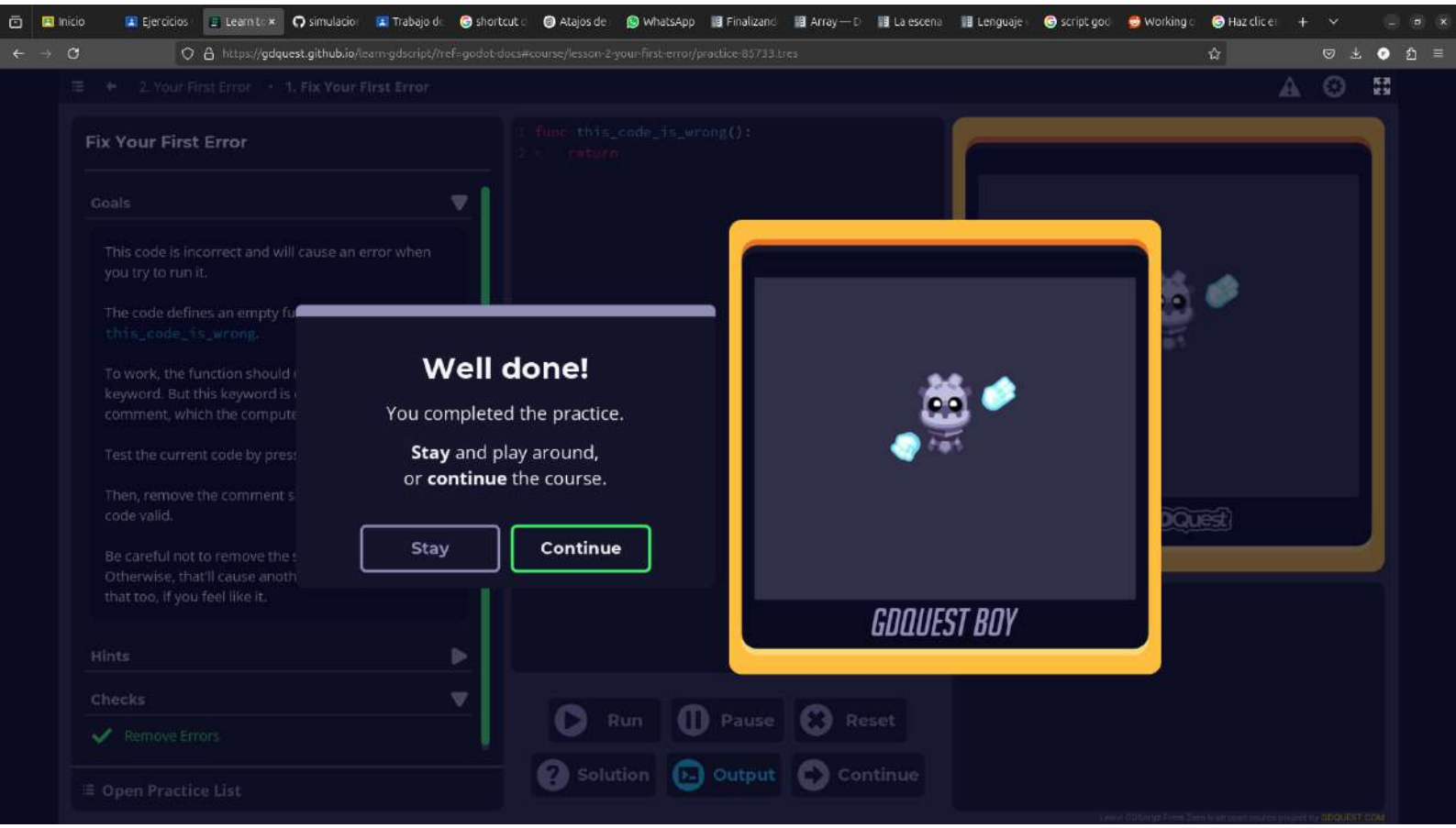

Inicio
Ejercicios
Learn
simulacio
Trabajo d
shortcut
Atajos de
WhatsAp
Finalizan
Array —
La escen
Lenguaje
script go
Working
Haz clic e
gdquest
https://gdquest.github.io/learn-gdscript/?ref=godot-docs#course/lesson-28-specifying-types/practice-UdOCQiGj.tres
27. Specifying types with type hints
2. Fix the values to match the type hints
Course Index - Learn GDScript From Zero
Lesson 1 What Code is Like 100%
Lesson 2 Your First Error 100%
Lesson 3 We Stand on the Shoulders of Gia 100%
Lesson 4 Drawing a Rectangle 100%
Lesson 5 Coding Your First Function 100%
Lesson 6 Your First Function Parameter 100%
Lesson 7 Introduction to Member Variable 100%
Introduction to Member Variables
Less 100%
Lesson 9 Adding and Subtracting 100%
Lesson 10 The Game Loop 100%
Lesson 11 Time Delta 100%
Lesson 12 Using Variables to Make Code Ea 100%
Lesson 13 Conditions 100%
Lesson 14 Multiplying 100%
Reset Progress
Learn GDScript From Zero is an open source project by GDQUEST.COM

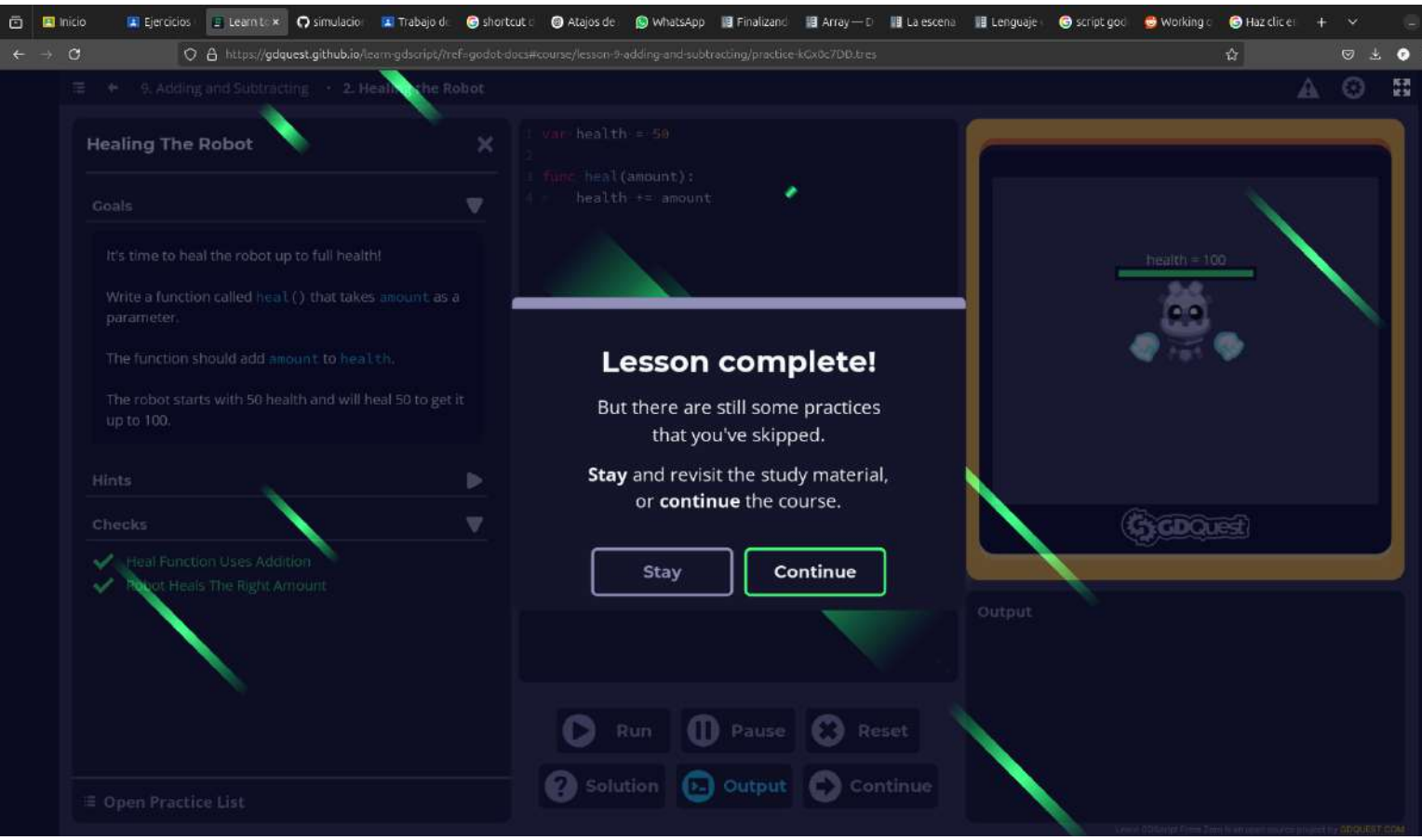

Inicio
Ejercicios
Learn
simulacio
Trabajo d
shortcut
Atajos de
WhatsAp
Finalizan
Array —
La escen
Lenguaje
script go
Working
Haz clic e
gdquest
https://gdquest.github.io/learn-gdscript/?ref=godot-docs#course/lesson-28-specifying-types/practice-UdOCQiGj.tres
27. Specifying types with type hints • 2. Fix the values to match the type hints
Course Index - Learn GDScript From Zero
Lesson 14 Multiplying 100%
Lesson 15 2D Vectors 100%
Lesson 16 Introduction to While Loops 100%
Lesson 17 Introduction to For Loops 100%
Lesson 18 Creating arrays 100%
Lesson 19 Looping over arrays 100%
Lesson 20 Strings 100%
Lesson 21 Functions that return a value 100%
Lesson 22 Appending and popping values f 100%
Lesson 23 Accessing values in arrays 100%
Lesson 24 Creating Dictionaries 100%
Lesson 25 Looping over dictionaries 100%
Lesson 26 Value types 100%
Lesson 27 Specifying types with type hints 100%
Reset Progress
Learn GDScript From Zero is an open source project by GDQUEST.COM

