

ROS2_DAY5(HW1)

```
WallFollower::WallFollower() : Node("wall_follower")
{
    this->declare_parameter<double>("left_distance_to_wall", 0.65); // 왼쪽 벽
    this->declare_parameter<double>("front_distance_to_wall", 3.1); // 전방 벽
    subscription_ = this->create_subscription<sensor_msgs::msg::LaserScan>(
        "/scan", 10, std::bind(&WallFollower::scan_callback, this, std::placeholders::_1));
    publisher_ = this->create_publisher<geometry_msgs::msg::Twist>("/cmd_vel")
}
```

- **파라미터 선언:**

- "left_distance_to_wall" 와 "front_distance_to_wall" 이라는 파라미터를 각각 0.65와 3.1로 선언하여, 왼쪽 벽과 전방 벽까지의 목표 거리를 설정한다.

- **구독자 및 퍼블리셔 생성:**

- /scan 토픽에서 sensor_msgs::msg::LaserScan 메시지를 구독하여 라이다 센서의 데이터를 수신한다.
- /cmd_vel 토픽에 geometry_msgs::msg::Twist 메시지를 퍼블리시하여, 터틀봇의 속도 명령을 전달한다.

```
// 전방이 inf이거나 목표 거리보다 멀 때 오른쪽으로 회전
if (std::isinf(front_distance) || front_distance > front_distance_to_wall)
{
    cmd_vel.linear.x = 0.0; // 정지
    cmd_vel.angular.z = -0.5; // 오른쪽으로 회전
}
else // 벽을 따라가는 기본 동작
{
    if (left_distance < left_distance_to_wall) // 왼쪽 벽과 너무 가까울 때
    {
        cmd_vel.linear.x = 0.0; // 정지
        cmd_vel.angular.z = -0.2; // 천천히 오른쪽 회전
    }
    else
    {
        cmd_vel.linear.x = 0.1; // 전진
        cmd_vel.angular.z = 0.0; // 직진
    }
}
```

- **전방이 inf이거나 목표 거리보다 멀 때:**

- 전방 거리가 무한대이거나 설정된 front_distance_to_wall 보다 멀 경우, 터틀봇은 오른쪽으로 회전한다.

- 이를 위해 `cmd_vel.linear.x = 0.0` 으로 설정해 정지하고, `cmd_vel.angular.z = -0.5` 로 설정하여 시계방향 (오른쪽) 회전을 한다.
- **벽을 따라 이동할 때:**
 - 왼쪽 벽과 너무 가까운 경우:
 - 왼쪽 거리가 `left_distance_to_wall` 보다 짧으면, 터틀봇은 오른쪽으로 천천히 회전(`angular.z = -0.2`) 을 한다.
 - 왼쪽 벽과 적당한 거리를 유지하는 경우:
 - 왼쪽 거리가 `left_distance_to_wall` 보다 크거나 같으면, 터틀봇은 직진(`linear.x = 0.1`)하여 벽을 따라 이동한다.