



Lecture 4. 로봇의 구동



Outline



- ▶ 메시지 자료형
- ▶ 로봇의 속도
- ▶ TurtleBot 패키지 설치
- ▶ 로봇의 구동



기본 메시지 자료형



Primitive Type	Serialization	C++	Python2	Python3
bool	unsigned 8-bit int	uint8_t	bool	bool
int8	signed 8-bit int	int8_t	int	int
uint8	unsigned 8-bit int	uint8_t	int	int
int16	signed 16-bit int	int16_t	int	int
uint16	unsigned 16-bit int	uint16_t	int	int
int32	signed 32-bit int	int32_t	int	int
uint32	unsigned 32-bit int	uint32_t	int	int
int64	signed 64-bit int	int64_t	long	int
uint64	unsigned 64-bit int	uint64_t	long	int
float32	32-bit IEEE float	float	float	float
float64	64-bit IEEE float	double	float	float
string	ascii string	std::string	str	bytes
time	secs/nsecs unsigned 32-bit ints	ros::Time	rospy.Time	rospy.Time
duration	secs/nsecs signed 32-bit ints	ros::Duration	rospy.Duration	rospy.Duration



추가 메시지 자료형



▶ 기본 자료형 패키지: std_msgs

```
user@hostname$ rosmmsg package std_msgs
```

...

결과 넣기

▶ 센서 관련 자료형 패키지: sensor_msgs

```
user@hostname$ rosmmsg package sensor_msgs
```

...

결과 넣기

▶ 사용자 정의

- ▶ 필요한 메시지 자료형을 정의할 수 있음



메시지 데이터



- ▶ 한 메시지 안에 다양한 데이터를 넣을 수 있음

```
geometry_msgs/Vector3.msg  
float64 x  
float64 y  
float64 z
```

```
geometry_msgs/Twist.msg  
Vector3 linear  
Vector3 angular
```

- ▶ Python에서 class처럼 사용

```
cmd = Twist()  
cmd.linear.x = 0.5  
cmd.angular.z = 0.2;
```

Twist msg {

Vector3 linear	{	float64 x
		float64 y
		float64 z
Vector3 angular	{	float64 x
		float64 y
		float64 z



Outline



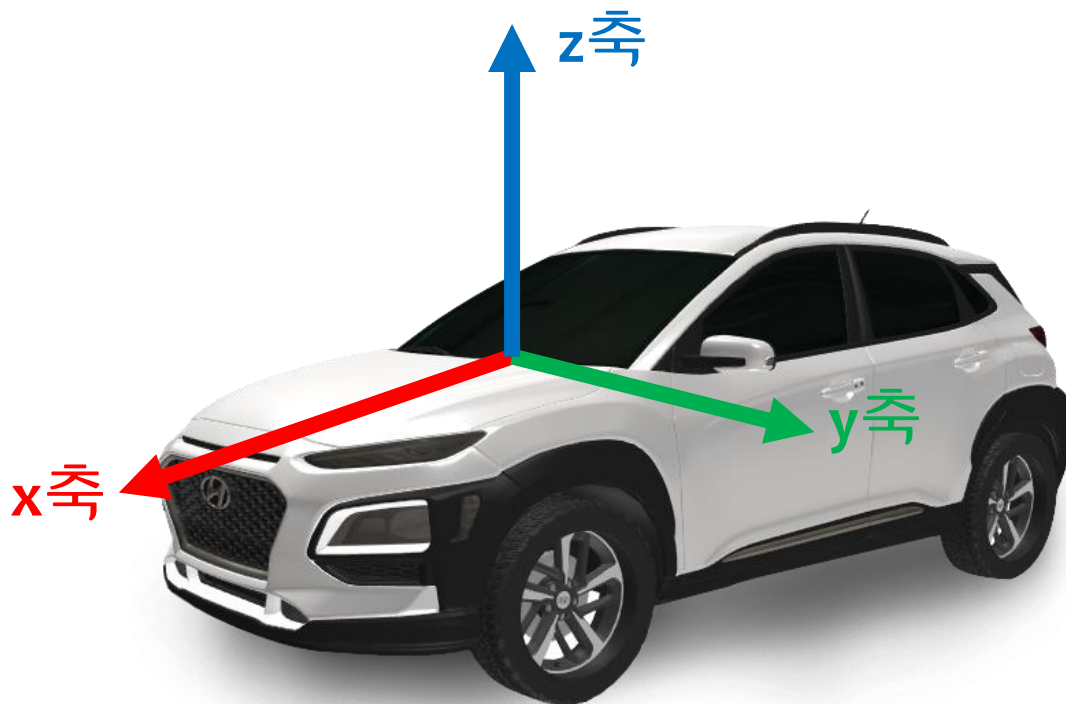
- ▶ 메시지 자료형
- ▶ 로봇의 속도
- ▶ TurtleBot 패키지 설치
- ▶ 로봇의 구동



