

1. 원통형 관절의 자유도는 몇인가?

1. 1
2. 2
3. 3
4. 4
5. 5

2. 다음 중 ROS1 과 ROS2 에 대한 설명 중 틀린 것은 무엇인가?

1. ROS1 은 기본적으로 TCPROS 통신 방식을 사용하고, ROS2 는 UDP 방식인 DDS(Data Distribution Service)를 사용한다.
2. ROS2 는 ROS1 과 달리 멀티스레딩과 실시간 처리를 더 잘 지원하도록 설계되었다.
3. ROS1 은 Windows 운영 체제를 공식적으로 지원하지 않지만, ROS2 는 Windows 에서도 동작한다.
4. ROS1 과 ROS2 모두 노드의 라이프사이클 관리 기능을 기본적으로 지원한다.
5. ROS2 는 ROS1 과 비교하여 보안 기능을 강화한 구조를 채택하였다.

3. 다음 중 올바른 ROS 2 service call 사용법은 무엇인가?

1. `ros2 service call type /service_name "{arg1: value1}"`
2. `ros2 service start /service_name type "{arg1: value1}"`
3. `ros2 service call /service_name type "{arg1: value1}"`
4. `ros2 service request type /service_name "{arg1: value1}"`
5. `ros2 run type /service_name "{arg1: value1}"`

4. 다음 중 ROS 2 package.xml 파일에서 사용되는 태그에 대한 설명으로 틀린 것은?

1. `<buildtool_depend>`: 패키지를 빌드하는 도구에 대한 의존성을 선언할 때 사용된다.
2. `<exec_depend>`: 패키지를 실행할 때 필요한 의존성을 선언할 때 사용된다.
3. `<test_depend>`: 패키지 테스트 시 필요한 의존성을 선언할 때 사용된다.
4. `<depend>`: 패키지의 빌드 과정에서만 필요한 의존성을 선언할 때 사용된다.
5. `<maintainer>`: 패키지 유지 보수를 담당하는 사람을 지정할 때 사용된다.

5. 다음 중 인터페이스 패키지(message, service, action 패키지)를 따로 모아서 빌드하는 이유로 올바른 것은 무엇인가?

1. 인터페이스 패키지는 ROS2 에서 제공하는 별도의 빌드 시스템을 사용하기 때문에.
2. 인터페이스 패키지를 별도로 빌드하면 다른 패키지들이 이 인터페이스에 종속되지 않기 때문에.
3. 인터페이스 패키지는 빌드 과정에서 먼저 생성되어야 다른 패키지들이 이를 참조할 수 있기 때문에.
4. 인터페이스 패키지는 사용자가 직접 수정할 수 없기 때문에 별도로 관리하기 위함이다.
5. 인터페이스 패키지에 포함된 파일들은 빌드가 불필요하므로 빌드를 따로 진행한다.

6. ROS 2 에서 사용되는 rqt, Gazebo, RViz 의 주요 기능에 대한 설명 중 올바른 것을 고르시오.

1. rqt 는 3D 로봇 시뮬레이션 환경을 제공하며, 센서 데이터와 물리적 상호작용을 시뮬레이션할 수 있습니다.

2. Gazebo 는 ROS 2 에서 그래픽 사용자 인터페이스(GUI)로 노드의 시각적 디버깅과 메시지 주고받기를 확인할 수 있도록 도와줍니다.
3. RViz 는 2D/3D 시뮬레이션을 통해 로봇의 움직임과 물리적 상호작용을 테스트하는 데 주로 사용됩니다.
4. rqt 는 플러그인 기반의 GUI 도구로, ROS 2 의 노드, 주제, 서비스 등을 시각적으로 관리 및 디버깅할 수 있습니다.
5. Gazebo 는 ROS 2 에서 데이터 시각화 도구로 주로 사용되며, 로봇 센서 데이터와 주제(topic) 데이터를 시각적으로 표현합니다.

7. 네임스페이스를 변경하려고 한다. 다음중 맞는 명령어는 무엇인가?

1. `ros2 run turtlesim turtlesim_node --ros-args -r __ns:=/tutorial`
2. `ros2 run turtlesim turtlesim_node --ros-args -r __name:=tutorial`
2. `ros2 run turtlesim turtlesim_node --ros-args -r __node:=my_turtle`
3. `ros2 run turtlesim turtlesim_node --ros-args -r turtle1/cmd_vel:=/cmd_vel`
4. `ros2 run turtlesim turtlesim_node --ros-args -p background_b:=0`

8. 다음 중 구독시작 시점 전의 데이터를 사용하도록 설정하는 qos 옵션은 무엇인가?

1. BEST\_EFFORT
2. LIFESPAN\_DURATION
3. TRANSIENT\_LOCAL
4. VOLATILE
5. KEEL\_ALL

9. 다음 중 my\_package 라는 이름의 패키지를 생성하는 명령어는 무엇인가?

1. `ros2 pkg create my_package --build-type ament_cmake --dependencies rclcpp std_msgs`
2. `ros2 pkg build my_package --build-type ament_python`
3. `ros2 pkg make my_package --build-type ament_cmake --dependencies rclcpp std_msgs`
4. `ros2 package create my_package --build-type ament_cmake`
5. `ros2 package create my_package --build-type ament_python`

10. lifecycle\_talker 라는 노드가 있다고 가정했을 때 다음 중 노드의 마지막 상태로 전환하는 명령어는 무엇인가?

1. `ros2 lifecycle set /lifecycle_talker configure`
2. `ros2 lifecycle set /lifecycle_talker activate`
3. `ros2 lifecycle set /lifecycle_talker shutdown`
4. `ros2 lifecycle set /lifecycle_talker finalized`
5. `ros2 lifecycle set /lifecycle_talker destroy`