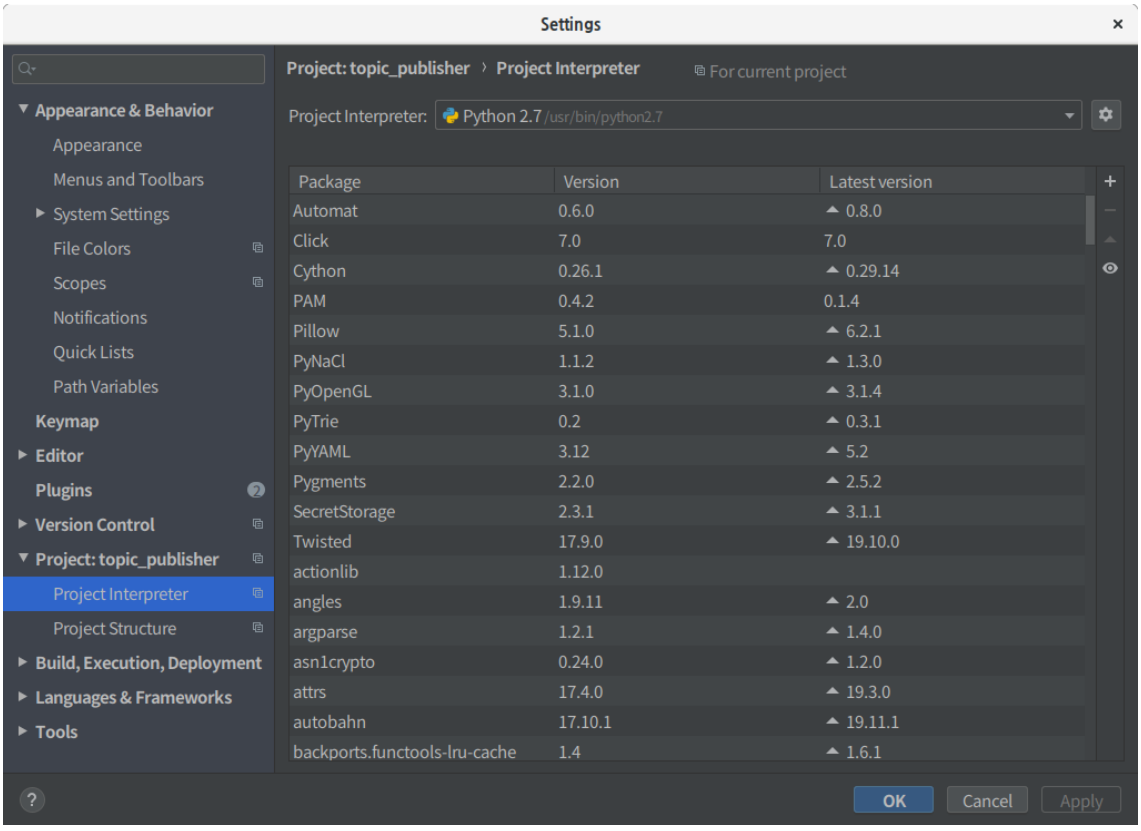
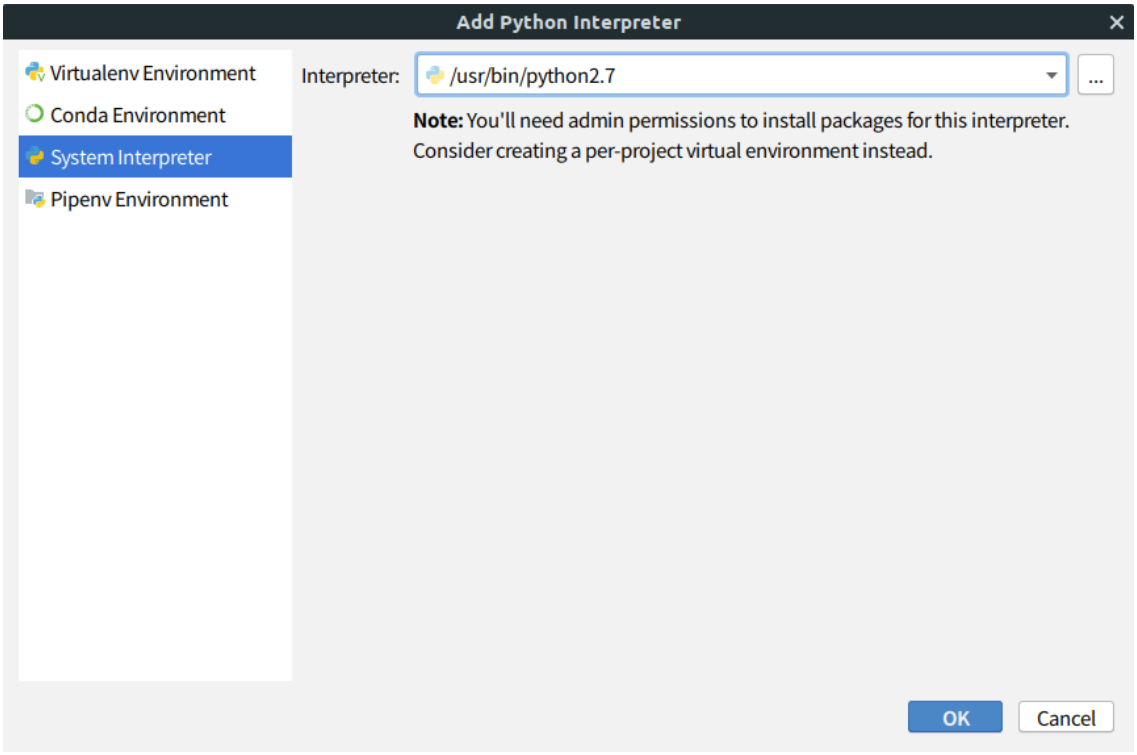
1. python 2.7 설정

