

## 1. 선언적 방식 - 프로그래밍 방식 - 명확한 설정 - 동적 계산 - 현대적 추세

### 패키지(Package) 설정

#### ▪ setup.py & setup.cfg 차이점

	setup.py	setup.cfg
접근 방식	프로그래밍 방식으로 패키지 설정을 제공	선언적 방식으로 설정을 제공
유연성	동적 계산과 사용자 정의 명령어를 지원하는 높은 유연성을 제공	보다 간단하고 명확한 패키지 설정을 지향
사용 추세	가능한 경우 setup.py 파일은 최소화하거나 제거	현대의 Python 패키징은 setup.cfg를 통한 선언적 패키지 설정을 선호

## 2. c. 파이썬으로 진행할 경우 colcon 빌드 과정이 필요하지 않다.

### ✓ setup.py 설정

#### ▪ entry\_points

- entry\_points는 Python의 setuptools에서 사용되는 설정의 일부
- 특히 파이썬 패키지를 설치할 때 커맨드 라인 스크립트를 자동으로 생성하도록 지시하는 데 사용

#### ▪ console\_scripts

- console\_scripts는 entry\_points의 하위 항목으로, 커맨드 라인에서 실행할 수 있는 스크립트를 지정

#### ▪ 'talker = py\_pubsub.publisher\_member\_function:main'

- 이 항목은 'talker'라는 커맨드 라인 명령어를 생성하라는 지시
- 사용자가 커맨드 라인에서 talker라고 입력하면, py\_pubsub.publisher\_member\_function 모듈의 main 함수가 실행
- 결과적으로, 이 설정을 사용하여 파이썬 패키지를 설치하면, 사용자는 커맨드 라인에서 바로 **talker** 명령어를 사용하여 해당 기능을 실행할 수 있게 됨
- ROS2에서 파이썬 노드를 쉽게 실행할 수 있도록 하는데 특히 유용함
- 이러한 방식을 통해 ROS2는 Python 스크립트를 바로 실행할 수 있는 실행 가능한 커맨드를 제공

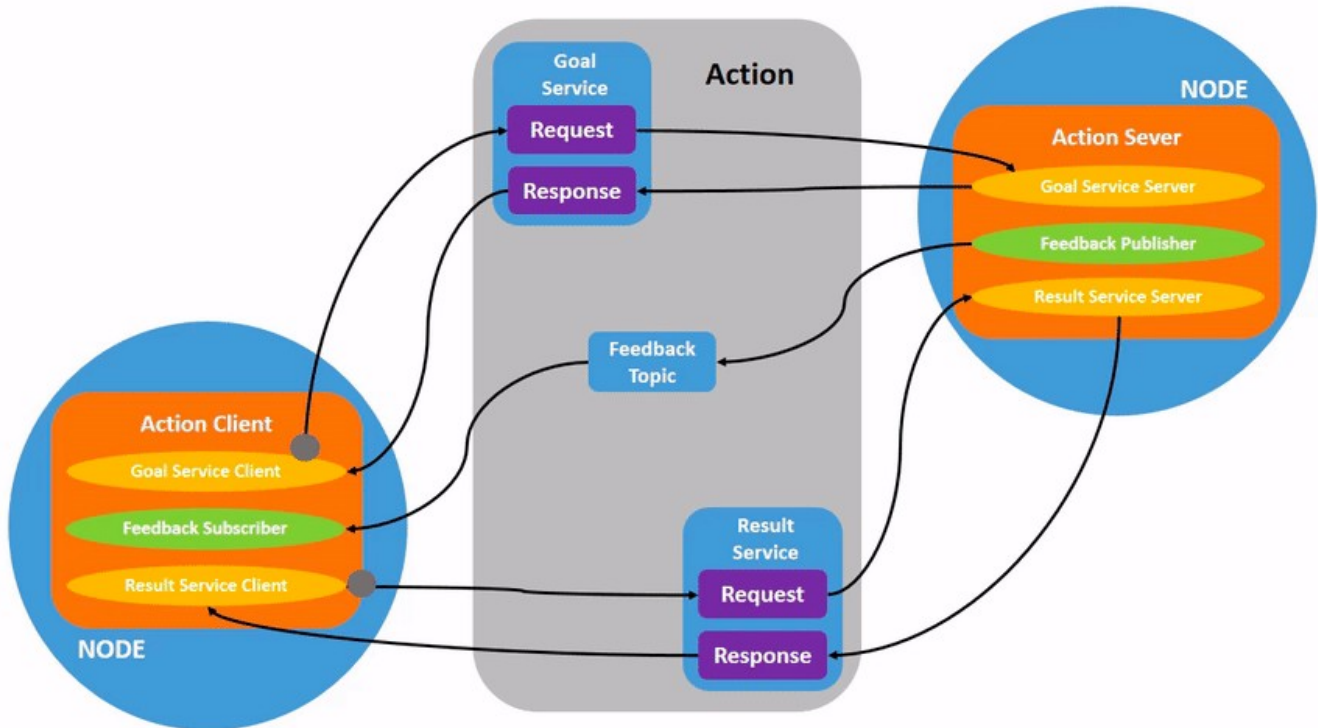
3.

