

정답

1. 4번

Python Style

✓ Python Enhancement Proposals(PEPs)의 PEP 8을 준수

✓ Wiki 참고: <https://wiki.ros.org/PyStyleGuide>

■ 기본 규칙

■ Python 3(Python 3.5 이상)

■ 라인 길이

■ 최대 100 문자

■ 이름 규칙 (Naming)

■ CamelCased, snake_case, ALL_CAPITALS 만 사용

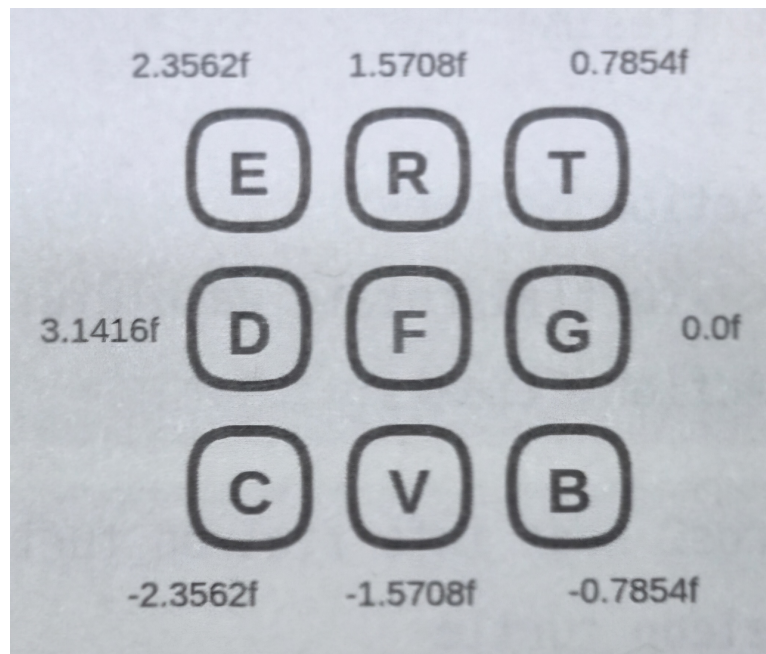
✓ CamelCased: 타입, 클래스

✓ snake_case: 파일, 패키지, 인터페이스, 모듈, 변수, 함수, 메소드

✓ ALL_CAPITALS: 상수

2. ROS_DOMAIN_ID

3. code



```
ros2 service call /spawn turtlesim/srv/Spawn "{x: 2, y: 2,
```

```
ynu@ynu-ROKEY:~$ ros2 service call /spawn turtlesim/srv/Sp
requester: making request: turtlesim.srv.Spawn_Request(x=2
response:
turtlesim.srv.Spawn_Response(name='ynu')
```

4. 3, 5번

3,5 = topic

5. code

```
#ROS2
alias humble='source /opt/ros/humble/setup.bash; echo "ros
```

- 빈칸1) source
- 빈칸2) setup.bash
- 빈칸3) echo
- 빈칸4) ros2 humble start

6. 1, 4, 5번

- **1)** source install/setup.bash는 ros2_ws 디렉토리 안에서 실행해야 함
- **4)** 의존성을 추가할 때는 --dependencies 옵션을 사용해야 하며, --license는 라이선스를 설정하는 옵션이다.
- **5)** mkdir 명령어는 기본적으로 한 번에 여러 폴더를 만들 수 없고, p 옵션을 사용해야 함; 예: mkdir -p ros2_ws/src

7. 2번 (get_logger())

```
def timer_callback(self):
    msg = String() # String 메시지 객체 생성
    msg.data = 'Hello World: %d' % self.i # 메시지 데이터 생성

    self.publisher_.publish(msg) # 발행자 사용하여 메시지를 top
    # 로깅
```

```
self.get_logger().info('Publishing: %s' % msg.data) #  
# 메시지 한번 출력할 때마다 카운트 증가  
self.i += 1
```

8. 1, 3번

9. [10 10 4]

10. code .