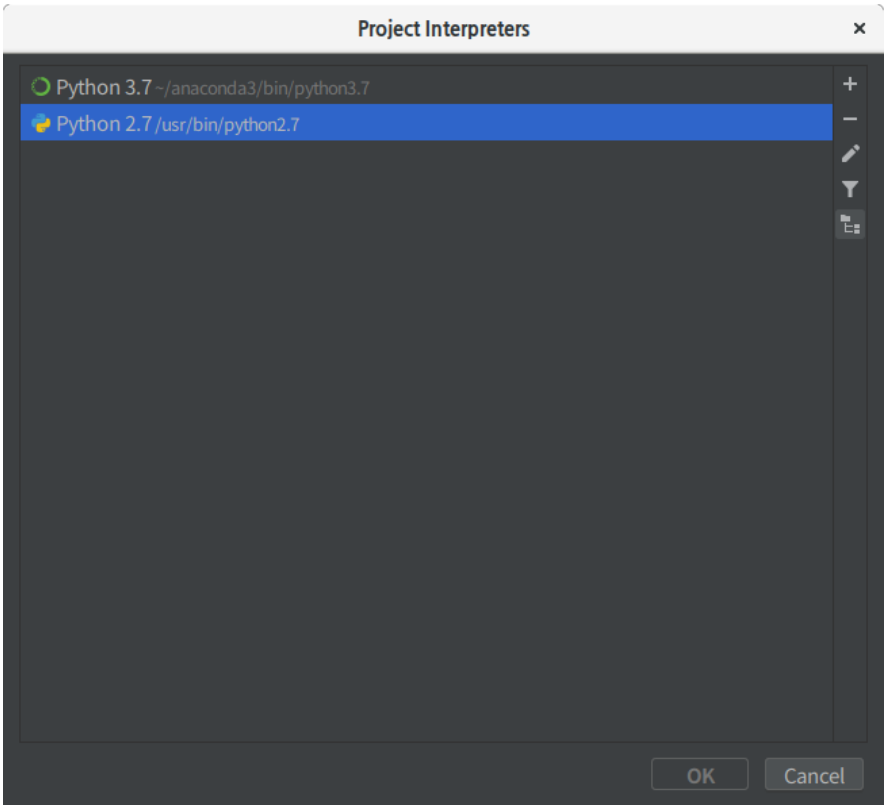


a. 우측의 톱니바퀴 > add

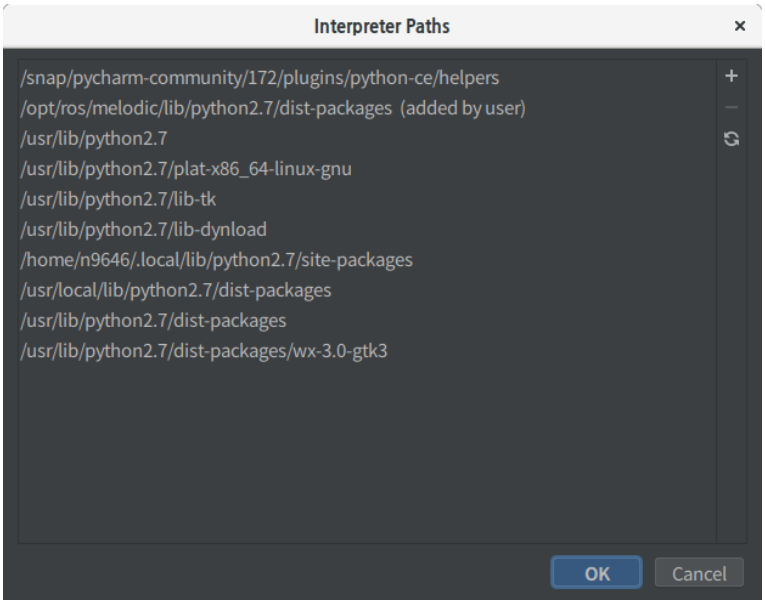


b. System Interpreter > Interpreter: /usr/bin/python2.7 > OK

c. 우측의 톱니바퀴 > Show All



d. show paths for the selected interpreter 선택(우측 폴더트리 버튼)



e. 우측의 +를 선택 후, /opt/ros/melodic/lib/python2.7/dist-packages 추가

2. pycharm 실행방법

bash -i -c “/snap/pycharm-community/current/bin/pycharm.sh” %f

3. 새 프로젝트 생성, python file 생성후 아래 예제 작성

turtle를 값을 받아 움직이는 예제 (속도, 거리, 방향(전/후))

```
#!/usr/bin/env python
import rospy
from geometry_msgs.msg import Twist

def move():
    # Starts a new node
    rospy.init_node('robot_cleaner', anonymous=True)
    velocity_publisher = rospy.Publisher('/turtle1/cmd_vel', Twist, queue_size=10)
    vel_msg = Twist()

    #Receiveing the user's input
    print("Let's move your robot")
    speed = input("Input your speed:")
    distance = input("Type your distance:")
    isForward = input("Foward?: ") #True or False

    #Checking if the movement is forward or backwards
    if(isForward):
        vel_msg.linear.x = abs(speed)
    else:
        vel_msg.linear.x = -abs(speed)
    #Since we are moving just in x-axis
    vel_msg.linear.y = 0
    vel_msg.linear.z = 0
    vel_msg.angular.x = 0
    vel_msg.angular.y = 0
    vel_msg.angular.z = 0

    while not rospy.is_shutdown():

        #Setting the current time for distance calculus
        t0 = rospy.Time.now().to_sec()
        current_distance = 0

        #Loop to move the turtle in an specified distance
        while(current_distance < distance):
            #Publish the velocity
            velocity_publisher.publish(vel_msg)
            #Takes actual time to velocity calculus
            t1=rospy.Time.now().to_sec()
            #Calculates distancePoseStamped
            current_distance= speed*(t1-t0)
        #After the loop, stops the robot
        vel_msg.linear.x = 0
        #Force the robot to stop
        velocity_publisher.publish(vel_msg)

if __name__ == '__main__':
    try:
        #Testing our function
        move()
    except rospy.ROSInterruptException: pass
```

4. 실행 방법 및 결과

a. 터미널 실행 후,
   $ roscore

b. 새 터미널 실행 후
   $ rosrun turtlesim turtlesim_node
c. pycharm에서 run(Ctrl + Shift + F10)