

## turtlesim 패키지 설치 여부 확인

```
$ ros2 pkg list
```

## turtlesim

### turtlesim 패키지 설치

```
$ sudo apt update
$ sudo apt install ros-foxy-turtlesim
$ cd /opt/ros/foxy/include | ls
```

### turtlesim 실행 파일 목록

```
$ ros2 pkg executables turtlesim
```

### turtlesim 패키지 노드 실행

```
$ ros2 run <package_name> <executable_name>
$ ros2 run turtlesim turtlesim_node
$ ros2 run turtlesim turtle_teleop_key
터미널 창 선택 후, 키보드 화살표 키로 거북이 움직임
중지하고 싶은 창에서 [Ctrl + C] 클릭시 멈춤
```

## rqt 실행

```
$ rqt
$ rqt_graph
```

## rqt

## 노드 목록 확인

```
$ ros2 node list
```

## node

### 노드 정보 확인

```
$ ros2 node info <node_name>
$ ros2 node info /turtlesim
```

### 노드 리매핑

```
$ ros2 run turtlesim turtlesim_node --ros-args --
remap __node:=sohi_turtle
```

## 토픽 목록 확인

```
$ ros2 topic list
$ ros2 topic list -t
```

## topic

### 토픽 정보 확인

```
$ ros2 topic info <topic_name>
$ ros2 topic info /turtle1/cmd_vel
```

### 토픽 타입 확인

```
$ ros2 topic type <topic_name>
$ ros2 topic type /turtle1/cmd_vel
```

### 토픽 내용 확인

```
$ ros2 topic echo <topic_name>
$ ros2 topic echo /turtle1/cmd_vel
```

### 토픽 대역폭 확인 (송수신 메시지 크기)

```
$ ros2 topic bw <topic_name>
$ ros2 topic bw /turtle1/cmd_vel
```

### 토픽 전송 주기 확인 (topic을 publish하는 주기)

```
$ ros2 topic hz <topic_name>
$ ros2 topic hz /turtle1/cmd_vel
```

### 토픽 지연 시간 확인 (header stamp 메시지 사용시 가능)

```
$ ros2 topic delay <topic_name>
$ ros2 topic delay /turtle1/cmd_vel
```

## 토픽 퍼블리시(publish)

## publish

```
$ ros2 topic pub <topic_name> <msg_type> '<args>'
--once : 하나의 메시지를 게시한 다음 종료
--rate 1 : 전송 주기 1hz마다 publish
$ ros2 topic pub --once /turtle1/cmd_vel
geometry_msgs/msg/Twist '{linear: {x: 2.0, y: 0.0, z:
0.0}, angular: {x: 0.0, y: 0.0, z: 1.8}}'
$ ros2 topic pub --rate 1 /turtle1/cmd_vel
geometry_msgs/msg/Twist '{linear: {x: 2.0, y: 0.0, z:
0.0}, angular: {x: 0.0, y: 0.0, z: 1.8}}'
x-z값 입력시 콜론(:) 뒤에 한 칸 공백 꼭 있어야 함.
```

## 서비스 목록 확인

## service

```
$ ros2 service list
$ ros2 service list -t
```

### 서비스 타입 확인

```
$ ros2 service type <service_name>
$ ros2 service type /clear
```

### 서비스 타입에 해당하는 서비스 찾기

```
$ ros2 service find <type_name>
$ ros2 service find std_srvs/srv/Empty
```

### 서비스 요청

```
$ ros2 service call <service_name> <service_type>
<arguments>
$ ros2 service call /clear std_srvs/srv/Empty
$ ros2 service call /spawn turtlesim/srv/Spawn '{x:
2, y: 2, theta: 0.2, name: 'sohi'}'
$ ros2 service call /kill turtlesim/srv/Kill
'{name: 'sohi'}'
$ ros2 service call /reset std_srvs/srv/Empty
$ ros2 service call /turtle1/set_pen
turtlesim/srv/SetPen '{r: 255, g: 255, b: 255, width:
10}'
```

## 파라미터 목록 확인

## parameter

```
$ ros2 param list
```

### 파라미터 내용 확인

```
$ ros2 param describe /turtlesim background_b
```

### 파라미터 값 얻기

```
$ ros2 param get <node_name> <parameter_name>
$ ros2 param get /turtlesim background_g
```

### 파라미터 값 설정

```
$ ros2 param set <node_name> <parameter_name>
<value>
$ ros2 param set /turtlesim background_r 150
```