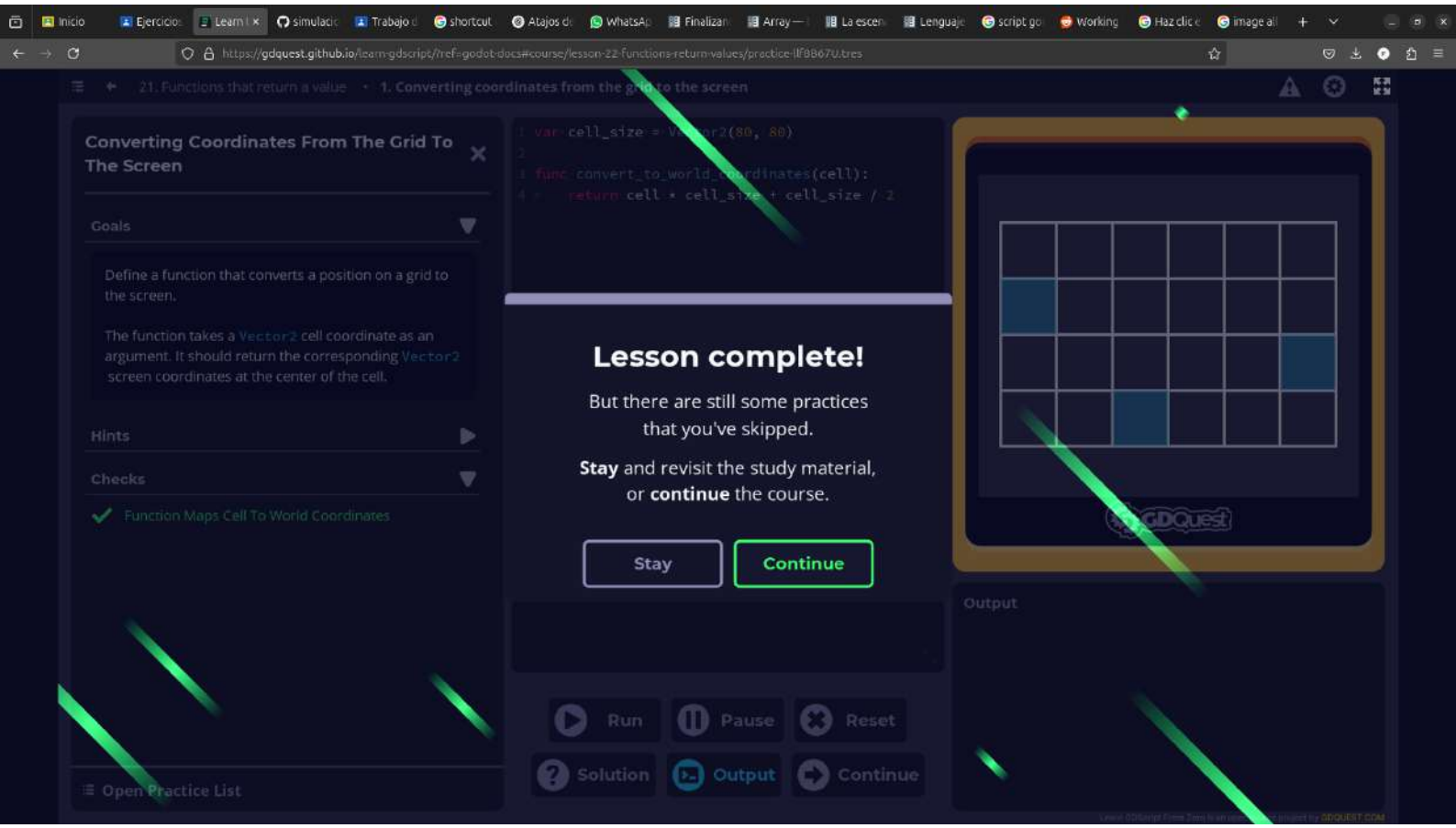

Inicio
Ejercicios
Learn to
simulacion
Trabajo de
shortcut
Atajos de
(1) Wh
Finalizand
Array — D
La escena
Lenguaje
script god
Working
Haz clic en
https://gdquest.github.io/learn-gdscript/?ref=godot-docs#course/lesson-1-what-code-is-like/practice-55916.tres
1. What Code is Like · 1. Try Your First Code
Try Your First Code
Goals
We prepared a code sample for you. For this practice, you don't need to change anything.
To test the code, click the Run
editor.
Hints
Checks
Run The Code
Open Practice List
func _ready():
print("Welcome!")
Well done!
You completed the practice.
Stay and play around,
or continue the course.
Stay
Continue
GDQUEST BOY
Run
Pause
Solution
Output
Continue

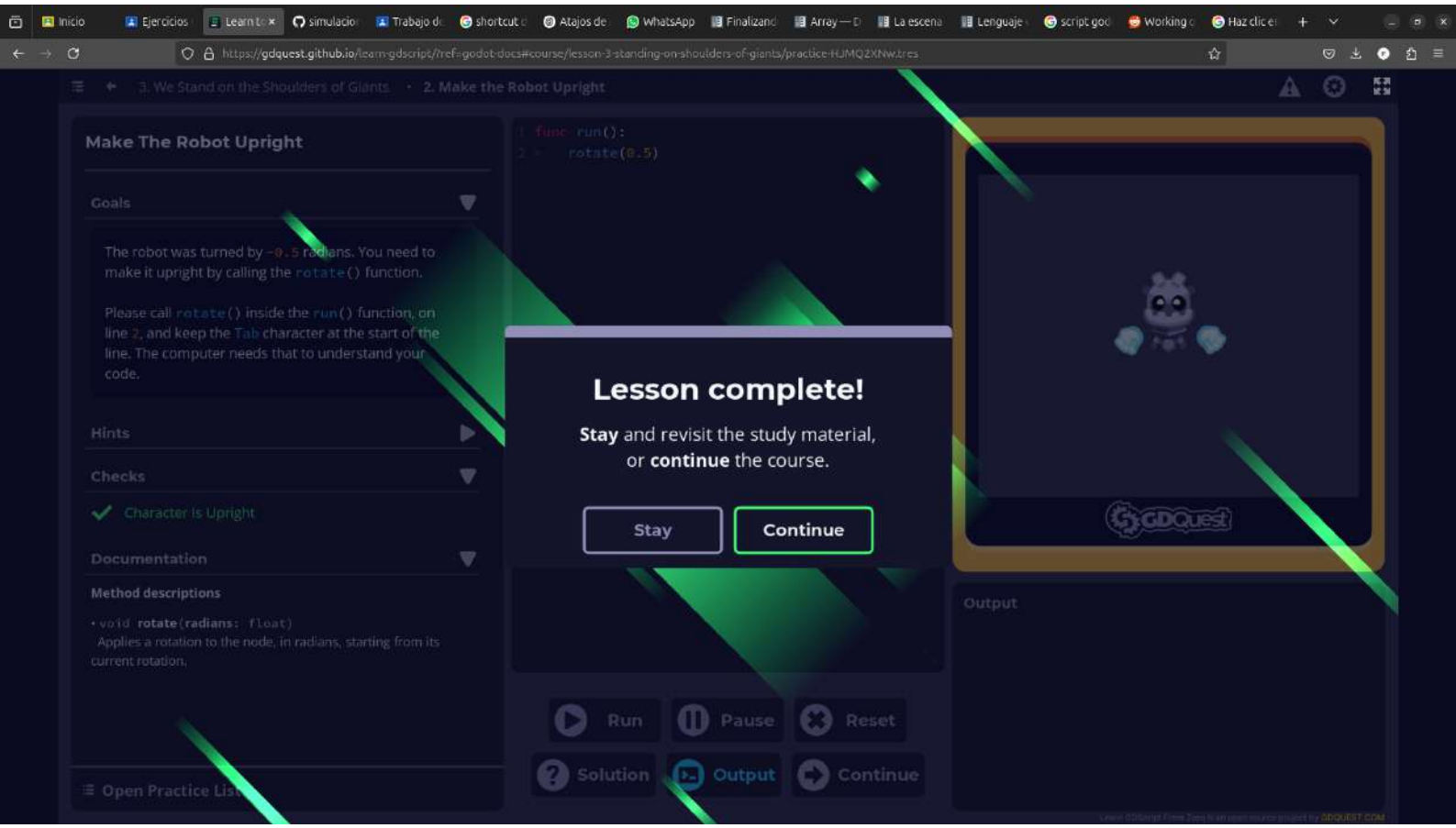

Inicio
Ejercicios
Learn to
simulacio
Trabajo d
shortcut
Atajos de
WhatsApp
Finalizand
Array — D
La escena
Lenguaje
script god
Working
Haz clic e
https://gdquest.github.io/learn-gdscript/?ref=godot-docs#course/lesson-8-defining-variables/practice-1kGx0c7D.tres
8. Defining Your Own Variables
1. Define a health variable
Define A Health Variable
Goals
Define a variable named health with a starting value of 100.
You can define variables inside or outside functions. In this practice, you shouldn't create a function.
Hints
Checks
Has Health Variable
Health Has Value Of 100
Open Practice List
var health = 100
Lesson complete!
But there are still some practices that you've skipped.
Stay and revisit the study material, or continue the course.
Stay
Continue
GDQuest
Output
Run
Pause
Reset
Solution
Output
Continue

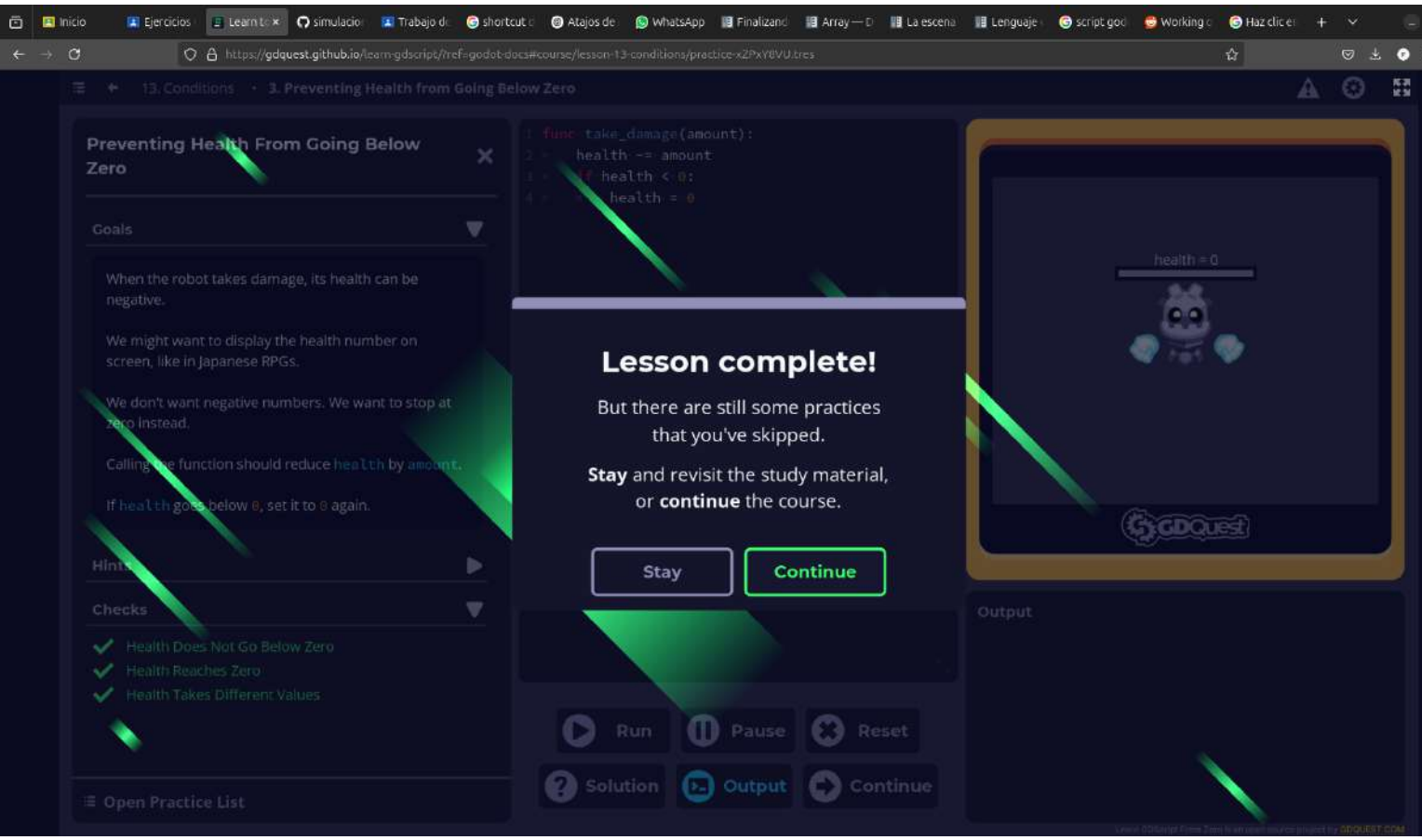

Inicio
Ejercicios
Learn to
simulacion
Trabajo de
shortcut
Atajos de
WhatsApp
Finalizand
Array — D
La escena
Lenguaje
script god
Working
Haz clic en
https://gdquest.github.io/learn-gdscript/?ref=godot-docs#course/lesson-3-standing-on-shoulders-of-giants/practice-QiGjB7tK.tres
3. We Stand on the Shoulders of Giants
1. Make The Character Visible
Make The Character Visible
Goals
Our robot character's invisible! Call the show() function to make it appear.
Please call show() inside the
, and keep the Tab character
The computer needs that to u
Hints
Checks
Character Is Visible
Documentation
Method descriptions
void show()
shows the current scene
Open Practice List
func run():
show()
Well done!
You completed the practice.
Stay and play around,
or continue the course.
Stay
Continue
GDQUEST BOY
Run
Pause
Reset
Solution
Output
Continue