선형 속도 (Linear Velocity)



- ▶ 직선으로 운동하는 속도
 - ▶ 3차원 공간이므로 세 방향의 속도



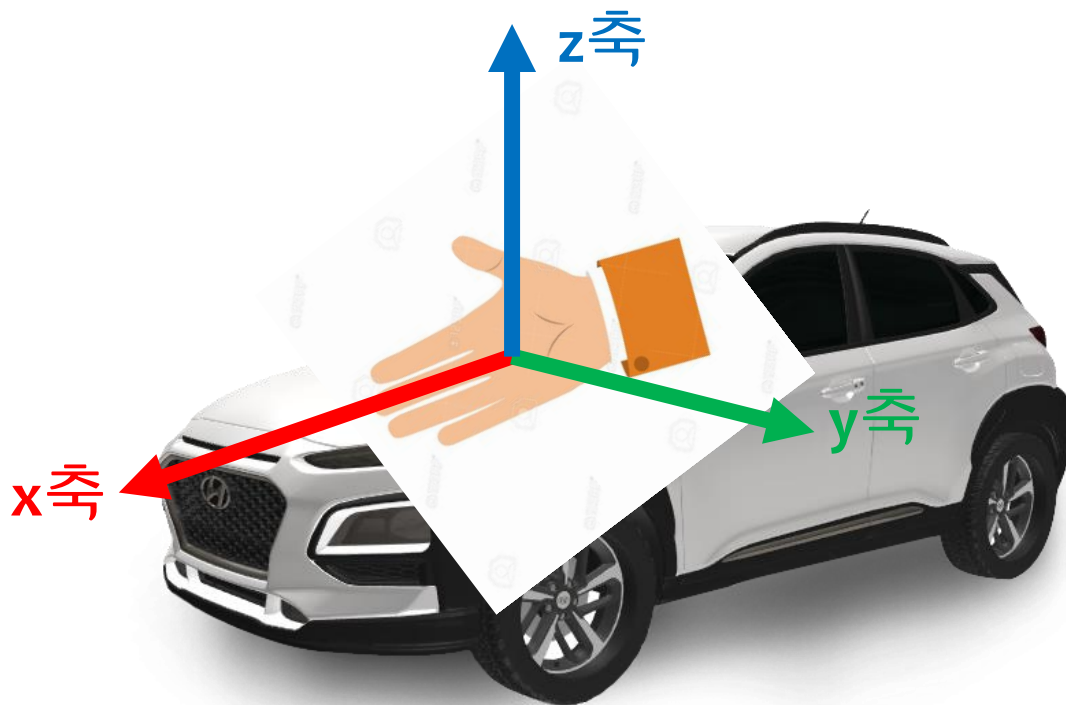
x축: 전진하는 방향
y축: x 축에 수직으로
왼쪽 방향
z축: x 축에 수직으로
위쪽 방향



선형 속도 (Linear Velocity)



