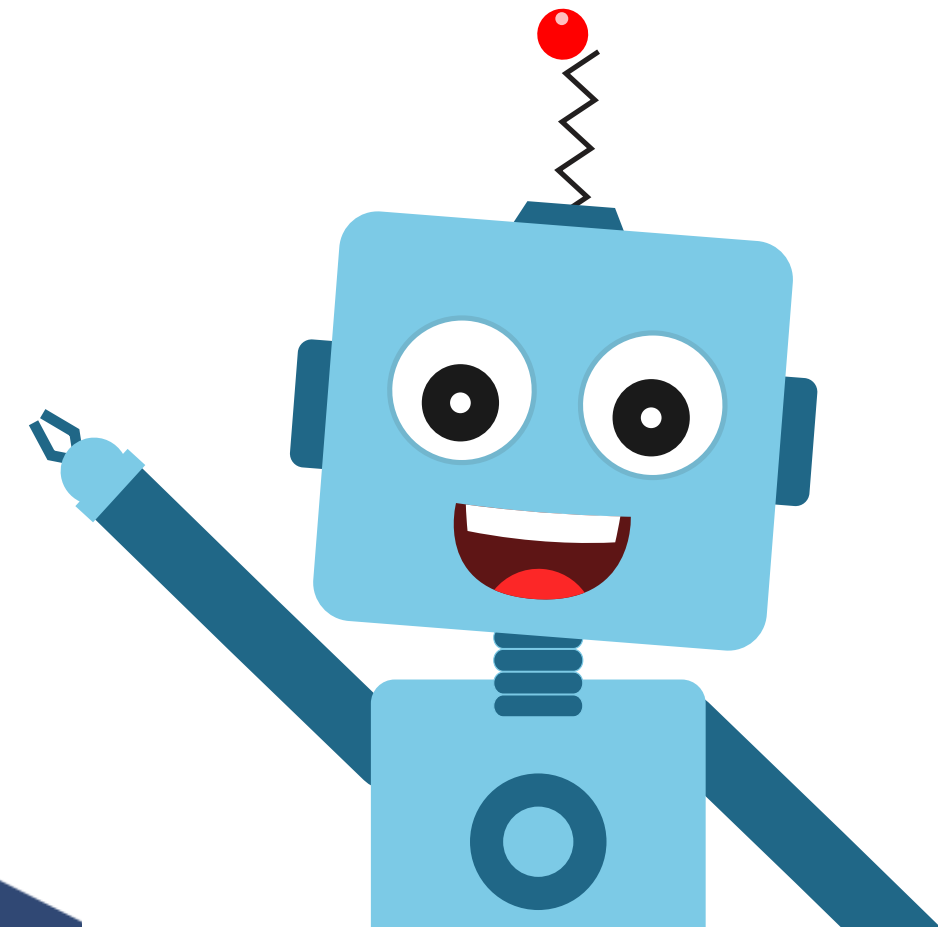


# Obstacle Avoidance Robot-Part 2

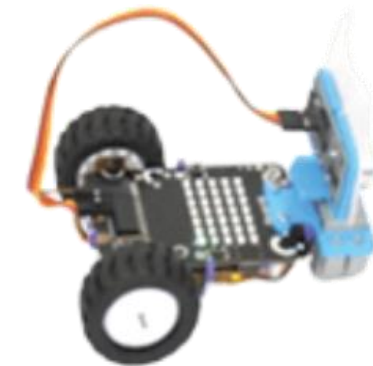
Session 30



# Topics covered

- In this activity, you will learn about will learn what an Obstacle Avoidance Robot is, how to assemble it, and the logic and flowchart behind the robot.
- Now we all know what an obstacle avoidance robot is and how it works. But we still need to learn how to implement the logic in code. Don't worry because we are going to cover it in this lesson.