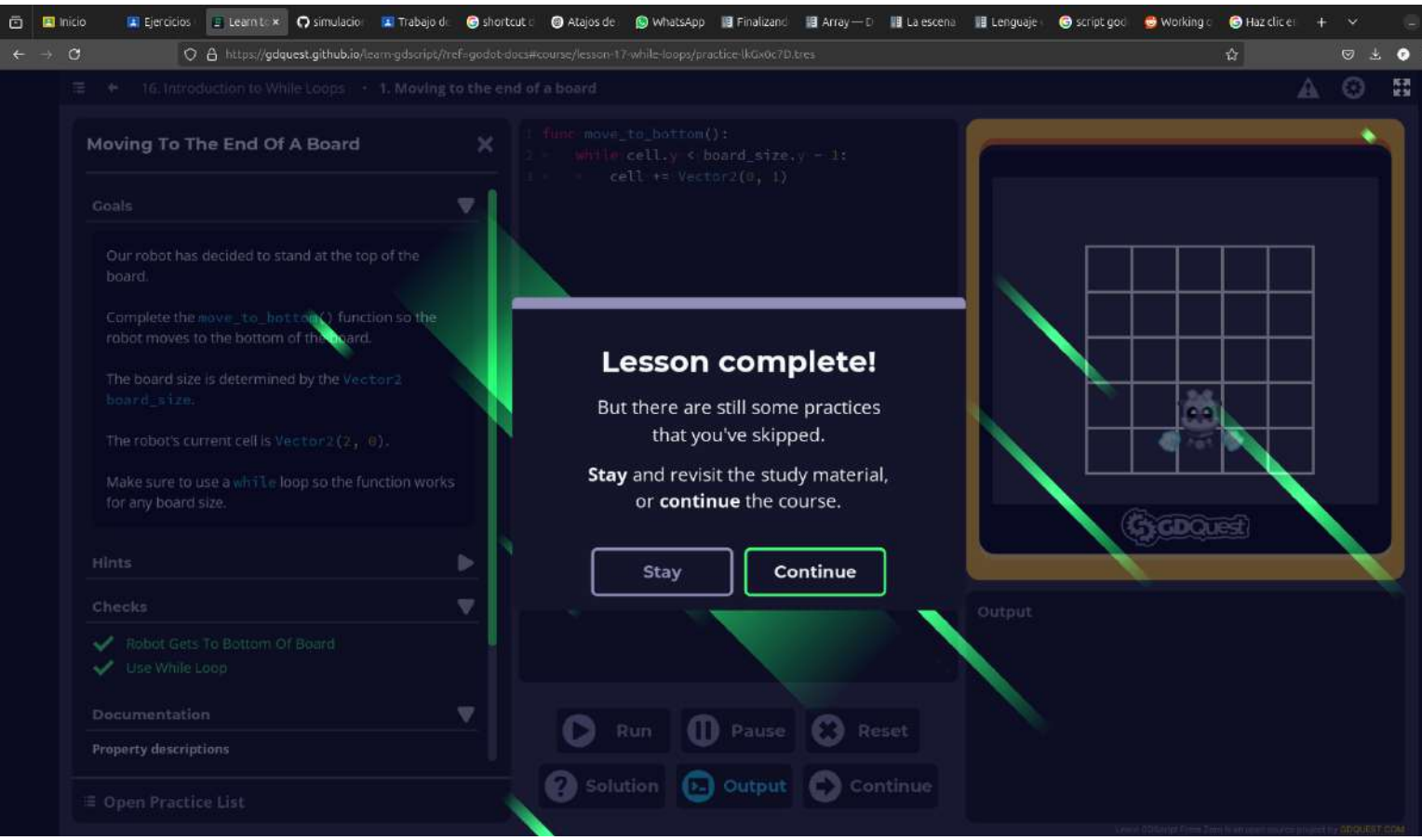

Inicio
Ejercicios
Learn to
simulacio
Trabajo d
shortcut
Atajos de
WhatsApp
Finalizand
Array — D
La escena
Lenguaje
script god
Working
Haz clic e
https://gdquest.github.io/learn-gdscript/?ref=godot-docs#course/lesson-2-your-first-error/practice-85733.tres
2. Your First Error
1. Fix Your First Error
Fix Your First Error
Goals
This code is incorrect and will cause an error when you try to run it.
The code defines an empty fu
this_code_is_wrong.
To work, the function should
keyword. But this keyword is
comment, which the compute
Test the current code by pres
Then, remove the comment s
code valid.
Be careful not to remove the
Otherwise, that'll cause anoth
that too, if you feel like it.
Hints
Checks
Remove Errors
Open Practice List
func this_code_is_wrong():
return
Well done!
You completed the practice.
Stay and play around,
or continue the course.
Stay
Continue
GDQUEST BOY
Run
Pause
Reset
Solution
Output
Continue

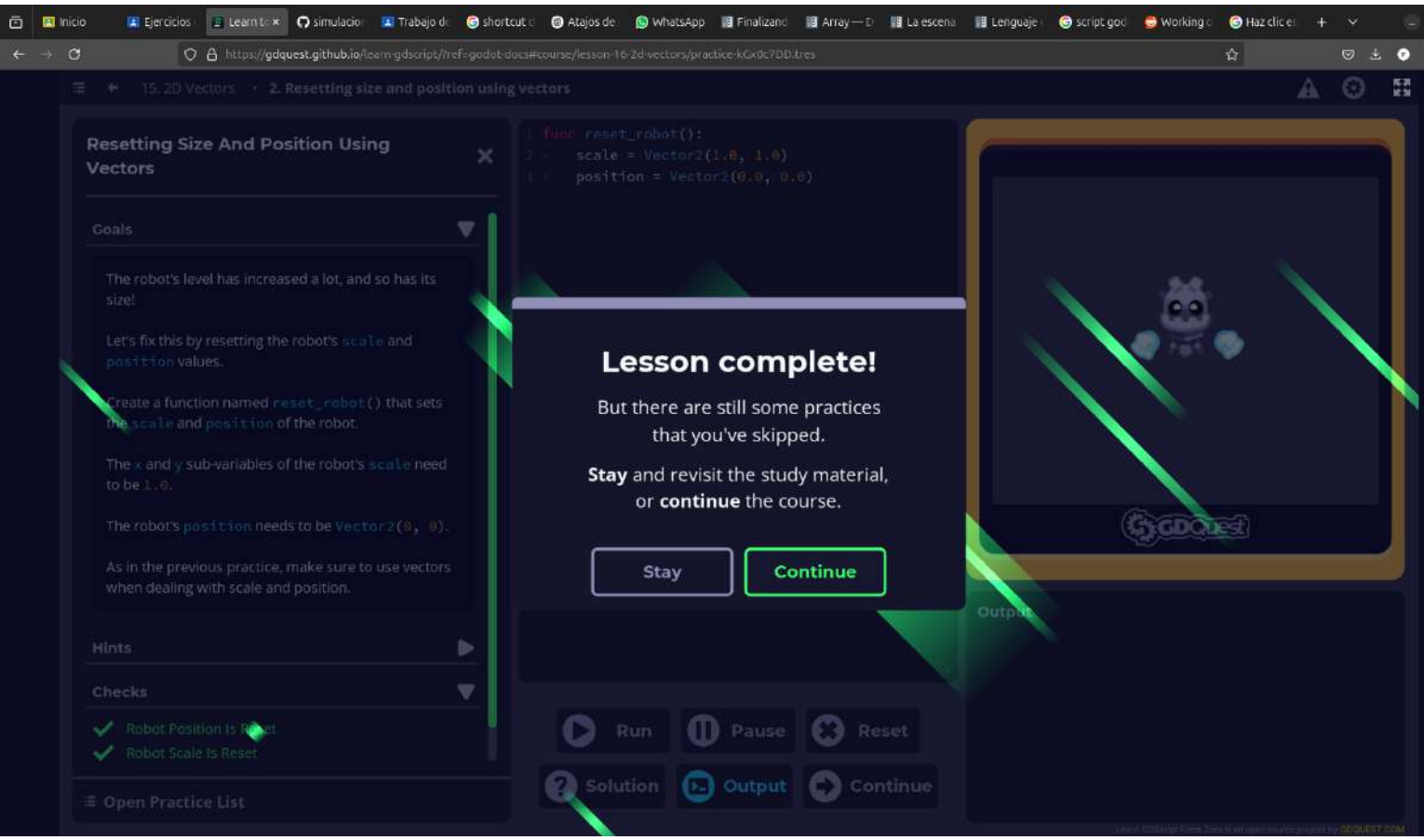

Inicio
Ejercicios
Learn to
simulacion
Trabajo de
shortcut
Atajos de
WhatsApp
Finalizand
Array — D
La escena
Lenguaje
script god
Working
Haz clic en
https://gdquest.github.io/learn-gdscript/?ref=godot-docs#course/lesson-9-adding-and-subtracting/practice-kGx0c7DD.tres
9. Adding and Subtracting
2. Healing the Robot
Healing The Robot
Goals
It's time to heal the robot up to full health!
Write a function called heal() that takes amount as a parameter.
The function should add amount to health.
The robot starts with 50 health and will heal 50 to get it up to 100.
Hints
Checks
Heal Function Uses Addition
Robot Heals The Right Amount
Open Practice List
var health = 50
func heal(amount):
health += amount
Lesson complete!
But there are still some practices that you've skipped.
Stay and revisit the study material, or continue the course.
Stay
Continue
health = 100
GDQuest
Output
Run
Pause
Reset
Solution
Output
Continue