- ▶ 직선으로 운동하는 속도
 - ▶ 3차원 공간이므로 세 방향의 속도



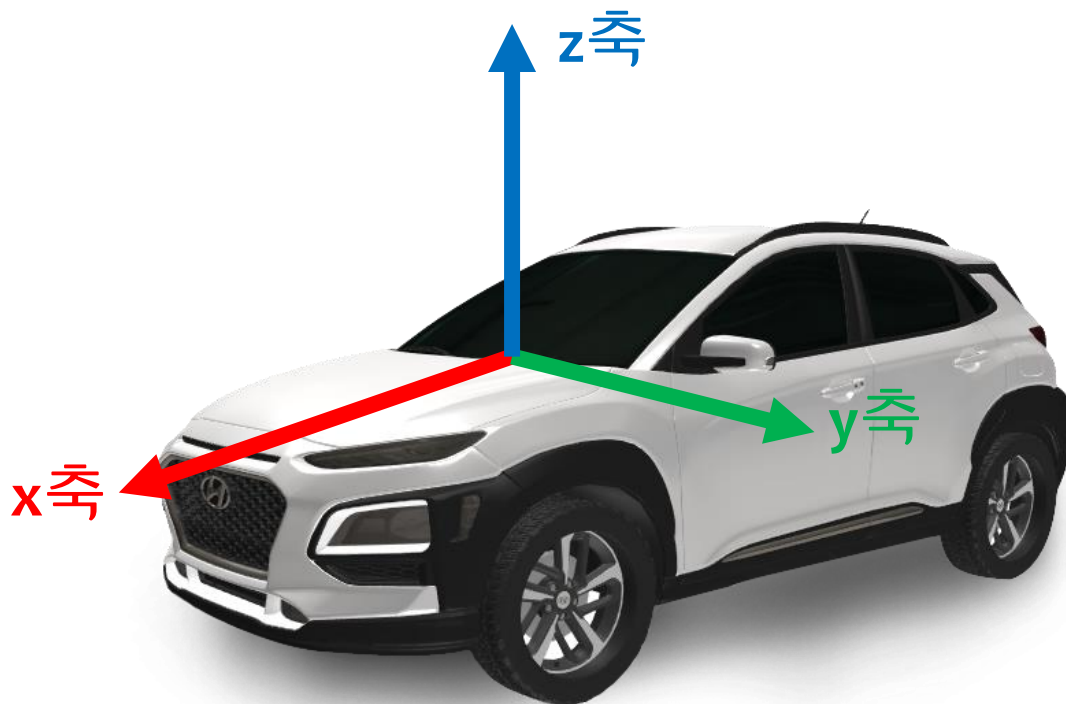
오른 손을 편 상태로
앞을 가리켰다면
손바닥이 가리키는
방향이 y축 방향
엄지 손가락이 가리키는
방향이 z축 방향



선형 속도 (Linear Velocity)



- ▶ 직선으로 운동하는 속도
 - ▶ 3차원 공간이므로 세 방향의 속도



x축: 전진하는 방향
y축: x 축에 수직으로
왼쪽 방향
z축: x 축에 수직으로
위쪽 방향

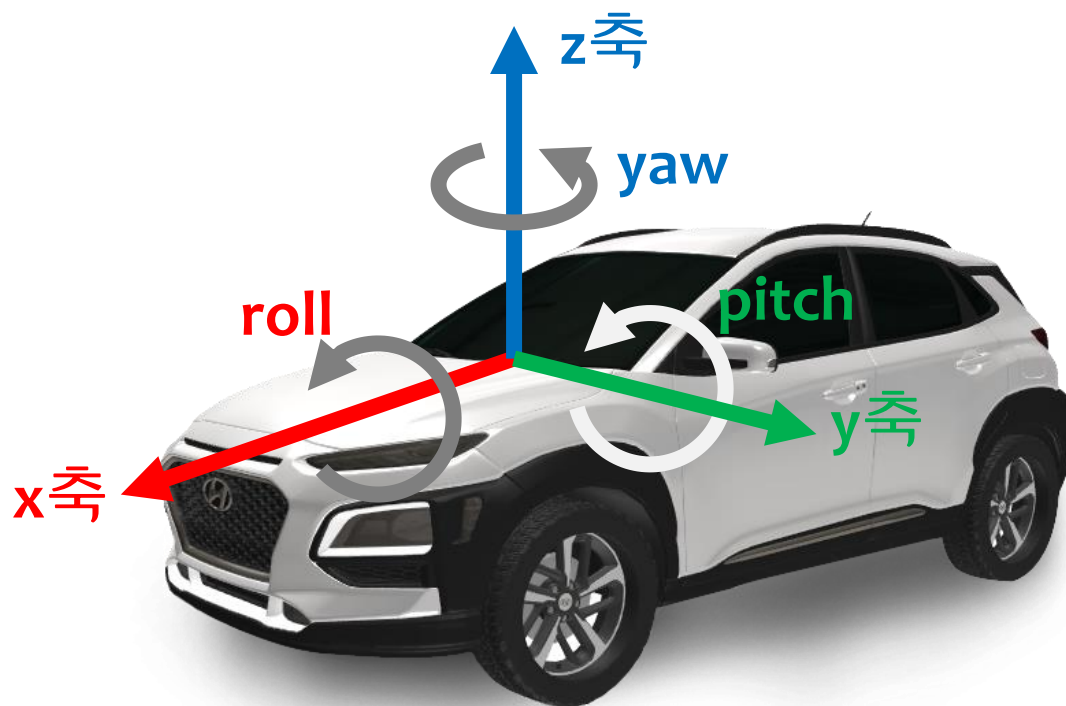
실제로는 의미 없음



각속도 (Angular Velocity)