**Turn to the  
direction where  
distance > 10cm**





# Activity: Obstacle Avoidance Robot



# Activity: Obstacle Avoidance Robot

- Declare two variables called Max and Min from the Variables.
- We are setting the maximum and minimum distances to 30 cm and 5 cm respectively.

```
Min = 10;
```

```
Max = 30;
```

- Go to the Sensors and connect the Ultrasonic to initialize the pins. The next step would be to set the initial servo angle as 90 degrees.
- To introduce a time delay of some seconds, use the `time.sleep()` function.
- A while loop is employed so that our code can execute repeatedly.
- If an obstacle comes at a defined distance. The ultrasonic sensor will detect the obstacle and stop.
- Next, using the servo motor the robot will scan the environment. But first, it will turn the servo to 45 and check for free area obstacles.

```
servo22.write(45)
```

```
time.sleep(0.5)
```

# Activity: Obstacle Avoidance Robot

- If there is no obstacle in a range of 30 cm at the 45-degrees angle. Quarky will take a right turn by 45 degrees.

```
if (quarky.getdistance(1) > Max):  
    quarky.stoprobot()
```

- If an obstacle is detected on the right side at the 45-degrees angle. Change the angle to 135 degrees. Robot will move Right with 50% speed.
- If there is no obstacle in a range of 30 cm at the 135-degrees angle. Quarky will take a left turn by 45 degrees. And run in left with 50% speed.

```
if (quarky.getdistance(1) > Max):  
    servo22.write(45)  
    quarky.runrobot("RIGHT",50)
```

- If an obstacle is detected on the right side at the 135-degrees angle. Change the angle to 0 degrees.

# Activity: Obstacle Avoidance Robot

- If there is no obstacle in a range of 30 cm at the 0-degrees angle. Quarky will take a right turn by 90 degrees. Robot will turn right with 50% speed.
- If an obstacle is detected on the right side at the 0-degrees angle. Change the angle to 180 degrees.
- If there is no obstacle in a range of 30 cm at the 180-degrees angle. Quarky will take a left turn by 90 degrees. And robot will turn right with 50% speed.
- If an obstacle is detected on the right side at a 180-degrees angle, then take the reverse.
- Press run button to run the code.

# Activity: Obstacle Avoidance Robot

```
from quarky import *  
  
import time  
  
servo22 = Servo(22)  
  
quarky.setultrasonicpins(1,18,19)  
  
Min = 10;  
  
Max = 30;  
  
servo22.write(45)  
  
time.sleep(0.5)  
  
while True:  
  
    if (quarky.getdistance(1) < Min):  
  
        quarky.stoprobot()  
  
        servo22.write(45)  
  
        time.sleep(0.5)  
  
        if (quarky.getdistance(1) > Max):  
  
            quarky.runrobot("RIGHT",50)  
  
            time.sleep(0.6)  
  
            quarky.stoprobot()
```

# Activity: Obstacle Avoidance Robot

else:

```
servo22.write(135)
```

```
time.sleep(0.5)
```

```
if (quarky.getdistance(1) > Max):
```

```
    quarky.runrobot("LEFT",50)
```

```
    time.sleep(0.6)
```

```
    quarky.stoprobot()
```

else:

```
servo22.write(0)
```

```
time.sleep(0.5)
```

```
if (quarky.getdistance(1) > Max):
```

```
    quarky.runrobot("RIGHT",50)
```

```
    time.sleep(1.2)
```

```
    quarky.stoprobot()
```



# Activity: Obstacle Avoidance Robot

else:

```
servo22.write(180)
```

```
time.sleep(0.5)
```

```
if (quarky.getdistance(1) > Max):
```

```
    quarky.runrobot("RIGHT",50)
```

```
    time.sleep(1.2)
```

```
    quarky.stoprobot()
```

```
else:
```

```
    quarky.runrobot("LEFT",50)
```

```
    time.sleep(1.5)
```