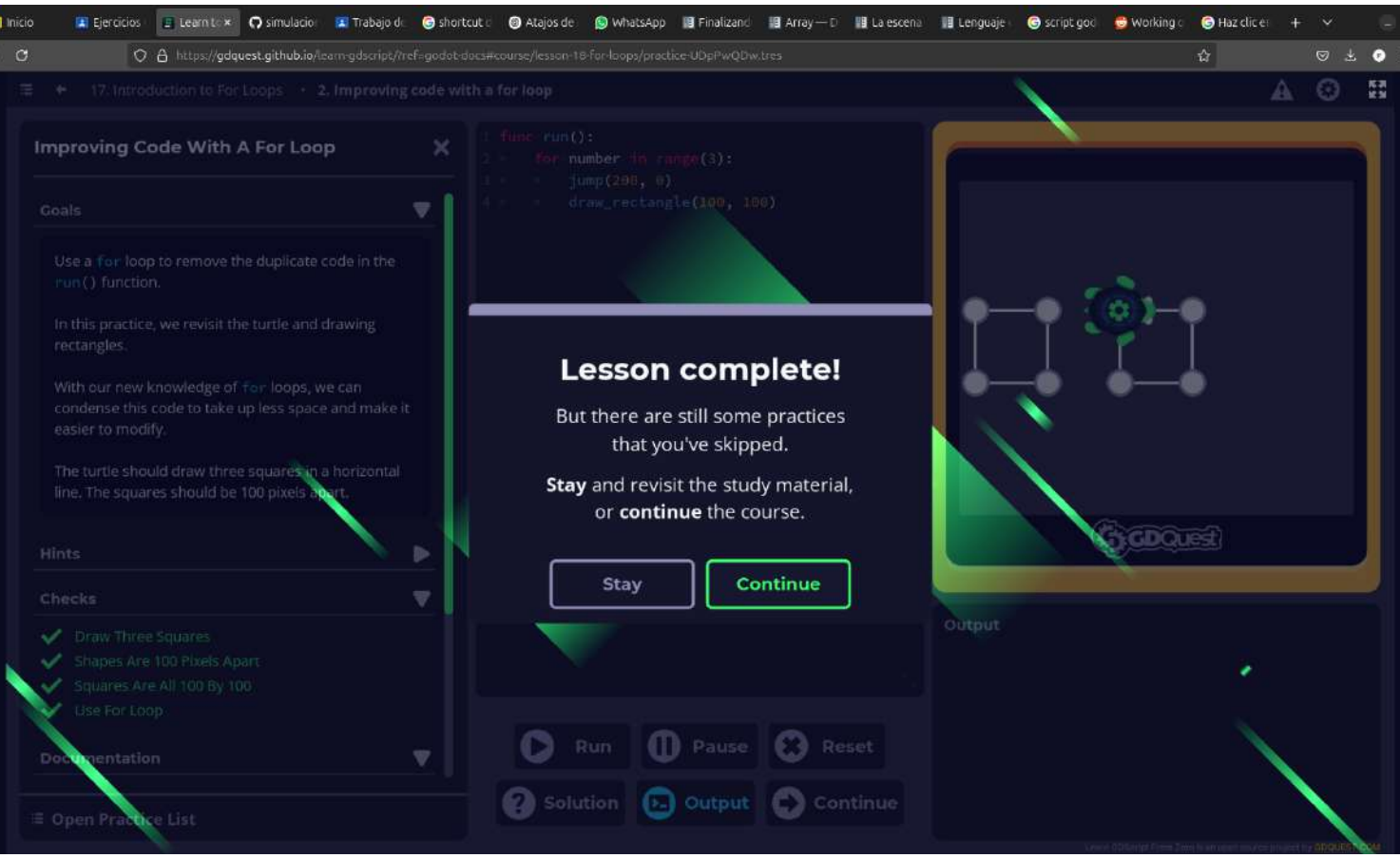

21. Functions that return a value
1. Converting coordinates from the grid to the screen
Converting Coordinates From The Grid To The Screen
Goals
Define a function that converts a position on a grid to the screen.
The function takes a Vector2 cell coordinate as an argument. It should return the corresponding Vector2 screen coordinates at the center of the cell.
Hints
Checks
Function Maps Cell To World Coordinates
Open Practice List
var cell_size = Vector2(80, 80)
func convert_to_world_coordinates(cell):
return cell * cell_size + cell_size / 2
Lesson complete!
But there are still some practices that you've skipped.
Stay and revisit the study material, or continue the course.
Stay
Continue
Run
Pause
Reset
Solution
Output
Continue
Output

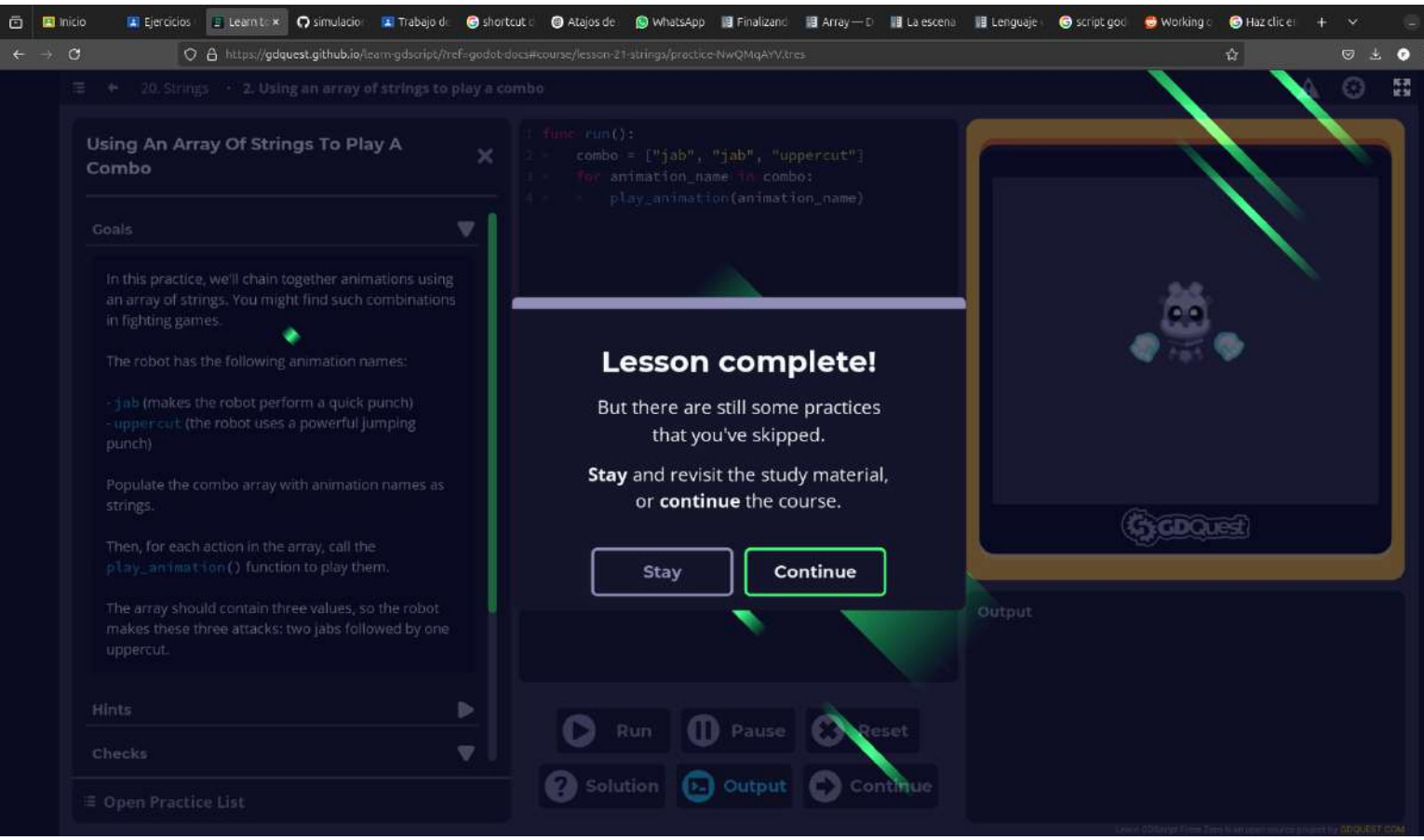

Inicio
Ejercicios
Learn to
simulacio
Trabajo d
shortcut
Atajos de
WhatsApp
Finalizand
Array — D
La escena
Lenguaje
script god
Working
Haz clic e
https://gdquest.github.io/learn-gdscript/?ref=godot-docs#course/lesson-3-standing-on-shoulders-of-giants/practice-HJMQ2XNw.tres
3. We Stand on the Shoulders of Giants
2. Make the Robot Upright
Make The Robot Upright
Goals
The robot was turned by -0.5 radians. You need to make it upright by calling the rotate() function.
Please call rotate() inside the run() function, on line 2, and keep the Tab character at the start of the line. The computer needs that to understand your code.
Hints
Checks
Character is Upright
Documentation
Method descriptions
void rotate(radians: float)
Applies a rotation to the node, in radians, starting from its current rotation.
Open Practice List
1 func run():
2 rotate(0.5)
Lesson complete!
Stay and revisit the study material, or continue the course.
Stay
Continue
Output
Run
Pause
Reset
Solution
Output
Continue

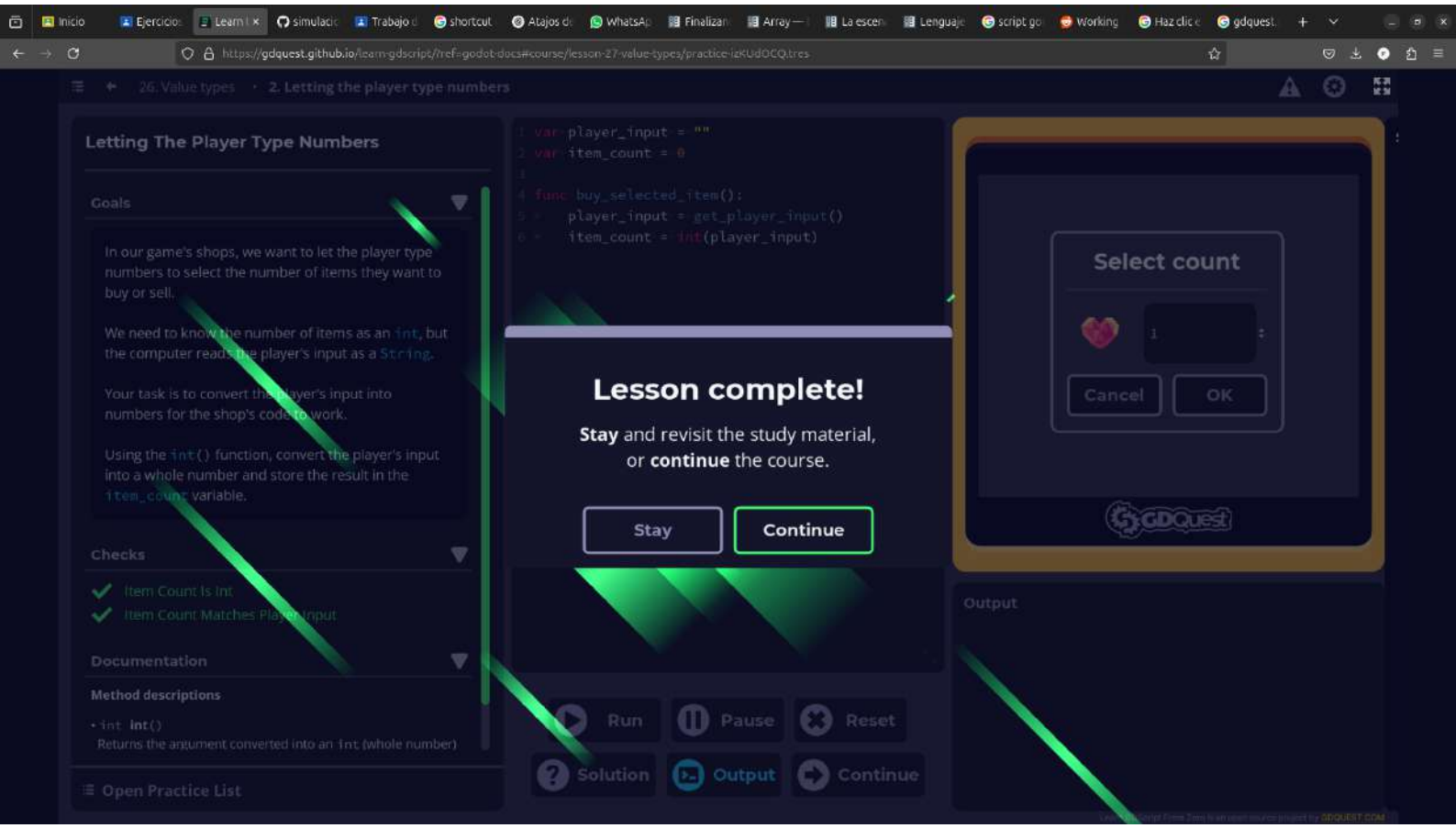

Inicio
Ejercicios
Learn to
simulacion
Trabajo de
shortcut
Atajos de
WhatsApp
Finalizand
Array — D
La escena
Lenguaje
script god
Working
Haz clic e
https://gdquest.github.io/learn-gdscript/?ref=godot-docs#course/lesson-13-conditions/practice-x2PxY8VU.tres
13. Conditions · 3. Preventing Health from Going Below Zero
Preventing Health From Going Below Zero
Goals
When the robot takes damage, its health can be negative.
We might want to display the health number on screen, like in Japanese RPGs.
We don't want negative numbers. We want to stop at zero instead.
Calling the function should reduce health by amount.
If health goes below 0, set it to 0 again.
Hints
Checks
Health Does Not Go Below Zero
Health Reaches Zero
Health Takes Different Values
Open Practice List
func take_damage(amount):
health -= amount
if health < 0:
health = 0
Lesson complete!
But there are still some practices that you've skipped.
Stay and revisit the study material, or continue the course.
Stay
Continue
health = 0
GDQuest
Output
Run
Pause
Reset
Solution
Output
Continue