### 토픽, 서비스, 액션 인터페이스

토픽, 서비스, 액션 복습

	토픽	서비스	액션
연속성	연속성	일회성	복합(토픽 + 서비스)
방향성	단방향	양방향	양방향
동기성	비동기	동기	동기 + 비동기
다자간연결	1:1, 1:N, N:1, N:N	1:1	1:1
노드역할	Publisher, Subscriber	Server, Client	Server, Client
동작트리거	퍼블리셔	클라이언트	action 인터페이스
인터페이스	msg	srv	action
CLI명령어	ros2 topic	ros2 service	ros2 action
사용 예	센서 데이터, 로봇 상태, 로봇 좌표, 로봇 속도 명령 등	LED 제어, 모터 토크 On/Off IK/FK 계산, 이동 경로 계산 등	목적지로 이동, 물건 파지, 복합 테스트

4. request, response, action server, feedback subscriber, feedback publisher



5. 콜백함수, ActionServer, 콜백함수, ArithmeticChecker, 'arithmetic\_checker', 멀티, 병렬 콜백함수, execute\_checker

액션 프로그래밍

## 액션 서버 코드

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

- calculator 노드는 서브스크라이브하여 저장하고 있는 변수 a와 b와 operator 노드로부터 요청 값으로 받은 연산자를 이용하여 계산(a 연산자 b)하고 operator 노드에게 연산의 결과값을 서비스 응답값으로 보낸다는 것은 이전 강좌에서 설명하였다. 여기에 각 연산을 기록하고 액션 서버로 동작하는 부분에 대해 추가 설명을 하도록 하겠다
- 액션 서버와 관련한 코드는 다음과 같다. 서버 관련 코드는 액션 서버로 선언하는 부분과 콜백함수를 지정하는 것이다. arithmetic\_action\_server이 액션 서버로 이는 rclpy.action 모듈의 ActionServer 클래스를 이용하여 액션 서버로 선언되었으며 액션의 타입으로 ArithmeticChecker로 선언하였고, 액션 이름으로는 'arithmetic\_checker', 액션 클라이언트로부터 액션 목표를 받으면 실행되는 콜백함수는 execute\_checker으로 지정했으며 멀티 스레드 병렬 콜백함수 실행을 위해 지난 번 강좌에서 설명한 callback\_group 설정을 하였다
- 이러한 설정들은 액션 서버를 위한 기본 설정이고 실제 액션 목표를 받은 후에 실행되는 콜백함수는 execute\_checker 함수임을 알아두자

6. ros2 param

ros2 param list

ros2 param set /turtlesim background\_r 250

ros2 param describe /turtlesim background\_r

ros2 param delete /turtlesim {PARAMETER 1}

7. ros2 bag

ros2 bag record -o my\_turtle /turtle1/cmd\_vel

ros2 bag info my\_topic

ros2 bag play my\_topic

8. ros2 doctor

ros2 doctor hello

ros2 doctor -r

ros2 doctor -rf

ros2 doctor -iw

9. ros2 pkg create --build-type ament\_python ros2env

10.

`ros2 -h`

`ros2 env list`

`ros2 env list -r`

`ros2 env list -d`