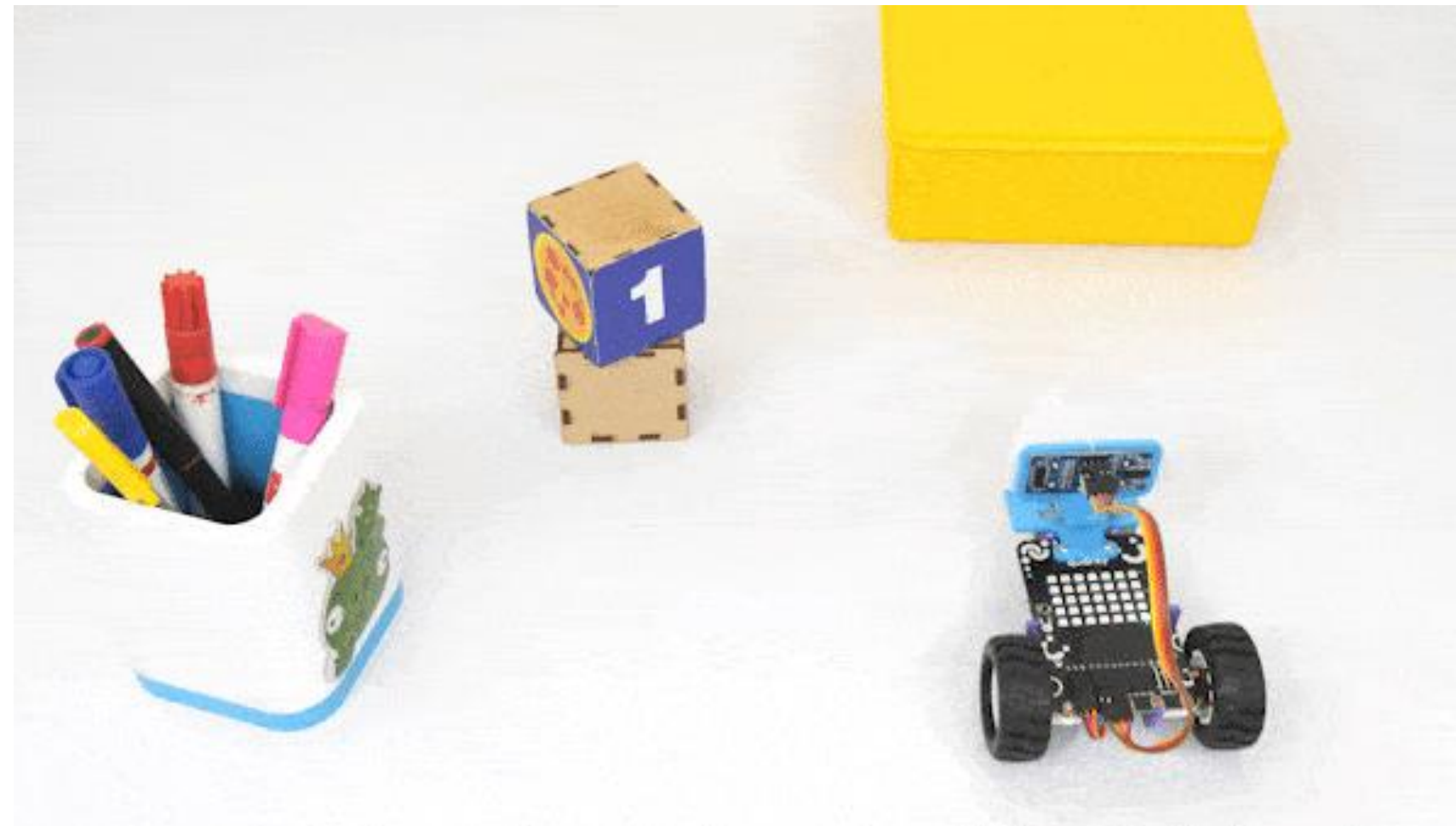
```
    quarky.stoprobot()
```

```
servo22.write(90)
```