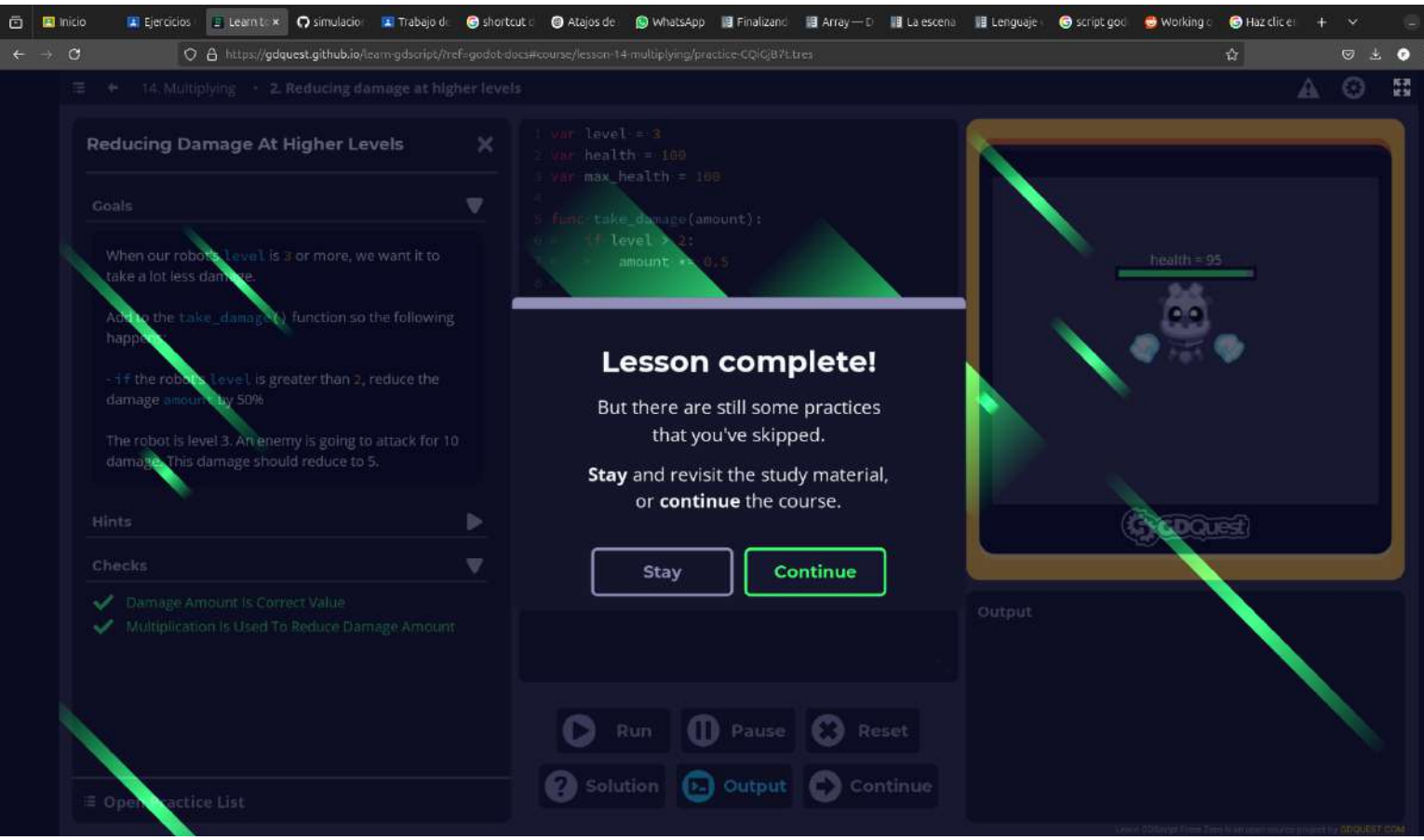

Inicio
Ejercicios
Learn to
simulacio
Trabajo d
shortcut
Atajos de
WhatsApp
Finalizand
Array — D
La escena
Lenguaje
script god
Working
Haz clic e
https://gdquest.github.io/learn-gdscript/?ref=godot-docs#course/lesson-17-while-loops/practice-lkGx0c7D.tres
16. Introduction to While Loops · 1. Moving to the end of a board
Moving To The End Of A Board
Goals
Our robot has decided to stand at the top of the board.
Complete the move_to_bottom() function so the robot moves to the bottom of the board.
The board size is determined by the Vector2 board_size.
The robot's current cell is Vector2(2, 0).
Make sure to use a while loop so the function works for any board size.
Hints
Checks
Robot Gets To Bottom Of Board
Use While Loop
Documentation
Property descriptions
Open Practice List
func move_to_bottom():
while cell.y < board_size.y - 1:
cell += Vector2(0, 1)
Lesson complete!
But there are still some practices that you've skipped.
Stay and revisit the study material, or continue the course.
Stay
Continue
GDQuest
Output
Run
Pause
Reset
Solution
Output
Continue

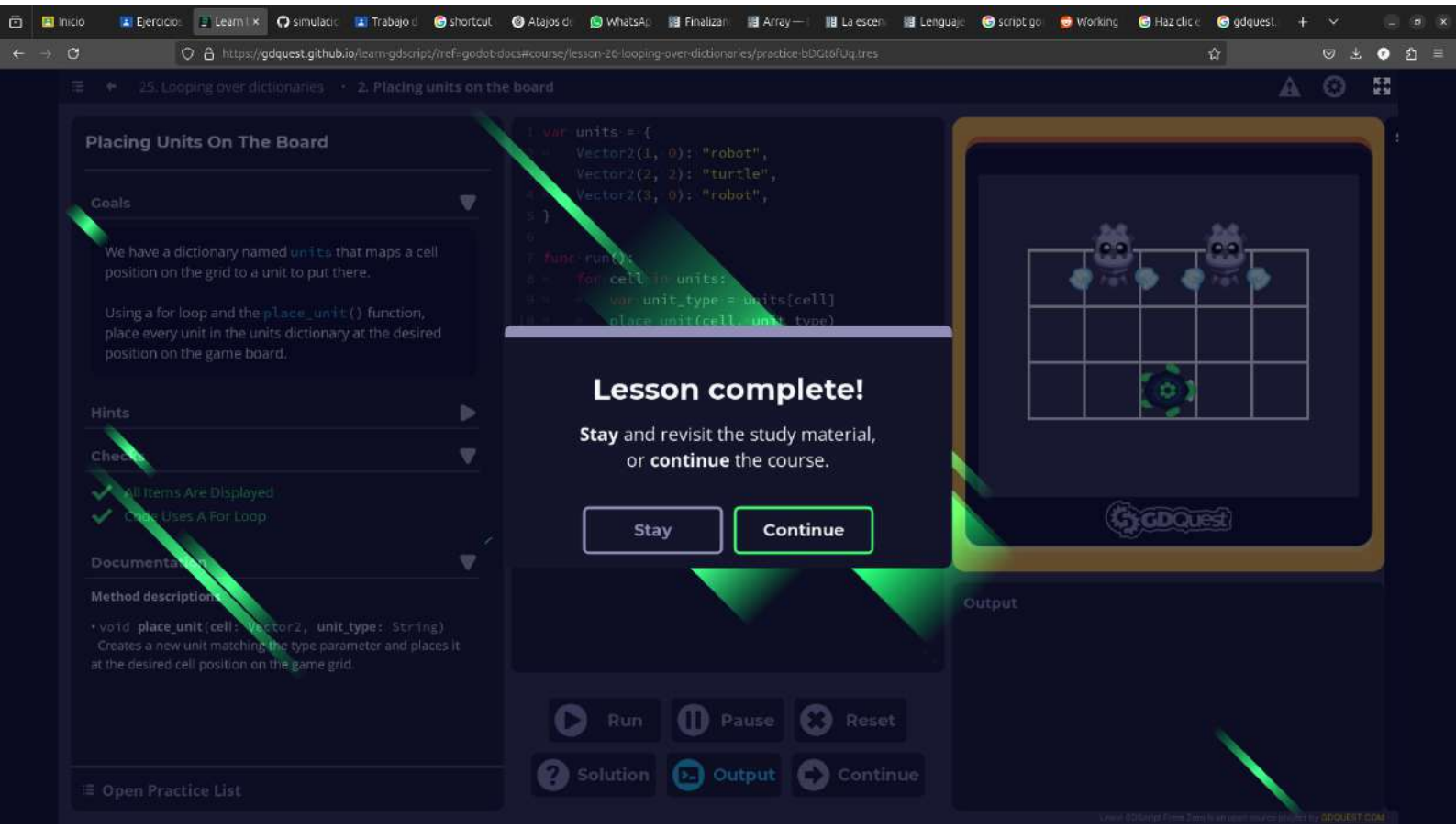

Inicio
Ejercicios
Learn to
simulacion
Trabajo de
shortcut
Atajos de
WhatsApp
Finalizand
Array — D
La escena
Lenguaje
script god
Working
Haz clic en
https://gdquest.github.io/learn-gdscript/?ref=godot-docs#course/lesson-16-2d-vectors/practice-kGx0c7DD.tres
15. 2D Vectors
2. Resetting size and position using vectors
Resetting Size And Position Using Vectors
Goals
The robot's level has increased a lot, and so has its size!
Let's fix this by resetting the robot's scale and position values.
Create a function named reset_robot() that sets the scale and position of the robot.
The x and y sub-variables of the robot's scale need to be 1.0.
The robot's position needs to be Vector2(0, 0).
As in the previous practice, make sure to use vectors when dealing with scale and position.
Hints
Checks
Robot Position Is Reset
Robot Scale Is Reset
Open Practice List
func reset_robot():
scale = Vector2(1.0, 1.0)
position = Vector2(0.0, 0.0)
Lesson complete!
But there are still some practices that you've skipped.
Stay and revisit the study material, or continue the course.
Stay
Continue
Output
Run
Pause
Reset
Solution
Output
Continue