### 파라미터 저장

```
$ ros2 param dump <node_name>
$ ros2 param dump /turtlesim
```

### 파라미터 로드

```
$ ros2 param load <node_name> <parameter_file>
$ ros2 param load /turtlesim ./turtlesim.yaml
```

### 파라미터 삭제

```
$ ros2 param delete <node_name> <parameter_name>
$ ros2 param delete /turtlesim background_r
```

**turtlesim 패키지 노드 실행(1개 노드)** **turtlesim**

```
$ ros2 run turtlesim turtlesim_node
$ ros2 run turtlesim turtle_teleop_key
```

터미널 창 선택 후, 키보드 화살표 키로 거북이 움직임  
키보드 G|B|V|C|D|E|R|T 키로 거북이 회전, F 키로 취소

**노드 확인** **node**

```
$ ros2 node list
$ ros2 node info /turtlesim
$ ros2 node info /teleop_turtle
```

**노드 시작시 파라미터 파일 로드**

```
$ ros2 run <package_name> <executable_name> --ros-args --params-file <file_name>
$ ros2 run turtlesim turtlesim_node --ros-args --params-file ./turtlesim.yaml
```

**액션 목록(list) 확인** **action**

```
$ ros2 action list
$ ros2 action list -t
```

**액션 정보(information) 확인**

```
$ ros2 action info <action_name>
$ ros2 action info /turtle1/rotate_absolute
```

**액션 목표(goal) 설정**

```
$ ros2 action send_goal <action_name> <action_type> <values>
$ ros2 action send_goal /turtle1/rotate_absolute turtlesim/action/RotateAbsolute '{theta: 1.57}'
$ ros2 action send_goal /turtle1/rotate_absolute turtlesim/action/RotateAbsolute '{theta: -1.57}' --feedback
```

**인터페이스 목록 확인** **interface**

```
$ ros2 interface list
```

**패키지에서 사용 가능한 인터페이스 확인**

```
$ ros2 interface package <package_name>
$ ros2 interface package std_msgs
```

**인터페이스를 제공하는 패키지 목록 확인**

```
$ ros2 interface packages
$ ros2 interface packages --only-msgs
```

**인터페이스의 기본 타입 표시**

```
$ ros2 interface proto <interface_name>
$ ros2 interface proto geometry_msgs/msgs/Twist
```

**토픽 인터페이스 확인**

```
$ ros2 interface show <msg type>
$ ros2 interface show geometry_msgs/msg/Twist
```

**서비스 인터페이스 확인**

```
$ ros2 interface show <type_name>.srv
$ ros2 interface show std_srvs/srv/Empty.srv
$ ros2 interface show turtlesim/srv/Spawn
```

**액션 인터페이스 확인**

```
$ ros2 interface show <type_name>
$ ros2 interface show turtlesim/action/RotateAbsolute
```

**런치 실행(2개 이상 노드)** **launch**

복수의 노드를 함께 실행할 경우

```
$ ros2 launch <package_name> <launch_file_name>
$ ros2 launch demo_nodes_cpp add_two_ints.launch.py
```

**목록 확인** **list**

```
$ ros2 node list
$ ros2 topic list
$ ros2 service list
$ ros2 action list
$ ros2 param list
```

**폴더 생성** **bag**

```
$ mkdir bag_files
$ cd bag_files
```

**토픽 선택**

```
$ ros2 topic list
$ ros2 topic echo /turtle1/cmd_vel
```

**Bag 데이터 확인**

```
$ ros2 bag info <bag_file_name>
$ ros2 bag info test.bag
```

**Bag 데이터 기록**

```
$ ros2 bag record <topic_name>
$ ros2 bag record /turtle1/cmd_vel
  turtlesim 움직이기, Ctrl+C로 레코딩 멈추기
  -a는 모든 토픽 기록

$ ros2 bag record -o <bag_file_name> <topic_name>
$ ros2 bag record -o test.bag /turtle1/cmd_vel /turtle1/pose
```

**Bag 데이터 재생**

```
$ ros2 bag play <bag_file_name>
$ ros2 bag play test.bag
```

**메시지 인터페이스(msg)** **interface file**

필드 타입과 필드 이름으로 구성

```
fieldtype1 fieldname1
fieldtype2 fieldname2
fieldtype3 fieldname3
```

**서비스 인터페이스(srv)**

요청(request) 메시지와 응답(response) 메시지로 구성  
대시(-)로 구분

```
fieldtype1 fieldname
---
fieldtype1 fieldname
```

**액션 인터페이스(action)**

액션 목표(goal), 액션 결과(result),  
액션 피드백(feedback)으로 구성, 대시(-)로 구분

```
fieldtype1 fieldname
---
fieldtype1 fieldname
---
fieldtype1 fieldname
```