- ▶ 회전 운동하는 속도
 - ▶ 3차원 공간이므로 세 방향의 속도



실제로는 의미 없음

roll: x축을 중심으로 회전

pitch: y축을 중심으로 회전

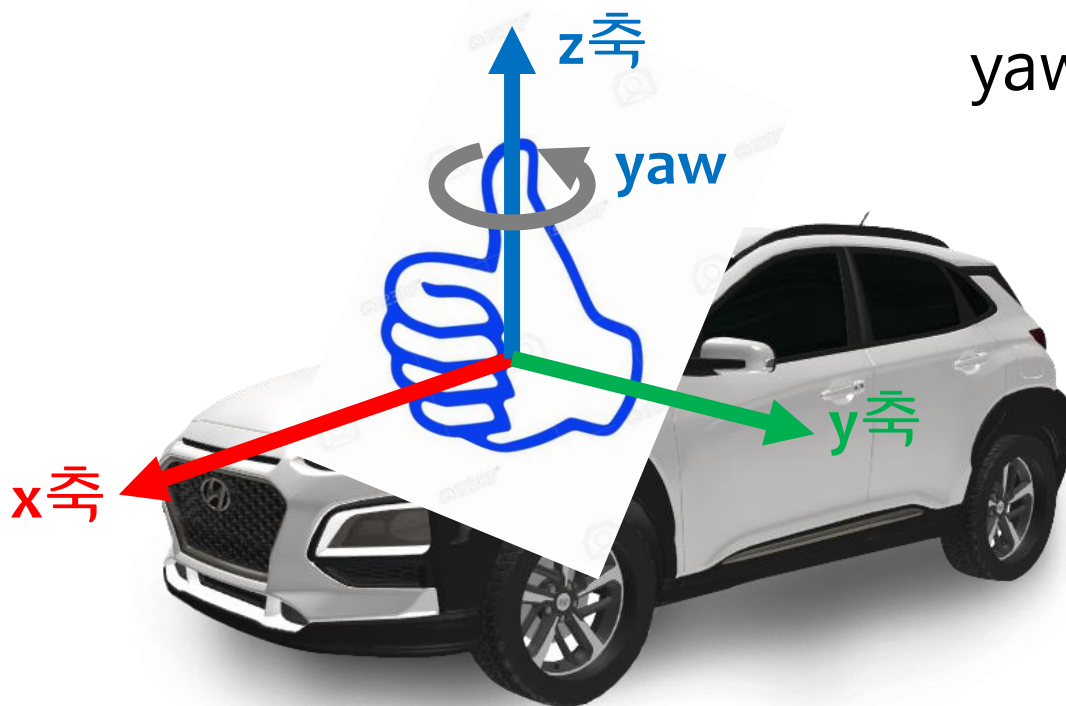
yaw: z축을 중심으로 회전



각속도 (Angular Velocity)



- ▶ 회전 운동하는 속도
 - ▶ 3차원 공간이므로 세 방향의 속도



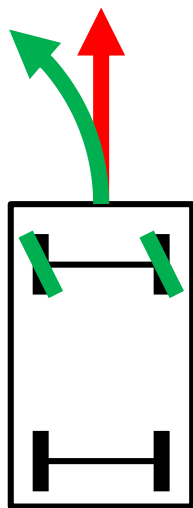
yaw: z축을 중심으로 회전
오른손 엄지를 위로 향했을
때 나머지 손가락을 따라
회전하는 방향



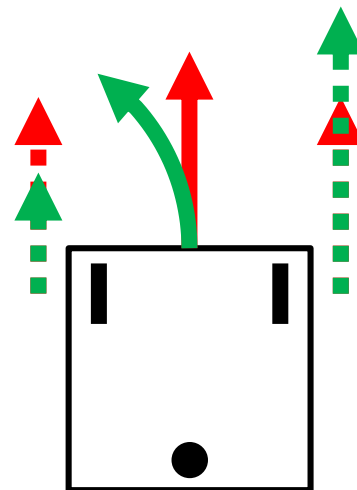
회전 방법



- ▶ 일반적인 자동차 (Car-like steering, Ackermann steering)