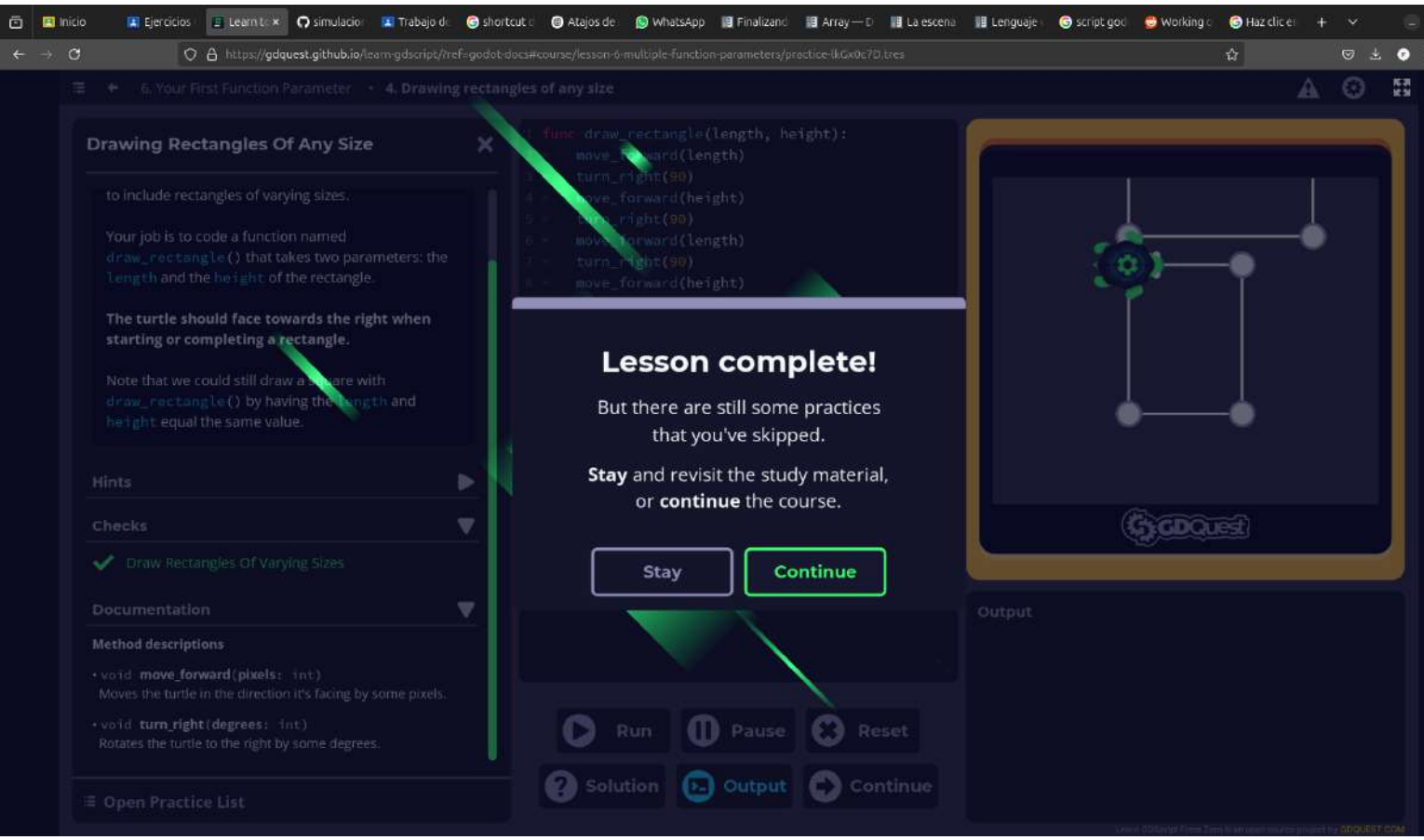

17. Introduction to For Loops
2. Improving code with a for loop
Improving Code With A For Loop
Goals
Use a for loop to remove the duplicate code in the run() function.
In this practice, we revisit the turtle and drawing rectangles.
With our new knowledge of for loops, we can condense this code to take up less space and make it easier to modify.
The turtle should draw three squares in a horizontal line. The squares should be 100 pixels apart.
Hints
Checks
Draw Three Squares
Shapes Are 100 Pixels Apart
Squares Are All 100 By 100
Use For Loop
Documentation
Open Practice List
func run():
for number in range(3):
jump(200, 0)
draw_rectangle(100, 100)
Lesson complete!
But there are still some practices that you've skipped.
Stay and revisit the study material, or continue the course.
Stay
Continue
Run
Pause
Reset
Solution
Output
Continue
Output

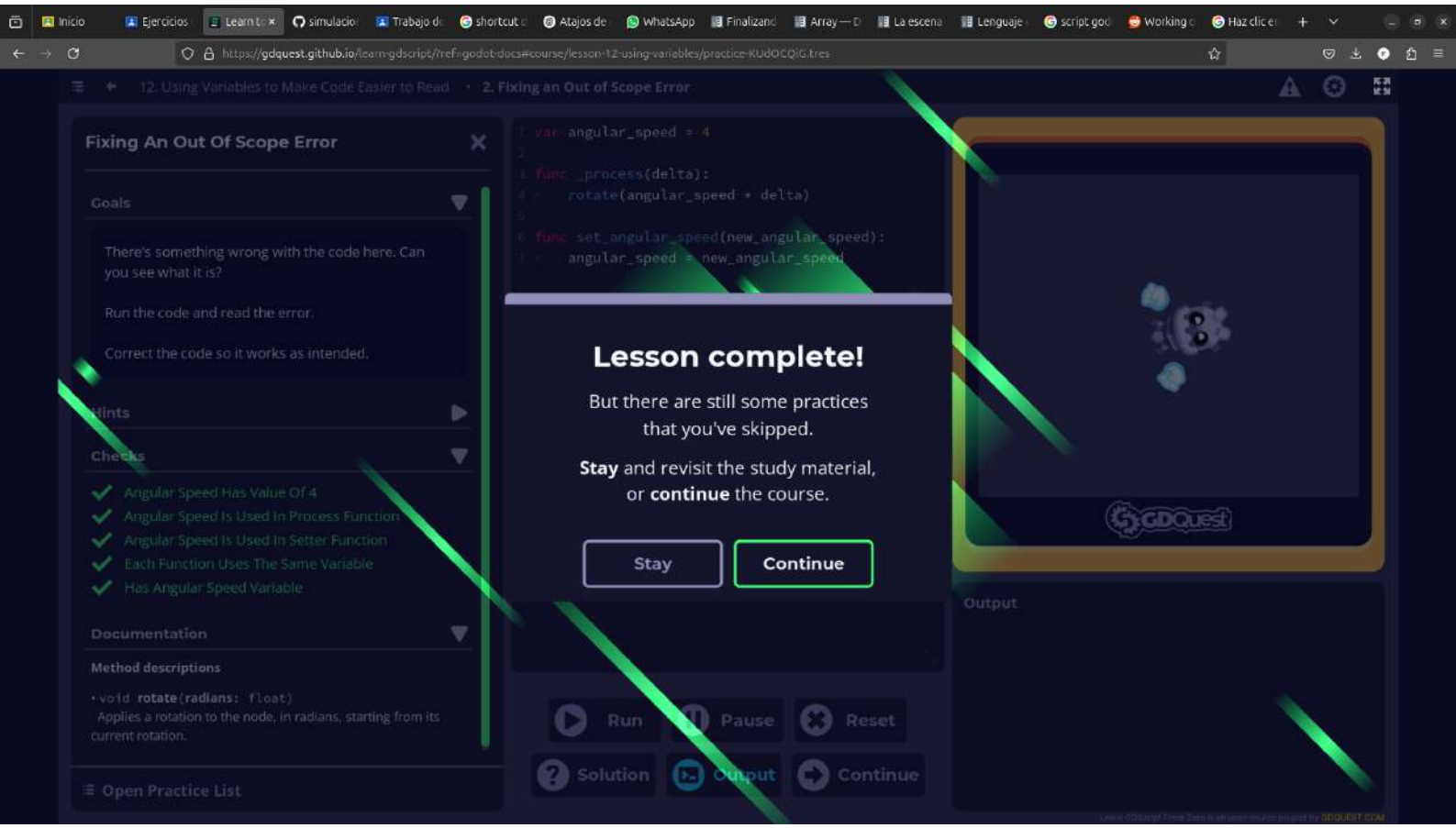

Inicio
Ejercicios
Learn to
simulacion
Trabajo de
shortcut
Atajos de
WhatsApp
Finalizand
Array — D
La escena
Lenguaje
script god
Working
Haz clic en
https://gdquest.github.io/learn-gdscript/?ref=godot-docs#course/lesson-21-strings/practice-NwQMqAYV.tres
20. Strings
2. Using an array of strings to play a combo
Using An Array Of Strings To Play A Combo
Goals
In this practice, we'll chain together animations using an array of strings. You might find such combinations in fighting games.
The robot has the following animation names:
- jab (makes the robot perform a quick punch)
- uppercut (the robot uses a powerful jumping punch)
Populate the combo array with animation names as strings.
Then, for each action in the array, call the play_animation() function to play them.
The array should contain three values, so the robot makes these three attacks: two jabs followed by one uppercut.
Hints
Checks
Open Practice List
func run():
combo = ["jab", "jab", "uppercut"]
for animation_name in combo:
play_animation(animation_name)
Lesson complete!
But there are still some practices that you've skipped.
Stay and revisit the study material, or continue the course.
Stay
Continue
GDQuest
Output
Run
Pause
Reset
Solution
Output
Continue

