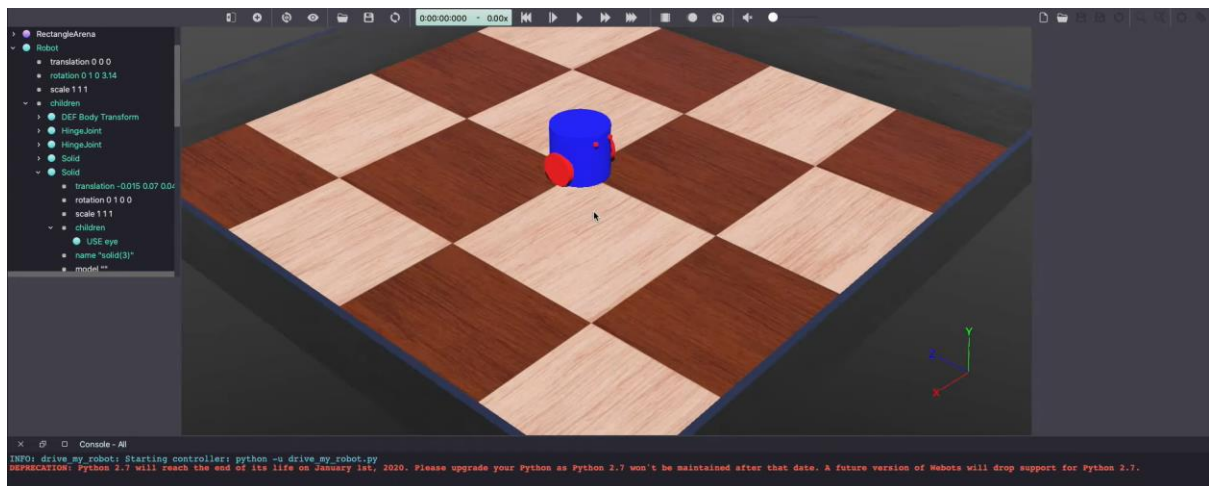


The tutorial is an introduction to Webots, a robotic simulation software

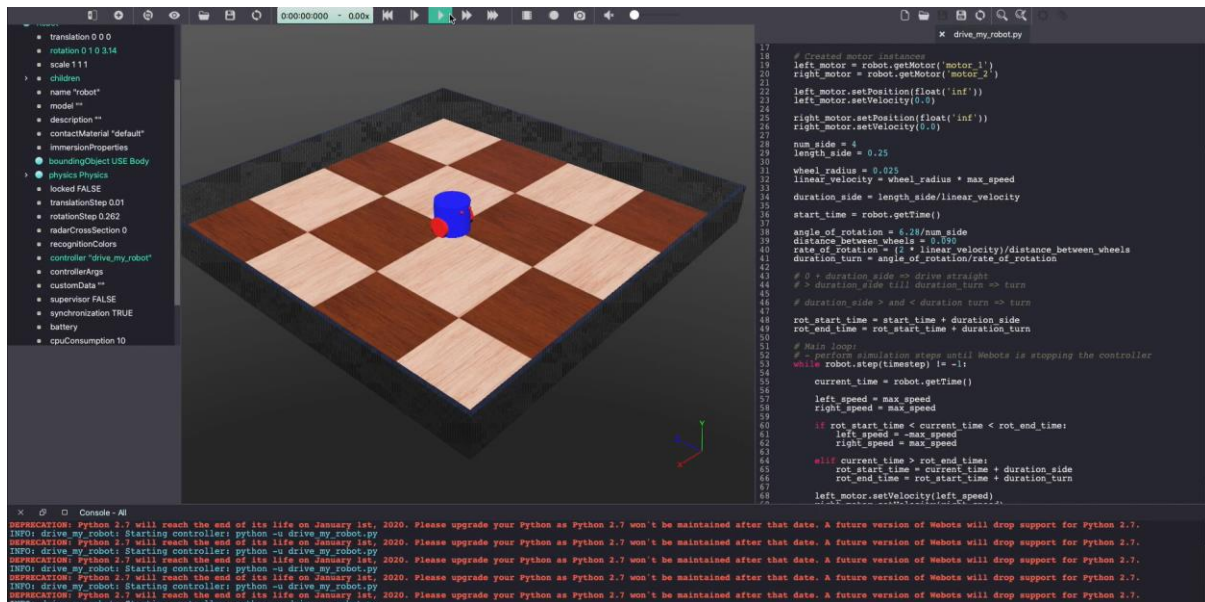
The tutorial covers the download, installation, and basic concepts of using Webots, providing specific timings for each step. Kajal demonstrates the installation process on Ubuntu, including opening a terminal and typing "webots" to start the simulation.

The tutorial walks through creating a new project, modifying the world with a rectangular arena, and adjusting floor size. Kajal explains how to create a custom object (a sphere) in Webots, adjusting its appearance, defining a bounding object, and adding physics. The tutorial also briefly touches on running a simulation with a pre-existing robot (e-puck) and editing its controller code written in C.



The tutorial is about designing a 2-wheel differential drive robot in Webots, presented.

The tutorial begins with creating a new project named "diff drive robot" with a rectangular arena in Webots. The robot's main body is created using a cylinder shape with specific dimensions and a blue color appearance. Two wheels are added using hinge joints, each with a rotational motor to allow turning. The wheels are positioned, and their appearances are customized (red color). The tutorial also covers adding eyes to the robot using solid nodes and box shapes, with specific translations and appearances. The presenter mentions that in the next tutorial, they will cover writing a controller to make the robot move forward, turn in place, and turn while driving forward.



The tutorial is part of a series on Webots and focuses on writing a controller code for a custom two-wheel differential drive robot. This video demonstrates how to create a new robot controller in Python, set up basic code structure, and instantiate motors for the left and right wheels.

The code includes a main loop where the robot's behavior is defined, setting motor velocities for driving straight and turning. The tutorial explores the concept of differential drive for turning the robot. The presenter introduces the idea of driving the robot in a square or polygon shape by calculating the time for driving straight and turning at each corner. The tutorial concludes by mentioning the limitations of an open-loop system and hints at the possibility of exploring a closed-loop system in future tutorials. Viewers are encouraged to like, subscribe, and leave questions or comments.