



You are currently viewing the documentation for the Opentrons OT-1. To view documentation for the OT-2, click [here](#).

## Advanced Control

### Note:

The below features are designed for advanced users who wish to use the Opentrons API in their own Python environment (ie Jupyter). This page is not relevant for users only using the Opentrons App, because the features described below will not be accessible.

The robot module can be thought of as the parent for all aspects of the Opentrons API. All containers, instruments, and protocol commands are added to and controlled by robot.

```
'''
Examples in this section require the following
'''
from opentrons import robot, containers, instruments

plate = containers.load('96-flat', 'B1', 'my-plate')
tiprack = containers.load('tiprack-200ul', 'A1', 'my-rack')

pipette = instruments.Pipette(axis='b', max_volume=200, name='my-pipette')
```

## Head Speed

The maximum speed of the robot's head can be set using `robot.head_speed()`. The value we set the speed to is in millimeters-per-second (mm/sec).

```
robot.head_speed(5000)
```

### Note:

Setting the head speed to above 6000 mm/sec may cause your robot to “skip”, which means the motors will lose their grip and make a loud vibrating noise. We recommend you try out different speed values on your robot, and see what works and what doesn't.

## Homing

You can *home* the robot by calling `home()`. You can also specify axes. The robot will home immediately when this call is made.

```
robot.home()           # home the robot on all axis
robot.home('z')        # home the Z axis only
```



commands by calling `robot.commands()`, which returns a [Python list](#).

```
pipette.pick_up_tip(tiprack.wells('A1'))
pipette.drop_tip(tiprack.wells('A1'))

for c in robot.commands():
    print(c)
```

will print out...

```
Picking up tip from <Deck><Slot A1><Container my-rack><Well A1>
Drop_tip at <Deck><Slot A1><Container my-rack><Well A1>
```

## Clear Commands

We can erase the robot command history by calling `robot.clear_commands()`. Any previously created instruments and containers will still be inside robot, but the commands history is erased.

```
robot.clear_commands()
pipette.pick_up_tip(tiprack['A1'])
print('There is', len(robot.commands()), 'command')

robot.clear_commands()
print('There are now', len(robot.commands()), 'commands')
```

will print out...

```
There is 1 command
There are now 0 commands
```

## Comment

You can add a custom message to the list of command descriptions you see when running `robot.commands()`. This command is `robot.comment()`, and it allows you to print out any information you want at the point in your protocol

```
robot.clear_commands()

pipette.pick_up_tip(tiprack['A1'])
robot.comment("Hello, just picked up tip A1")

pipette.pick_up_tip(tiprack['A1'])
robot.comment("Goodbye, just dropped tip A1")

for c in robot.commands():
    print(c)
```

will print out...

```
Picking up tip from <Deck><Slot A1><Container my-rack><Well A1>
Hello, just picked up tip A1
Picking up tip from <Deck><Slot A1><Container my-rack><Well A1>
Goodbye, just dropped tip A1
```

## Get Containers

When containers are loaded, they are automatically added to the `robot`. You can see all currently held containers by calling `robot.get_containers()`, which returns a [Python list](#).



```
my-plate 96-flat
my-rack tiprack-200ul
```

## Get Instruments

When instruments are created, they are automatically added to the `robot`. You can see all currently held instruments by calling `robot.get_instruments()`, which returns a [Python list](#).

```
for axis, pipette in robot.get_instruments():
    print(pipette.name, axis)
```

will print out...

```
my-pipette B
```

## Reset

Calling `robot.reset()` will remove everything from the robot. Any previously added containers, pipettes, or commands will be erased.

```
robot.reset()
print(robot.get_containers())
print(robot.get_instruments())
print(robot.commands())
```

will print out...

```
[]
[]
[]
```



Sign Up For Our Newsletter

Email\*\*

**SUBMIT**

© OPENTRONS 2025