PX4 Autonomous Drone Mission Script

```
import asyncio
from mavsdk import System
from mavsdk.mission import MissionItem, MissionPlan, MissionItem

async def run():
    # Step 1: Connect to PX4 SITL
    drone = System()
    await drone.connect(system_address="udp://:14540")
    print("Waiting for drone to connect...")
    async for state in drone.core.connection_state():
        if state.is_connected:
            print(f"Drone discovered!")
            break

    # Step 2: Wait for Global Position Estimate
    print("Waiting for global position estimate...")
    async for health in drone.telemetry.health():
        if health.is_global_position_ok and health.is_home_position_ok:
            print("Global position estimate OK")
            break

    # Step 3: Define Mission Items
    mission_items = [
        MissionItem(
            47.398039859999997,
            8.5455725400000002,
            10,
            5,
            True,
            float('nan'),
            float('nan'),
            MissionItem.CameraAction.NONE,
            float('nan'),
            float('nan'),
            float('nan'),
            float('nan'),
            float('nan'),
            MissionItem.VehicleAction.NONE
        )
    ]

    # Step 4: Upload the mission
    mission_plan = MissionPlan(mission_items)
    await drone.mission.set_return_to_launch_after_mission(True)
    print("Uploading mission...")
    await drone.mission.upload_mission(mission_plan)
    print("Mission uploaded.")

    # Step 5: Arm and Start the Mission
    print("Arming...")
    await drone.action.arm()
```

```
    print("Starting mission...")
    await drone.mission.start_mission()

    # Step 6: Monitor Mission Progress
    async for mission_progress in drone.mission.mission_progress():
        print(f"Mission progress: {mission_progress.current}/{mission_progress.total}")
        if mission_progress.current == mission_progress.total:
            print("Mission completed!")
            break

    # Step 7: Wait until landed
    async for in_air in drone.telemetry.in_air():
        if not in_air:
            print("Landed!")
            break

    print("Disarming...")
    await drone.action.disarm()
    print("Mission finished.")

if __name__ == "__main__":
    asyncio.run(run())
```