

ROS Hello World

一、程式目的說明

示範如何使用 ROS 2 (rclpy) 建立一個最基本的 Python Node，並透過 logger 在終端機中輸出 Hello World 訊息。

二、完整程式碼

```
import rclpy

def main(args=None):
    rclpy.init(args=args)    # 初始化 ROS

    # 創建一個叫做 hello_world_py 的 Node
    node = rclpy.create_node('hello_world_py')

    # 用 Node 的 get_logger() function 來印出 Hello World!
    node.get_logger().info('Hello World!')

    # 指定 rate 為 1Hz (每秒一次)
    rate = node.create_rate(1)

    # 讓 Node 持續運行直到 ROS 關閉 (Ctrl+C)
    while rclpy.ok():
        node.get_logger().info('Hello World in Loop!')
        rclpy.spin_once(node)  # 讓 Node 執行一次，會等待 1/rate 秒
    # 關閉 ROS

    rclpy.shutdown()

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

三、程式逐行講解

1. 匯入 rclpy 套件

```
import rclpy
```

rclpy 是 ROS 2 提供的 Python Client Library，用來建立 Node、Publisher、Subscriber 等 ROS 2 元件。

2. main 函式與 ROS 初始化

```
def main(args=None):  
    rclpy.init(args=args)
```

- `rclpy.init()`：初始化 ROS 2 系統
 - 必須在建立任何 Node 之前呼叫
 - `args` 用來接收 ROS 2 的命令列參數
-

3. 建立 Node

```
node = rclpy.create_node('hello_world_py')
```

- 建立一個名稱為 `hello_world_py` 的 ROS 2 Node
 - Node 名稱在同一個 ROS Graph 中必須是唯一的
-

4. 使用 Logger 輸出訊息

```
node.get_logger().info('Hello World!')
```

- `get_logger()`：取得該 Node 的 logger
 - `info()`：印出等級為 INFO 的訊息
 - 輸出會顯示在終端機，並包含時間與 Node 名稱
-

5. 設定執行頻率 (Rate)

```
rate = node.create_rate(1)
```

- 設定執行頻率為 1 Hz
- 表示每秒執行一次
- 常用於控制 while 迴圈的執行速度

⚠ 注意：在實務上通常會搭配 `rate.sleep()` 使用

6. 主迴圈 (while rclpy.ok())

```
while rclpy.ok():
```

- `rclpy.ok()` 會在 ROS 2 尚未被關閉時回傳 `True`
 - 按下 `Ctrl + C` 後會結束迴圈
-

7. 迴圈內的行為

```
node.get_logger().info('Hello World in Loop!')  
rclpy.spin_once(node)
```

- 每次迴圈都印出一行訊息
 - `spin_once(node)`：
 - 讓 Node 處理一次 callback
 - 會阻塞等待一小段時間（與 rate 有關）
-

8. 關閉 ROS

```
rclpy.shutdown()
```

- 正確釋放 ROS 2 資源
 - 程式結束前一定要呼叫
-

四、執行結果示意

終端機可能會看到類似以下輸出：

```
[INFO] [hello_world_py]: Hello World!  
[INFO] [hello_world_py]: Hello World in Loop!  
[INFO] [hello_world_py]: Hello World in Loop!  
...
```

每秒印出一次訊息，直到按下 `Ctrl + C`。

五、重點整理

- ROS 2 程式一定要先 `rclpy.init()`
 - Node 是 ROS 的基本運算單位
 - `get_logger()` 是標準的輸出方式（不要用 `print`）
 - `rclpy.spin_once()` 或 `rclpy.spin()` 用來讓 Node 開始運作
 - 程式結束前一定要 `rclpy.shutdown()`
-

六、延伸練習

1. 將 1 Hz 改成 10 Hz 觀察輸出變化
2. 改變 Node 名稱，並用 `ros2 node list` 查看