- ▶ 차동 구동(Differential drive)





회전 방법



▶ 일반적인 자동차 (Car-like steering, Ackermann steering)

- ▶ 제자리 회전 불가능
- ▶ 효율적

▶ 차동 구동(Differential drive)

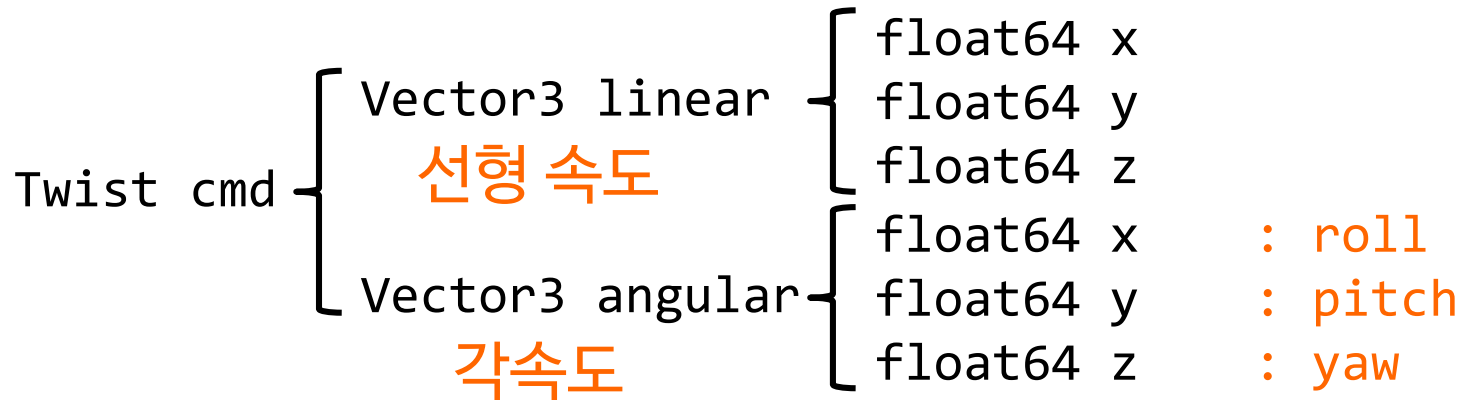
- ▶ 제자리 회전 가능
- ▶ 회전시 바퀴가 미끄러짐



ROS에서 속도 지정



- ▶ ROS에서 속도를 지정할 때는 Twist 자료형 사용



```
cmd = Twist()
```

```
cmd.linear.x = 0.5 -----> 0.5m/s 전진
```

```
cmd.angular.z = 0.2; -----> 0.2rad/s 반시계방향 회전  
(좌회전)
```

180° = 3.1416 rad이므로 0.2rad/s로 16초 동안
회전하면 3.2rad를 회전해서 대략 반 바퀴 회전



Outline



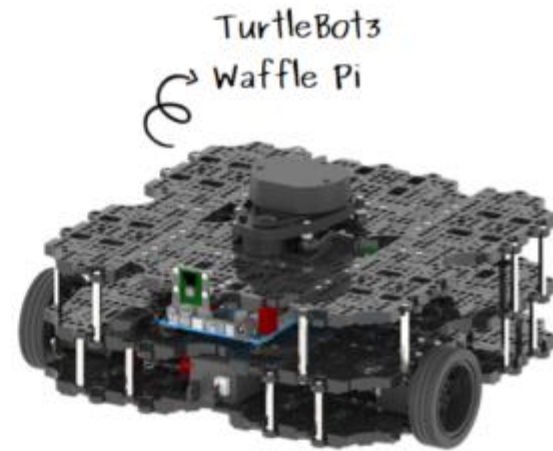
- ▶ 메시지 자료형
- ▶ 로봇의 속도
- ▶ TurtleBot 패키지 설치
- ▶ 로봇의 구동



TurtleBot3



- ▶ TurtleBot
 - ▶ ROS의 표준 플랫폼 로봇
 - ▶ TurtleBot1: 2010년 개발
 - ▶ TurtleBot3: 2017년 개발
 - ▶ Burger, Waffle, Waffle Pi





