1. Body control

1.1 Experimental purpose

In this course, we will learn how to control the body posture of dogzilla, and control the body posture change of robot dog by using the flexible function of twelve joints under the condition that the foot end is not moved.

1.2 Experimental preparation

The functions of dogzilla Python library involved in this course.

translation(direction, data)：Body position translation.

direction：Single character or character list, 'x' represents back to back translation, 'y' represents back to left translation, 'z' represents height.

data：which means the translation distance of the fuselage position(mm). The value range is X: [-35,35], Y: [- 18,18], Z: [75115]

attitude(direction, data)：Fuselage attitude adjustment.

direction：Single character or character list, 'R' represents roll angle, 'p' represents pitch angle, 'y' represents yaw angle

data：Represents the adjustment range of fuselage attitude, in °.r:[-20,20], p:[-15,15], y:[11,11]

1.3 Experimental process

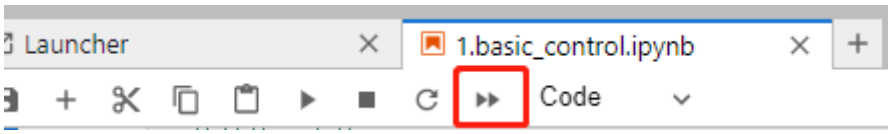
Open the jupyterlab client and find the code path:

```
DOGZILLA/Samples/2_Control/3.body_control.ipynb
```

By default g_ENABLE_CHINESE=False, if you need to display Chinese, please set g_ENABLE_CHINESE=True.

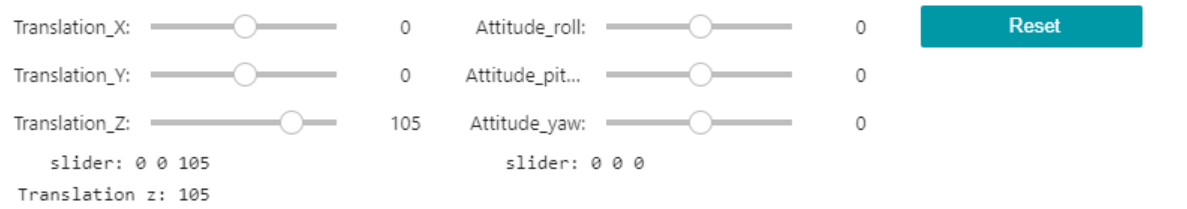
```
# 中文开关，默认为英文 Chinese switch. The default value is English
g_ENABLE_CHINESE = False

Name_widgets = {
    'Reset': ("Reset", "恢复默认"),
    'Translation_X': ("Translation_X", "前后平移"),
    'Translation_Y': ("Translation_Y", "左右平移"),
    'Translation_Z': ("Translation_Z", "身高调节"),
    'Attitude_roll': ("Attitude_roll", "滚转角"),
    'Attitude_pitch': ("Attitude_pitch", "俯仰角"),
    'Attitude_yaw': ("Attitude_yaw", "偏航角")
}
```

Click the following icon to run all cells, and then pull to the bottom to see the generated controls.



布局控件并显示 Layout widgets and display them

```
output = widgets.Output()
box_slider1 = widgets.interactive(on_slider_translation, x=slider_x, y=slider_y, z=slider_z)
box_slider2 = widgets.interactive(on_slider_attitude, roll=slider_roll, pitch=slider_pitch, yaw=slider_yaw)
box_h = widgets.HBox([box_slider1, box_slider2, button_reset])
box_display = widgets.VBox([box_h, output])
display(box_display)
```



The left slider bar controls the position translation of the robot dog body_ X controls the back translation, 'Translation_ Y controls the left and right translation of the back_ Z controls height.

身体位置平移 Translation of body position

```
# 滑块滑动事件处理  Slider event handling
def on_slider_translation(x, y, z):
    global g_translation_x, g_translation_y, g_translation_z
    print("   slider:", x, y, z)
    if g_translation_x != x:
        g_translation_x = x
        g_dog.translation('x', x)
        with output:
            print("Translation x:", x)
    if g_translation_y != y:
        g_translation_y = y
        g_dog.translation('y', y)
        with output:
            print("Translation y:", y)
    if g_translation_z != z:
        g_translation_z = z
        g_dog.translation('z', z)
        with output:
            print("Translation z:", z)
```

The middle slider bar controls the posture adjustment of the robot dog body, attitude_ Roll control robot dog roll angle attitude change, attitude_ Pitch control robot dog pitch angle attitude change, attitude_ Yaw controls the yaw angle and attitude change of robot dog.

机身姿态调整 Fuselage attitude adjustment

```
# 滑块滑动事件处理  Slider event handling
def on_slider_attitude(roll, pitch, yaw):
    global g_attitude_roll, g_attitude_pitch, g_attitude_yaw
    print("   slider:", roll, pitch, yaw)
    if g_attitude_roll != roll:
        g_attitude_roll = roll
        g_dog.attitude('r', roll)
        with output:
            print("Attitude roll:", roll)
    if g_attitude_pitch != pitch:
        g_attitude_pitch = pitch
        g_dog.attitude('p', pitch)
        with output:
            print("Attitude pitch:", pitch)
    if g_attitude_yaw != yaw:
        g_attitude_yaw = yaw
        g_dog.attitude('y', yaw)
        with output:
            print("Attitude yaw:", yaw)
```

The reset button on the right is to restore the robot dog to its default posture.

```
# 按键按下事件处理   Key press event processing
def on_button_clicked(b):
    with output:
        print("Button clicked:", b.description)
    if b.description == Name_widgets['Reset'][g_ENABLE_CHINESE]:
        slider_x.value = 0
        slider_y.value = 0
        slider_z.value = 105
        slider_roll.value = 0
        slider_pitch.value = 0
        slider_yaw.value = 0
```

1.4 Summary

In this course, we use JupyterLab controls to control the body posture of DOGZILLA.

When the foot end of the robot dog is not moving, the body posture can be controlled to change the X, y, Z translation, or the roll pitch yaw posture.