"""week1 controller."""

# You may need to import some classes of the controller module. Ex:

# from controller import Robot, Motor, DistanceSensor

from controller import Robot

# create the Robot instance.

robot = Robot()

# get the time step of the current world.

timestep = int(robot.getBasicTimeStep())

# You should insert a getDevice-like function in order to get the

# instance of a device of the robot. Something like:

# motor = robot.getDevice('motorname')

# ds = robot.getDevice('dsname')

# ds.enable(timestep)

# Main loop:

# - perform simulation steps until Webots is stopping the controller

while robot.step(timestep) != -1:

# Read the sensors:

# Enter here functions to read sensor data, like:

# val = ds.getValue()

# Process sensor data here.

# Enter here functions to send actuator commands, like:

# motor.setPosition(10.0)

pass

# Enter here exit cleanup code. """week1 controller."""

# You may need to import some classes of the controller module. Ex:

# from controller import Robot, Motor, DistanceSensor

from controller import Robot

# create the Robot instance.

robot = Robot()

# get the time step of the current world.

timestep = int(robot.getBasicTimeStep())

# You should insert a getDevice-like function in order to get the

# instance of a device of the robot. Something like:

# motor = robot.getDevice('motorname')

# ds = robot.getDevice('dsname')

# ds.enable(timestep)

# Main loop:

# - perform simulation steps until Webots is stopping the controller

while robot.step(timestep) != -1:

# Read the sensors:

# Enter here functions to read sensor data, like:

# val = ds.getValue()

# Process sensor data here.

# Enter here functions to send actuator commands, like:

# motor.setPosition(10.0)

pass

# Enter here exit cleanup code. """week1 controller."""

# You may need to import some classes of the controller module. Ex:

# from controller import Robot, Motor, DistanceSensor

from controller import Robot

# create the Robot instance.

robot = Robot()

# get the time step of the current world.

timestep = int(robot.getBasicTimeStep())

# You should insert a getDevice-like function in order to get the

# instance of a device of the robot. Something like:

# motor = robot.getDevice('motorname')

# ds = robot.getDevice('dsname')

# ds.enable(timestep)

# Main loop:

# - perform simulation steps until Webots is stopping the controller

while robot.step(timestep) != -1:

# Read the sensors:

# Enter here functions to read sensor data, like:

# val = ds.getValue()

# Process sensor data here.

# Enter here functions to send actuator commands, like:

# motor.setPosition(10.0)

pass

# Enter here exit cleanup code.