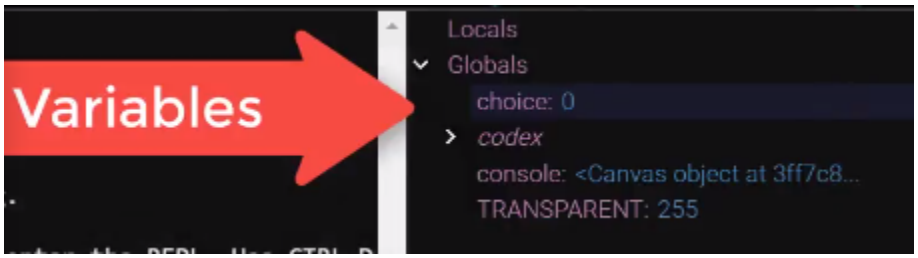
CodeBot Python Code By Mission

Mission 4 – Animatronics (Objectives 1-5)	
Infinite loop	`while True:`
Updating a variable	`n_led = n_led + 1`
Use debugger to view variables	 Open the console panel while debugging

Reset a variable to stay within a range	`n_led = n_led + 1`
	`if n_led == 8:`
	`    n_led = 0`

Break out of a loop	`break`
Increment	`n_guests = n_guests + 1` `count = count + 1`
Turn on LED using a variable	`leds.ls_num(n_guests, True)`
Mission 4 – Animatronics (Objectives 6-12)	
Play a tone on the speaker	`spkr.pitch(440)` `sleep(0.1)` the (argument) is the pitch frequency
Turn off the speaker	`spkr.off()`
Debounce a button press	`buttons.was_pressed(0)`
While loop	`while count < 10:` (will iterate, or repeat, 10 times if count starts at 0)
Import random library	`from random import randrange`
Get a random number within a range	`f = randrange(100, 1000)`

Define a function	`def flashLEDs():`
	`    leds.user(0b11111111)`
	`    sleep(0.5)`
	`    leds.user(0b00000000)`
	`    sleep(0.5)`
	
	`# Function to play a note`
	`def note(freq, duration):`
	`    spkr.pitch(freq)`
	`    sleep(duration)`
	`    spkr.off()`
	`    sleep(0.05)`

Call a function	`flashLEDs()` `note(F4, 0.4)`
Mission 5 - Fence Patrol	
Read a line sensor	`ls.read(num)  # Sensor 'num' can be 0, 1, 2, 3, or 4` `val = ls.read(n)` (returns a value between 0 and 4095)
Display the value of a variable in the console	`print(val)` `print("Line sensor value = ", val)`

Assign a Boolean result of a comparison to a variable
 	
Use the Boolean variable in code
	`threshold = 2500`
	`is_detected = val < threshold`
	`leds.ls_num(0, is_detected)`

Detection	Dark line on light surface – use val > threshold Light line on dark surface – use val < threshold

Use a comparison with a while loop and use the control variable as an argument in a function call	`n = 0`
	`while n < 5:`
	`    detect_line(n)`
	`    n = n + 1`

Wait loop
(safe driving)
	`while True:`
	`    if buttons.was_pressed(0):`
	`        break`

Return statement	`return is_detected` `return got_line`
Call to a function that has a return	`hit = scan_lines()` `if detect_line(count):`
Use a variable to turn on LEDs	`leds.user(line_count)` line_count will be from 0 to 255

Wrap-around the line_count variable for binary numbers