

1. **setup.cfg** 와 **setup.py**의 차이점이 표 안의 번호 순서대로 알맞게 연결된 것을 고르시오.

	setup.cfg	setup.py
접근 방식	1	2
유연성	3	4
사용 추세	5	

- 프로그래밍 방식 - 선언적 방식 - 높은 유연성 - 명확한 설정 - 가능한 최소화
 - 프로그래밍 방식 - 선언적 방식 - 명확한 설정 - 높은 유연성 - 현대적 추세
 - 선언적 방식 - 프로그래밍 방식 - 명확한 설정 - 높은 유연성 - 가능한 최소화
 - 선언적 방식 - 프로그래밍 방식 - 명확한 설정 - 동적 계산 - 현대적 추세
 - 선언적 방식 - 프로그래밍 방식 - 높은 유연성 - 명확한 설정 - 가능한 최소화
2. 아래 **setup.py**의 **entry_point** 필드에서 '**talker = py_pubsub.publisher_member_function:main**'에 관한 설명으로 옳지 않은 것을 고르시오.

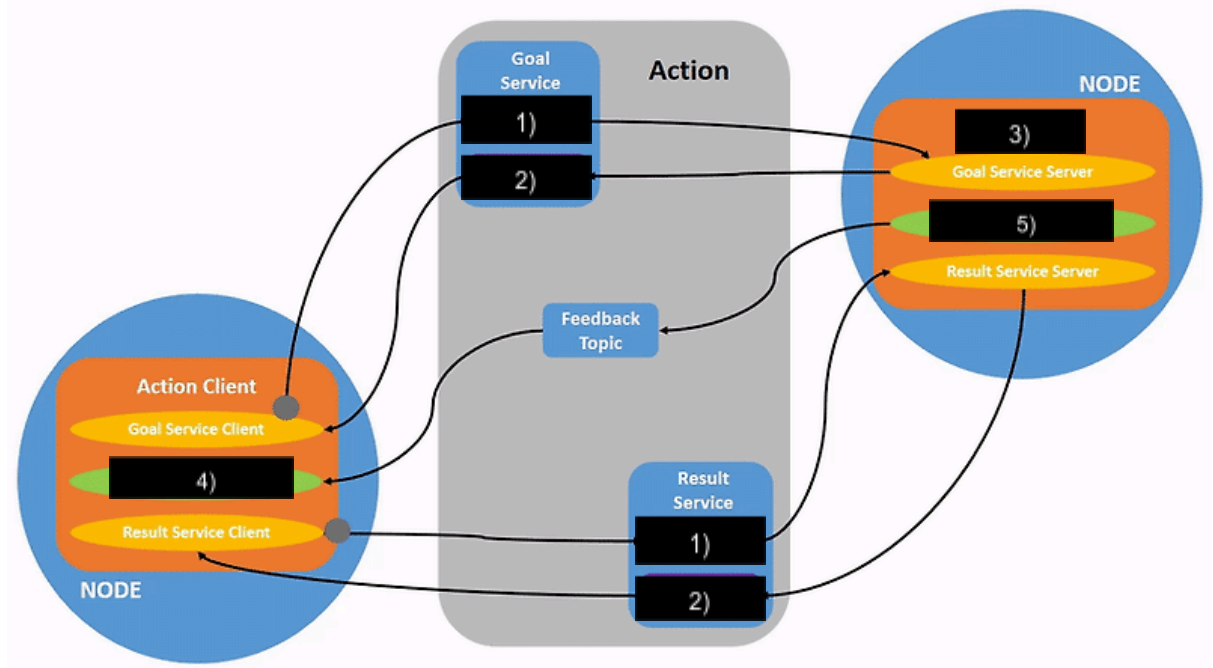
```
entry_points={
    'console_scripts': [
        'talker =
py_pubsub.publisher_member_function:main',
    ],
},
```

- 'talker'라는 커맨드 라인 명령어를 생성하라는 지시이다.
 - 사용자가 커맨드 라인에서 **talker**라고 입력하면,
py_pubsub.publisher_member_fucntion 모듈의 **main** 함수가 실행된다.
 - 이를 **colcon** 빌드 과정을 통해 컴파일한 후에 사용할 수 있다.
 - 이 설정을 사용하여 파이썬 패키지를 설치하면, 사용자는 커맨드 라인에서 바로 **talker** 명령어를 사용하여 해당 기능을 실행할 수 있다.
 - talker** 명령어는 **py_pubsub** 패키지가 사용자 정의 워크스페이스 설치되어 있어도 사용할 수 있다.
3. 나머지 모든 빈칸을 채우시오.