TurtleBot3 패키지 설치



▶ Install dependent packages

```
user@hostname$ sudo apt-get install ros-noetic-joy ros-noetic-teleop-twist-joy \
ros-noetic-teleop-twist-keyboard ros-noetic-laser-proc \
ros-noetic-rgbd-launch ros-noetic-rosserial-arduino \
ros-noetic-rosserial-python ros-noetic-rosserial-client \
ros-noetic-rosserial-msgs ros-noetic-amcl ros-noetic-map-server \
ros-noetic-move-base ros-noetic-urdf ros-noetic-xacro \
ros-noetic-compressed-image-transport ros-noetic-rqt* ros-noetic-rviz \
ros-noetic-gmapping ros-noetic-navigation ros-noetic-interactive-markers
```

▶ Install TurtleBot3 packages

```
user@hostname$ sudo apt install ros-noetic-dynamixel-sdk
user@hostname$ sudo apt install ros-noetic-turtlebot3-msgs
user@hostname$ sudo apt install ros-noetic-turtlebot3
```



설정 및 실행



▶ TurtleBot3 모델 이름 설정

```
user@hostname$ echo "export TURTLEBOT3_MODEL=waffle_pi" >> ~/.bashrc
```

▶ TurtleBot3 시뮬레이션 설치

```
user@hostname$ cd ~/my_ws/src/  
user@hostname$ git clone -b noetic-devel https://github.com/ROBOTIS-  
GIT/turtlebot3_simulations.git  
user@hostname$ cd ../  
user@hostname$ catkin_make
```

▶ TurtleBot3 월드 실행

```
user@hostname$ source devel/setup.bash  
user@hostname$ roslaunch turtlebot3_gazebo turtlebot3_empty_world.launch
```



Outline



- ▶ 메시지 자료형
- ▶ 로봇의 속도
- ▶ TurtleBot 패키지 설치
- ▶ 로봇의 구동



터틀봇 구동 예제 1



▶ 터틀봇을 전진시키자.

```
#!/usr/bin/env python3
import rospy
from geometry_msgs.msg import Twist

pub = rospy.Publisher('cmd_vel', Twist, queue_size = 1)
rospy.init_node('go_rot')

rate = rospy.Rate(10)
cmd = Twist()

cmd.linear.x = 0.2

while not rospy.is_shutdown():
    pub.publish(cmd)
    rate.sleep()
```

토픽 이름 자료형은 Twist

-----> Twist 자료형의 변수를 만든 뒤

-----> linear.x를 0.5로
→ x축 방향 선형 속도를 0.5m/s로 지정

-----> 속도를 토픽으로 발행

-----> 1초에 10번 반복



노드 만들기



▶ 패키지 만들기

```
user@hostname$ cd ~/my_ws/src
```

```
user@hostname$ catkin_create_pkg wanderbot rospy geometry_msgs sensor_msgs
```

패키지 이름



▶ 노드 소스 파일 작성

```
user@hostname$ cd wanderbot
```

```
user@hostname$ mkdir scripts
```

```
user@hostname$ cd scripts
```

```
user@hostname$ gedit go_rot.py
```

▶ 실행 파일 지정

```
user@hostname$ chmod +x go_rot.py
```



노드 실행



▶ 다른 터미널에서 TurtleBot3 월드 실행

```
user@hostname$ cd ~/my_ws  
user@hostname$ source devel/setup.bash  
user@hostname$ roslaunch turtlebot3_gazebo turtlebot3_empty_world.launch
```

▶ 실행

```
user@hostname$ cd ~/my_ws  
user@hostname$ source devel/setup.bash  
user@hostname$ rosrun wanderbot go_rot.py
```

패키지 이름

노드 실행 파일 이름



터틀봇 구동 예제 2



▶ 터틀봇을 회전시키자.

```
#!/usr/bin/env python3
import rospy
from geometry_msgs.msg import Twist
```

```
pub = rospy.Publisher('cmd_vel', Twist, queue_size = 1)
rospy.init_node('go_rot')
```

```
rate = rospy.Rate(10)
cmd = Twist()
```

```
cmd.linear.z = 0.2
```

-----> angular.z를 0.2로
→ yaw 방향 각속도를 0.2rad/s로 지정

```
while not rospy.is_shutdown():
    pub.publish(cmd)
    rate.sleep()
```

-----> 속도를 토픽으로 발행



▶ 소스 파일 수정 및 실행

```
user@hostname$ cd wanderbot/scripts  
user@hostname$ gedit go_rot.py  
user@hostname$ cd ~/my_ws  
user@hostname$ rosrun wanderbot go_rot.py
```