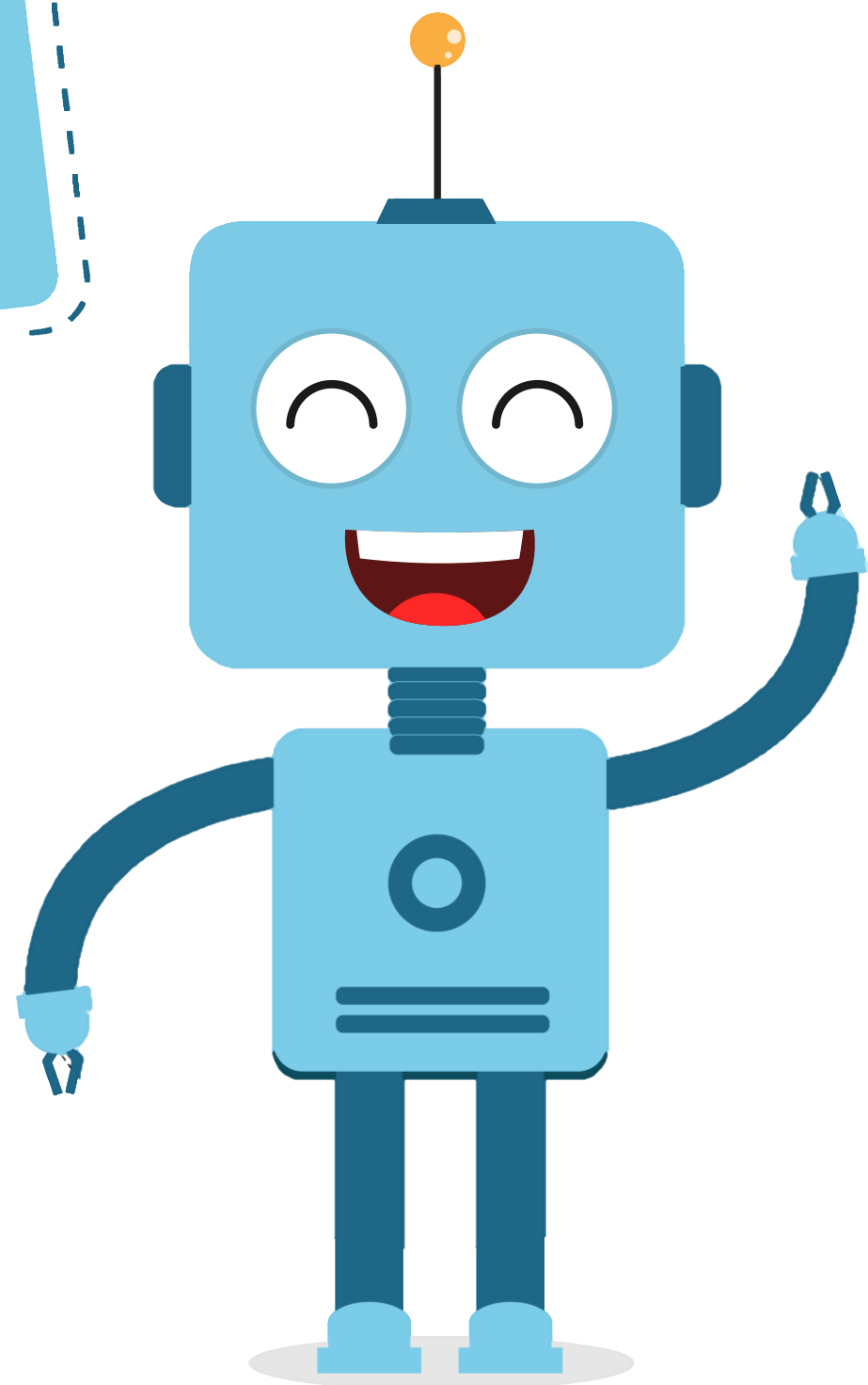
```
quarky.runrobot("FORWARD",50)
```

# Final Output

Note: Proper indentation is crucial in Python code to avoid errors, so it's important to refer to the activity sheet for correct code.



THANK  
YOU



POWERED BY  
**STEMpedia**