

정답:

1) 2, 앞뒤로 움직이고 회전하기 때문에 2 가지 방향이 자유롭다.

2) 4, ROS2 만 라이프싸이클 관리기능이 있다.

3) 3, /service_name 이 앞에 오고 type 이 뒤에 온다.

4) 4, 패키지의 빌드 과정에서뿐만 아니라 실행 과정에서 필요한 의존성도 포함한다.

5) 3

1: 인터페이스 패키지도 다른 ROS2 패키지와 마찬가지로 **ament** 빌드 시스템을 사용하여 빌드됨.

2: 별도로 빌드하면 다른 패키지들은 인터페이스 패키지에 종속됨.

3: 다른 패키지들은 인터페이스를 기반으로 동작하기 때문에 반드시 먼저 참조되어야 한다.

4: 인터페이스 패키지는 사용자가 직접 수정할 수 있다. 메시지, 서비스, 액션 정의 파일(.msg, .srv, .action 등)은 사용자에게 의해 작성되고 수정됨.

5: 인터페이스 패키지의 빌드 과정에서 메시지, 서비스, 액션 정의 파일들이 실제 코드로 변환되어야 하므로 반드시 빌드 과정이 필요함.

6) 4

rqt: 플러그인 기반의 GUI 도구로 Node, Topic, Service 등을 시각적으로 관리, 보기 2 번이 rqt 에 대한 내용,

gazebo : 2D/3D 시뮬레이션용이며 디버깅, 메시지 주고 받기 기능은 없음, 보기 1 번, 3 번이 gazebo 에 대한 설명임,

rviz: 데이터 시각화 도구이며 시뮬레이션 기능이 없음, 보기 5 번이 rviz 에 대한 내용

7) 1

- **--ros-args**: ROS 2 에서 추가적인 인자를 노드에 전달할 수 있게 해주는 옵션
- **-r**: 리맵(remap) 옵션. 일반적으로 토픽이나 서비스, 액션의 이름을 바꿀 때 사용

2: 없는 명령어

3: 터틀심 노드를 실행하면서 Node 명 바꾸기, 이를 통해 다른 노드 또는 컨트롤러가 turtlesim_node 가 아닌 my_turtle 로 노드를 인식한다.

4: 터틀심 노드를 실행하면서 속도 제어 명령을 기본 토픽 turtle1/cmd_vel 이 아닌 /cmd_vel 로 변경하여 통신할 수 있게 설정. 이를 통해 다른 노드 또는 컨트롤러가 /cmd_vel 토픽을 사용해 터틀봇을 제어할 수 있다.

8) 3

1. BEST_EFFORT: Reliability 옵션 중의 하나, 데이터 송신에 집중. 전송 속도를 중시하며 네트워크에 따라 유실 발생 가능성이 있다. RELIABLE: 데이터 수신에 집중. 신뢰성 중시하며 유실 발생 시 재전송을 통해 수신 보장한다.

2. LIFESPAN_DURATION: Lifespan 의 옵션, 정해진 주기 내 수신되는 데이터에만 유효 판정, 이외 데이터는 삭제

4. VOLATILE: Durability 옵션 중의 하나, 구독시작 이후 시점부터 데이터 관리

5. KEEP_ALL: History 의 옵션 중 하나, 모든 데이터 보관 (최대 사이즈는 DDS 벤더마다 다름).

KEEP_LAST 정해진 메시지 큐 사이즈(depth) 만큼 데이터 보관

9) 1

10) 3 , 노드를 "**Shutting down**" 상태로 전환. 노드를 안전하게 종료하는 과정

1: 노드를 "**Configuring**" 상태로 전환하는 명령어. 이 상태에서는 노드가 초기화되고 필요한 설정을 완료하게 됨.

2: 노드를 "**Active**" 상태로 전환. 즉, 노드가 완전히 활성화되어 실제로 데이터를 처리하거나 메시지를 송수신할 준비가 된 상태.

4: ROS 2 의 표준 명령어가 아님. **Finalized** 상태는 노드가 메모리에서 해제되기 직전 상태
노드가 파괴되기 전 디버깅이나 내부 검사를 진행할 수 있는 상태

5: 노드를 메모리에서 완전히 제거하는 명령어.