	토픽	서비스	액션
연속성			복합(토픽+서비스)
방향성			
동기성			
다자간 연결		1:1	
노드 역할		Server, Client	

동작 트리거			action 인터페이스
인터페이스			action
CLI명령어			

4. 빈칸 1)~5)을 채우시오.



- 1)
- 2)
- 3)
- 4)
- 5)

5. 다음은 `ex_calculator`의 액션 서버와 관련한 코드이다. 아래 목록에서 빈 칸에 알맞은 단어를 채우시오.

```
self.arithmetic_action_server = ActionServer(
    self,
    ArithmeticChecker,
    'arithmetic_checker',
    self.execute_checker,
    callback_group=self.callback_group)
```

액션 서버로 선언하는 부분과 ()를 지정하는 부분으로 나뉘어져 있다. `rcipy.action` 모듈의 () 클래스를 이용하여 액션 서버로 선언되었으며, ()는 `execute_checker`로 지정되었다. 액션의 타입은 ()으로 선언하였고, 액션 이름은 ()이다. `callback_group`은 () 스레드 () 실행을 위해 설정되었고, 실제 액션 목표를 받은 후에 실행되는 콜백함수는 ()이다.

목록

arithmetic_action_server, rclpy.action, operator, ActionServer, ArithmeticChecker, 'arithmetic_checker', execute_checker, self.callback_group, 콜백함수, 복합, 멀티, 병렬 콜백함수, 다중 콜백함수, 서브스크라이버, 서버, 클라이언트, 액션 목표

6. 아래 설명에 부합하며, 각각의 조건에 해당하는 명령어를 작성하시오.

파라미터의 정보를 확인하고 파라미터를 설정하거나 읽어오는 등의 일을 수행할 수 있는 명령어

- a. 사용 가능한 모든 파라미터 목록을 확인
- b. /turtlesim 노드의 background_r 파라미터를 250이라는 값으로 설정
- c. 파라미터가 어떤 형태, 목적, 인터페이스 형태, 최소/최댓값을 갖는지 확인
- d. /turtlesim 노드의 PARAMETER 1 이라는 이름을 갖는 파라미터를 삭제

7. 아래 설명에 부합하며, 각각의 조건에 해당하는 명령어를 작성하시오.

토픽을 저장하거나 재생할 때 사용하는 명령어

- a. /turtle1/cmd_vel 토픽을 'my_turtle'이라는 이름으로 저장
- b. 'my_topic'이라는 rosbag 파일의 정보를 확인
- c. 지정한 rosbag 파일을 재생

8. 아래 설명에 부합하며, 각각의 조건에 해당하는 명령어를 작성하시오.

ROS2 설정 및 네트워크 패키지 버전, RMW 등과 같은 ROS2 개발환경의 잠재적 문제를 확인하는 명령어

- a. 네트워크 연결 확인
- b. 체크한 모든 아이템을 확인
- c. 체크할 때 실패한 아이템을 확인
- d. 경고성 아이템을 확인

9. ros2env 라는 이름을 갖는 python 패키지를 생성하기 위해 아래 과정에 따라 ros2 폴더의 src 폴더로 이동한 후 빈 칸에 들어갈 명령어를 작성하시오

\$ cd ~/ros2_ws/src

\$ ()

10. `env` 패키지를 빌드한 후, 주석에 따라 제대로 빌드가 되었음을 확인하기 위해 사용하는 명령어들을 작성하시오.

```
$          # Show this help message and exit
$          # Display all environment variables.
$          # Display the ROS environment variables.
$          # Display the DDS environment variables.
```