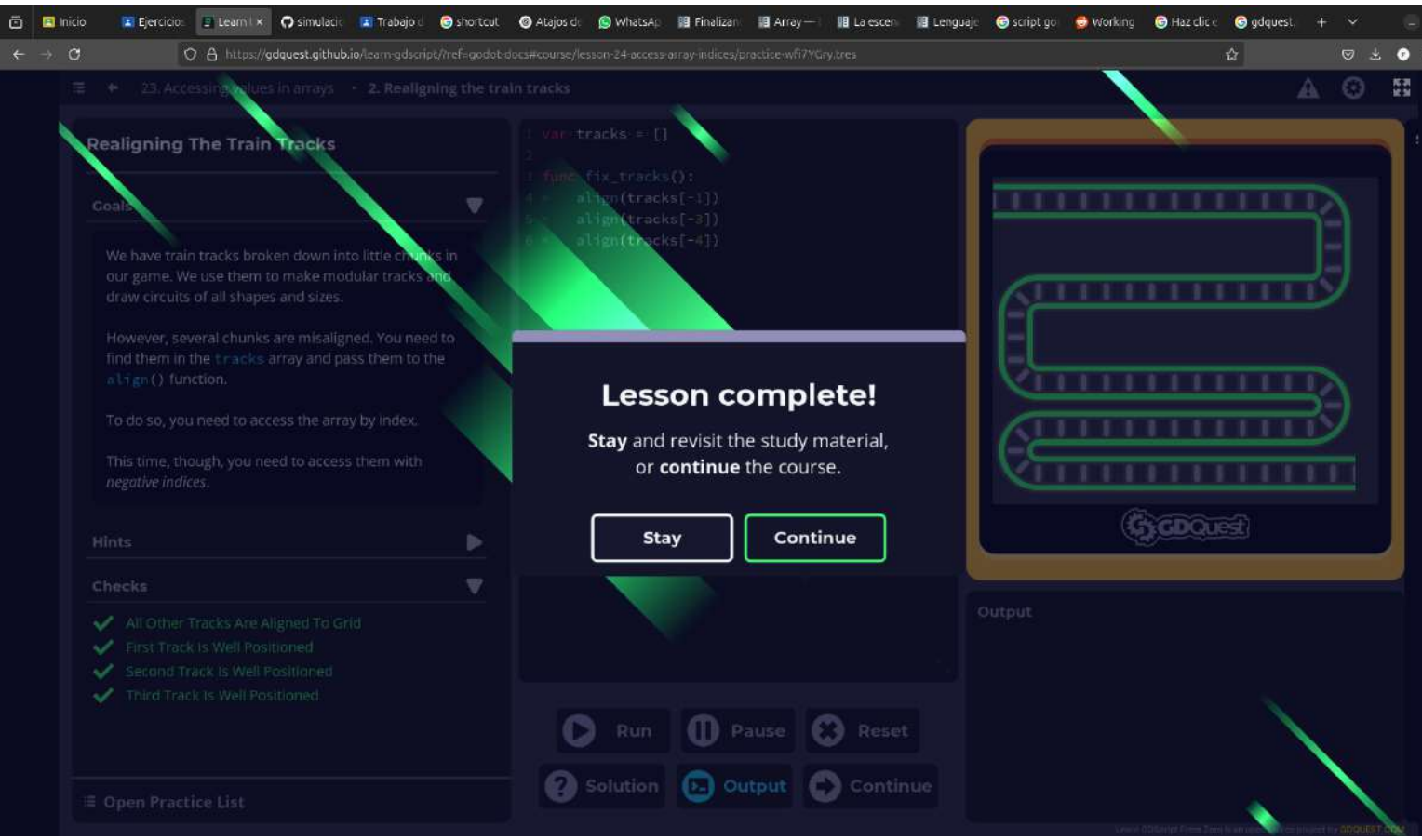

Inicio
Ejercicios
Learn
simulacio
Trabajo d
shortcut
Atajos de
WhatsAp
Finalizan
Array —
La escen
Lenguaje
script go
Working
Haz clic e
gdquest
https://gdquest.github.io/learn-gdscript/?ref=godot-docs#course/lesson-27-value-types/practice-izKUdOCQ.tres
26. Value types
2. Letting the player type numbers
Letting The Player Type Numbers
Goals
In our game's shops, we want to let the player type numbers to select the number of items they want to buy or sell.
We need to know the number of items as an int, but the computer reads the player's input as a String.
Your task is to convert the player's input into numbers for the shop's code to work.
Using the int() function, convert the player's input into a whole number and store the result in the item_count variable.
Checks
Item Count Is Int
Item Count Matches Player Input
Documentation
Method descriptions
int int()
Returns the argument converted into an int (whole number)
Open Practice List
var player_input = ""
var item_count = 0
func buy_selected_item():
player_input = get_player_input()
item_count = int(player_input)
Lesson complete!
Stay and revisit the study material, or continue the course.
Stay
Continue
Select count
1
Cancel
OK
GDQuest
Output
Run
Pause
Reset
Solution
Output
Continue

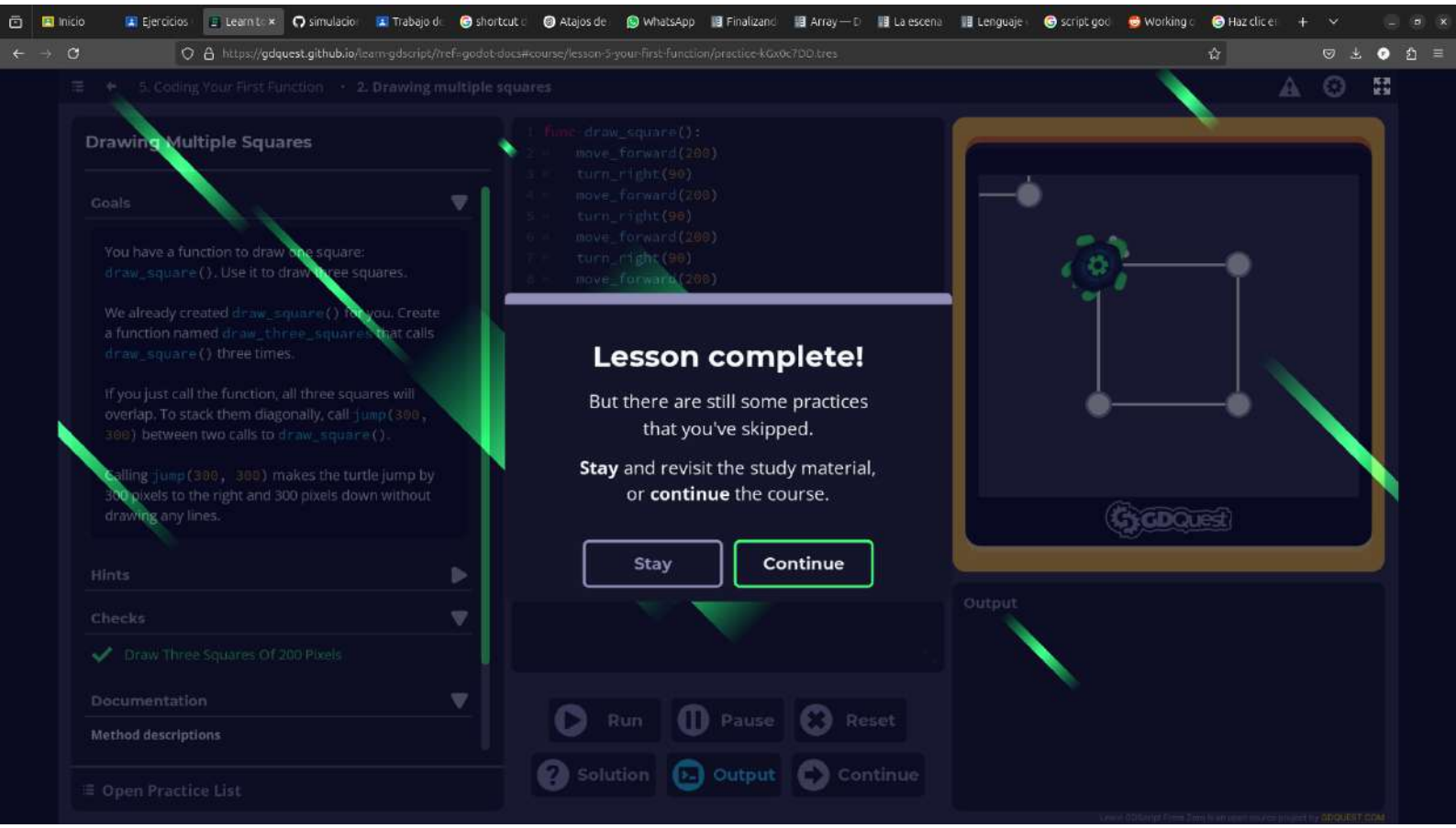

Inicio
Ejercicios
Learn to
simulacion
Trabajo de
shortcut
Atajos de
WhatsApp
Finalizand
Array — D
La escena
Lenguaje
script god
Working
Haz clic e
https://gdquest.github.io/learn-gdscript/?ref=godot-docs#course/lesson-14-multiplying/practice-CQiGJB7t.tres
14. Multiplying
2. Reducing damage at higher levels
Reducing Damage At Higher Levels
Goals
When our robot's level is 3 or more, we want it to take a lot less damage.
Add to the take_damage() function so the following happens:
- if the robot's level is greater than 2, reduce the damage amount by 50%
The robot is level 3. An enemy is going to attack for 10 damage. This damage should reduce to 5.
Hints
Checks
Damage Amount Is Correct Value
Multiplication Is Used To Reduce Damage Amount
Open Practice List
var level = 3
var health = 100
var max_health = 100
func take_damage(amount):
if level > 2:
amount *= 0.5
Lesson complete!
But there are still some practices that you've skipped.
Stay and revisit the study material, or continue the course.
Stay
Continue
health = 95
GDQuest
Output
Run
Pause
Reset
Solution
Output
Continue

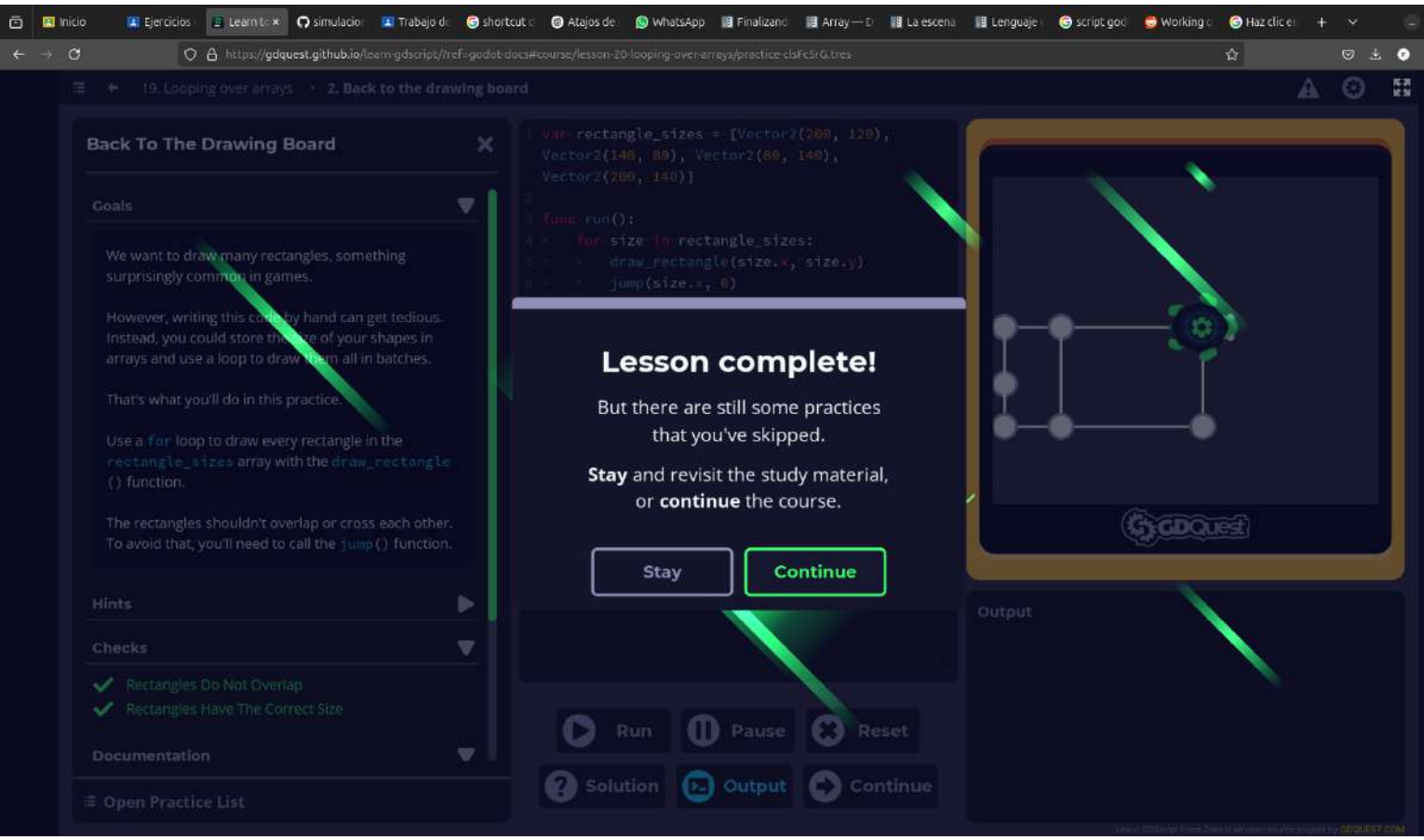

Inicio
Ejercicios
Learn
simulacio
Trabajo d
shortcut
Atajos de
WhatsAp
Finalizan
Array —
La escen
Lenguaje
script go
Working
Haz clic e
gdquest
https://gdquest.github.io/learn-gdscript/?ref=godot-docs#course/lesson-26-looping-over-dictionaries/practice-bDGt6fUq.tres
25. Looping over dictionaries
2. Placing units on the board
Placing Units On The Board
Goals
We have a dictionary named units that maps a cell position on the grid to a unit to put there.
Using a for loop and the place_unit() function, place every unit in the units dictionary at the desired position on the game board.
Hints
Checks
All Items Are Displayed
Code Uses A For Loop
Documentation
Method description:
void place_unit(cell: Vector2, unit_type: String)
Creates a new unit matching the type parameter and places it at the desired cell position on the game grid.
Open Practice List
var units = {
Vector2(1, 0): "robot",
Vector2(2, 2): "turtle",
Vector2(3, 0): "robot",
}
func run():
for cell in units:
var unit_type = units[cell]
place_unit(cell, unit_type)
Lesson complete!
Stay and revisit the study material, or continue the course.
Stay
Continue
GDQuest
Output
Run
Pause
Reset
Solution
Output
Continue

