

```
1 cmake_minimum_required(VERSION 3.8)
2 project(tutorial_interfaces)
3
4 if(CMAKE_COMPILER_IS_GNUCXX OR CMAKE_CXX_COMPILER_ID MATCHES "Clang")
5     add_compile_options(-Wall -Wextra -Wpedantic)
6 endif()
7
8 # find dependencies
9 find_package(ament_cmake REQUIRED)
10 # uncomment the following section in order to fill in
11 # further dependencies manually.
12 # find_package(<dependency> REQUIRED)
13 find_package(rclcpp REQUIRED)
14 find_package(rosidl_default_generators REQUIRED)
15
16
17 rosidl_generate_interfaces(${PROJECT_NAME}
18     "msg/Num.msg"
19     "srv/AddThreeInts.srv"
20 )
21 if(BUILD_TESTING)
22     find_package(ament_lint_auto REQUIRED)
23     # the following line skips the linter which checks for copyrights
24     # comment the line when a copyright and license is added to all source files
25     set(ament_cmake_copyright_FOUND TRUE)
26     # the following line skips cpplint (only works in a git repo)
27     # comment the line when this package is in a git repo and when
28     # a copyright and license is added to all source files
29     set(ament_cmake_cpplint_FOUND TRUE)
30     ament_lint_auto_find_test_dependencies()
31 endif()
32
33 ament_package()
```

<https://docs.ros.org/en/crystal/Tutorials/Custom-ROS2-Interfaces.html>

해당 링크를 참고하여 제작.

Msg에 int vector 설정,
해당 파일 불러오기
Srv는 서비스 프로토콜이라 현재로서는 필요없음

```
1  <?xml version="1.0"?>
2  <?xml-model href="http://download.ros.org/schema/package_format3.xsd" schematypens="http://www.w3.org/2001/XMLSchema"?>
3  <package format="3">
4    <name>tutorial_interfaces</name>
5    <version>0.0.0</version>
6    <description>TODO: Package description</description>
7    <maintainer email="david@todo.todo">david</maintainer>
8    <license>TODO: License declaration</license>
9
10   <buildtool_depend>ament_cmake</buildtool_depend>
11
12   <build_depend>roscpp</build_depend>
13   <exec_depend>roscpp</exec_depend>
14   <member_of_group>roscpp_interface_packages</member_of_group>
15
16   <test_depend>ament_lint_auto</test_depend>
17   <test_depend>ament_lint_common</test_depend>
18
19   <export>
20     <build_type>ament_cmake</build_type>
21   </export>
22 </package>
```

<https://docs.ros.org/en/crystal/Tutorials/Custom-ROS2-Interfaces.html>

해당 링크를 참고하여 제작.

```
#include <chrono>
#include <memory>
#include <iostream>

#include "rclcpp/rclcpp.hpp"
#include "tutorial_interfaces/msg/num.hpp"    // CHANGE

using namespace std::chrono_literals;

tutorial_interfaces::msg::Num data;

class MinimalPublisher : public rclcpp::Node
{
public:
    MinimalPublisher()
    : Node("minimal_publisher")
    {
        p = this->create_publisher<tutorial_interfaces::msg::Num>("topic", 10);    // CHANGE

        int i=0;

        std::cout<<"숫자 입력 ";
        std::cin>>i;
        if(!std::cin){
            std::cout<<" 입력 실패"<<std::endl;
            std::cin.clear();
            std::cin.ignore(1000, '\n');
        }
        else{
            data.nv.push_back(i);
        }
        p->publish(data);
    }
};
```

<https://docs.ros.org/en/crystal/Tutorials/Custom-ROS2-Interfaces.html>

해당 링크를 참고하여 제작
-전역으로 msg를 data로 설정

-숫자 입력 받기

-예외 처리
(숫자 아니면 버퍼 지움)

-data가 벡터이기에 해당 벡터에 1 push_back 하여 추가

Data 송신

<https://docs.ros.org/en/crystal/Tutorials/Custom-ROS2-Interfaces.html>

해당 링크를 참고하여 제작

송신 받은 백터를 msg로 저장

입력 받은 백터를 출력(1일차 송신부와 동일)

```
#include <memory>
#include <iostream>
#include "rclcpp/rclcpp.hpp"
#include "tutorial_interfaces/msg/num.hpp"    // CHANGE
using std::placeholders::_1;

class MinimalSubscriber : public rclcpp::Node
{
public:
    MinimalSubscriber()
    : Node("minimal_subscriber")
    {
        subscription_ = this->create_subscription<tutorial_interfaces::msg::Num>(    // CHANGE
            "topic", 10, std::bind(&MinimalSubscriber::topic_callback, this, _1));
    }

private:
    void topic_callback(const tutorial_interfaces::msg::Num::SharedPtr msg) const    // CHANGE
    {
        for(auto i : msg->nv){
            std::cout<< i<<std::endl;
        }
    }

    rclcpp::Subscription<tutorial_interfaces::msg::Num>::SharedPtr subscription_;    // CHANGE
};

int main(int argc, char * argv[])
{
    rclcpp::init(argc, argv);
    rclcpp::spin(std::make_shared<MinimalSubscriber>());
    rclcpp::shutdown();
    return 0;
}
```