

워크 스페이스(Workspaces)

Workspaces

원하는 작업 수행을 위한 code를 작성하는 공간
워크 스페이스 위치 : ~/Workspaces/<ws_name>

디렉토리 생성

```
$ mkdir <dir_name>
$ mkdir <dir_name1> <dir_name2> <dir_name3>
dir_name을 여러개 쓰면, 여러 폴더를 동시에 생성함
-p : 계층형 디렉토리를 생성함
```

워크 스페이스(ros2_ws)와 소스폴더(src) 생성

```
$ mkdir -p ~/Workspaces/ros2_ws/src
```

패키지(Package)

Package

파이썬 모듈을 계층적(디렉터리 구조)으로 관리함.

패키지 생성

```
$ ros2 pkg create <package_name> --build-type
<build_type> --dependencies <package1> <package2>
--build-type : ament_python 파이썬으로 코딩한 파일 빌드
--dependencies : 의존하는 패키지들(import한 패키지 작성)
https://docs.ros2.org/foxy/api/rclpy/
```

디렉토리 이동

```
$ cd <dir_name>
```

```
$ cd ~/Workspaces/ros2_ws/src
$ ros2 pkg create hello_pkg --build-type
ament_python --dependencies rclpy
```

노드 프로그램 파일

hello_ros2.py

```
$ cd ~/Workspaces/ros2_ws/src/hello_pkg/hello_pkg
$ code ./hello_ros2.py
```

```
import rclpy
from rclpy.node import Node
```

```
def main(args=None):
    rclpy.init(args=args)
    node = Node('hello_node')
    node.get_logger().info('Hello ROS2')
    rclpy.spin(node)
    rclpy.shutdown()
```

```
if __name__ == '__main__':
    main()
```

파이썬 패키지 설정 파일

setup.py

```
ROS2 파이썬 패키지에서 배포를 위한 파일
$ cd ~/Workspaces/ros2_ws/src/hello_pkg
$ code setup.py
```

entry_points 옵션의 console_scripts 키를 사용한 실행 파일의 설정을 함. 예) hello_script 콘솔 스크립트는 hello_pkg.hello_ros2 모듈의 main 함수를 호출함

```
entry_points={
    'console_scripts': [
        'hello_script = hello_pkg.hello_ros2:main'
    ],
},
```

빌드(Build)

colcon build

소스 코드 파일을 컴퓨터에서 실행할 수 있는 독립적인 형태로 변환함

```
$ cd ~/Workspaces/<ws_name>
```

Home폴더에 위 경로로 폴더를 생성한 후 이동함.

```
$ colcon build
```

각 패키지에 기술되어 있는 종속성 그래프를 해석하고 토폴로지 순서로 각 패키지에 대한 특정 빌드 시스템을 호출함.

```
$ source install/setup.bash
```

설정 스크립트를 적용함. 패키지 빌드 후 setup.bash를 실행해야 ROS 2 패키지를 찾거나 노드를 실행하는 것이 가능함.

빌드 옵션

```
$ colcon build --symlink-install
--packages-select <package_name> <package_name>
--packages-up-to <package_name>
```

--symlink-install : 파일 복사 대신 심볼릭 링크 형태로 저장
심볼릭 링크(symbolic link) : 링크를 연결하여 원본 파일을 직접 사용하는 것과 같은 효과를 내는 링크

--packages-select 옵션 : 특정 패키지만 선택해서 빌드
특정 패키지 및 의존성 패키지를 함께 빌드

--packages-up-to 옵션 : 특정 패키지의 첫 빌드 후에는 환경설정 파일을 불러와서 실행 가능한 패키지의 노드 설정을 해줘야 빌드 된 노드 실행 가능함

파이썬 파일 실행

```
$ cd ~/Workspaces/ros2_ws/src/hello_pkg/hello_pkg
$ python3 hello_ros2.py
```

빌드 후 노드 실행(ros2 run)

```
$ cd ~/Workspaces/ros2_ws
$ colcon build
$ source install/setup.bash
$ ros2 pkg executables hello_pkg
$ ros2 run hello_pkg hello_script
```

설정 스크립트

setup script

새로운 패키지를 빌드 시 설정 스크립트를 터미널에서 실행해야 함.

local_setup.bash : 워크스페이스의 모든 패키지에 대한 환경을 설정함

setup.bash : 워크스페이스와 ROS가 설치된 개발환경에 대한 환경을 설정함

bash : interactive shell, user input에 의해 script가 실행되는 shell mode

~/.bashrc : bash로 실행될 때 먼저 자동으로 실행되는 파일.
새로운 터미널을 열 때마다 파일에 적힌 명령어가 수행됨

~/.bashrc 파일에 다음 내용 추가하면 터미널 열 때 명령어 수행이 되어, 빌드 후 source install/setup.bash를 하지 않아도 됨. ~/.bashrc에 다음 내용을 저장한 후 사용하는 것을 추천함.

```
export my_ws=~/Workspaces/ros2_ws
source /opt/ros/foxy/setup.bash
source $my_ws/install/setup.bash
```

```
alias eb='code ~/.'
alias nb='nano ~/.'
alias sb='source ~/.'
alias cw='cd $my_ws'
alias cs='cd $my_ws/src'
alias cb='cd $my_ws && colcon build --symlink-install && source ~/.bashrc'
```