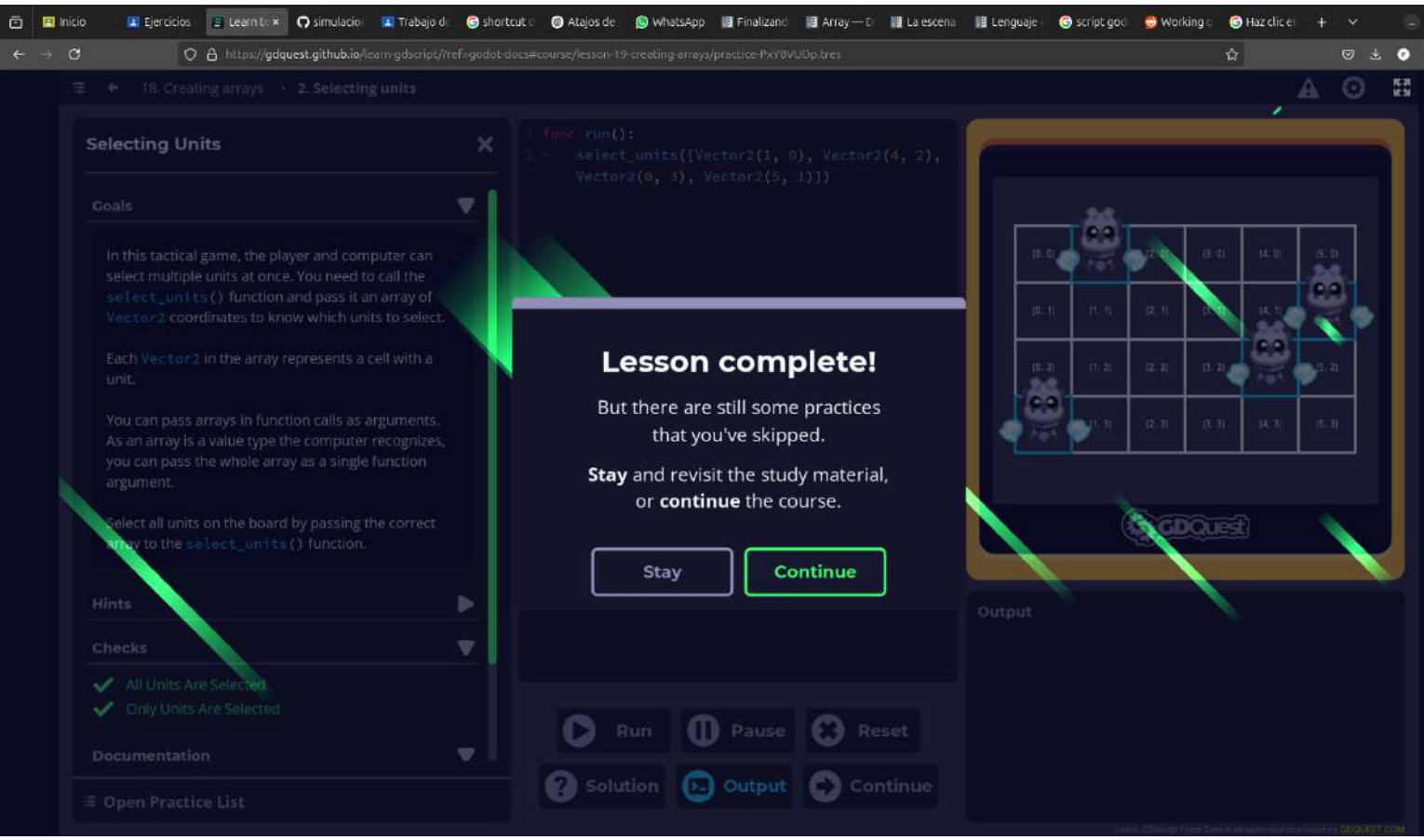

Inicio
Ejercicios
Learn to
simulacion
Trabajo de
shortcut
Atajos de
WhatsApp
Finalizand
Array — D
La escena
Lenguaje
script god
Working
Haz clic en
https://gdquest.github.io/learn-gdscript/?ref=godot-docs#course/lesson-6-multiple-function-parameters/practice-IkGx0c7D.tres
6. Your First Function Parameter
4. Drawing rectangles of any size
Drawing Rectangles Of Any Size
to include rectangles of varying sizes.
Your job is to code a function named draw_rectangle() that takes two parameters: the length and the height of the rectangle.
The turtle should face towards the right when starting or completing a rectangle.
Note that we could still draw a square with draw_rectangle() by having the length and height equal the same value.
Hints
Checks
Draw Rectangles Of Varying Sizes
Documentation
Method descriptions
• void move_forward(pixels: int)
Moves the turtle in the direction it's facing by some pixels.
• void turn_right(degrees: int)
Rotates the turtle to the right by some degrees.
Open Practice List
func draw_rectangle(length, height):
move_forward(length)
turn_right(90)
move_forward(height)
turn_right(90)
move_forward(length)
turn_right(90)
move_forward(height)
Lesson complete!
But there are still some practices that you've skipped.
Stay and revisit the study material, or continue the course.
Stay
Continue
Run
Pause
Reset
Solution
Output
Continue
Output


12. Using Variables to Make Code Easier to Read
2. Fixing an Out of Scope Error
Fixing An Out Of Scope Error
Goals
There's something wrong with the code here. Can you see what it is?
Run the code and read the error.
Correct the code so it works as intended.
Hints
Checks
Angular Speed Has Value Of 4
Angular Speed Is Used In Process Function
Angular Speed Is Used In Setter Function
Each Function Uses The Same Variable
Has Angular Speed Variable
Documentation
Method descriptions
void rotate(radians: float)
Applies a rotation to the node, in radians, starting from its current rotation.
Open Practice List
var angular_speed = 4
func _process(delta):
rotate(angular_speed * delta)
func set_angular_speed(new_angular_speed):
angular_speed = new_angular_speed
Lesson complete!
But there are still some practices that you've skipped.
Stay and revisit the study material, or continue the course.
Stay
Continue
Run
Pause
Reset
Solution
Output
Continue
Output


Realigning The Train Tracks
Goals
We have train tracks broken down into little chunks in our game. We use them to make modular tracks and draw circuits of all shapes and sizes.
However, several chunks are misaligned. You need to find them in the tracks array and pass them to the align() function.
To do so, you need to access the array by index.
This time, though, you need to access them with negative indices.
Hints
Checks
All Other Tracks Are Aligned To Grid
First Track Is Well Positioned
Second Track Is Well Positioned
Third Track Is Well Positioned
Open Practice List
var tracks = []
func fix_tracks():
align(tracks[-1])
align(tracks[-3])
align(tracks[-4])
Lesson complete!
Stay and revisit the study material, or continue the course.
Stay
Continue
Run
Pause
Reset
Solution
Output
Continue
Output


Inicio
Ejercicios
Learn to
simulacion
Trabajo de
shortcut
Atajos de
WhatsApp
Finalizand
Array — D
La escena
Lenguaje
script god
Working
Haz clic en
https://gdquest.github.io/learn-gdscript/?ref=godot-docs#course/lesson-5-your-first-function/practice-kGx0c7DD.tres
5. Coding Your First Function
2. Drawing multiple squares
Drawing Multiple Squares
Goals
You have a function to draw one square: draw_square(). Use it to draw three squares.
We already created draw_square() for you. Create a function named draw_three_squares that calls draw_square() three times.
If you just call the function, all three squares will overlap. To stack them diagonally, call jump(300, 300) between two calls to draw_square().
Calling jump(300, 300) makes the turtle jump by 300 pixels to the right and 300 pixels down without drawing any lines.
Hints
Checks
Draw Three Squares Of 200 Pixels
Documentation
Method descriptions
Open Practice List
func draw_square():
move_forward(200)
turn_right(90)
move_forward(200)
turn_right(90)
move_forward(200)
turn_right(90)
move_forward(200)
Lesson complete!
But there are still some practices that you've skipped.
Stay and revisit the study material, or continue the course.
Stay
Continue
GDQuest
Output
Run
Pause
Reset
Solution
Output
Continue


Inicio
Ejercicios
Learn to
simulacio
Trabajo d
shortcut
Atajos de
WhatsApp
Finalizand
Array — D
La escena
Lenguaje
script god
Working
Haz clic e
https://gdquest.github.io/learn-gdscript/?ref=godot-docs#course/lesson-20-looping-over-arrays/practice-clsFc5rG.tres
19. Looping over arrays
2. Back to the drawing board
Back To The Drawing Board
Goals
We want to draw many rectangles, something surprisingly common in games.
However, writing this code by hand can get tedious. Instead, you could store the size of your shapes in arrays and use a loop to draw them all in batches.
That's what you'll do in this practice.
Use a for loop to draw every rectangle in the rectangle_sizes array with the draw_rectangle() function.
The rectangles shouldn't overlap or cross each other. To avoid that, you'll need to call the jump() function.
Hints
Checks
Rectangles Do Not Overlap
Rectangles Have The Correct Size
Documentation
Open Practice List
var rectangle_sizes = [Vector2(200, 120), Vector2(140, 80), Vector2(80, 140), Vector2(200, 140)]
func run():
for size in rectangle_sizes:
draw_rectangle(size.x, size.y)
jump(size.x, 0)
Lesson complete!
But there are still some practices that you've skipped.
Stay and revisit the study material, or continue the course.
Stay
Continue
GDQuest
Output
Run
Pause
Reset
Solution
Output
Continue


Inicio
Ejercicios
Learn to
simulacio
Trabajo d
shortcut
Atajos de
WhatsApp
Finalizand
Array — D
La escena
Lenguaje
script god
Working
Haz clic e
https://gdquest.github.io/learn-gdscript/?ref=godot-docs#course/lesson-19-creating-arrays/practice-PxY8vUDp.tres
18. Creating arrays
2. Selecting units
Selecting Units
Goals
In this tactical game, the player and computer can select multiple units at once. You need to call the select_units() function and pass it an array of Vector2 coordinates to know which units to select.
Each Vector2 in the array represents a cell with a unit.
You can pass arrays in function calls as arguments. As an array is a value type the computer recognizes, you can pass the whole array as a single function argument.
Select all units on the board by passing the correct array to the select_units() function.
Hints
Checks
All Units Are Selected
Only Units Are Selected
Documentation
Open Practice List
func run():
select_units([Vector2(1, 0), Vector2(4, 2), Vector2(0, 3), Vector2(5, 1)])
Lesson complete!
But there are still some practices that you've skipped.
Stay and revisit the study material, or continue the course.
Stay
Continue
Run
Pause
Reset
Solution
Output
Continue
Output
GDQuest