You are currently viewing the documentation for the Opentrons OT-1. To view documentation for the OT-2, click here.

Examples

All examples on this page assume the following containers and pipette:

```
from opentrons import robot, containers, instruments

plate = containers.load('96-flat', 'B1')
trough = containers.load('trough-12row', 'C1')

tiprack_1 = containers.load('tiprack-200ul', 'A1')
tiprack_2 = containers.load('tiprack-200ul', 'A2')

p200 = instruments.Pipette(
    axis="b",
    max_volume=200,
    tip_racks=[tiprack_2])
```

Basic Transfer

Moving 100uL from one well to another:

```
p200.transfer(100, plate.wells('A1'), plate.wells('B1'))
```

If you prefer to not use the `.transfer()` command, the following pipette commands will create the some results:

```
p200.pick_up_tip()
p200.aspirate(100, plate.wells('A1'))
p200.dispense(100, plate.wells('A1'))
p200.return_tip()
```

Loops

Loops in Python allows your protocol to perform many actions, or act upon many wells, all within just a few lines. The below example loops through the numbers `0` to `11`, and uses that loop's current value to transfer from all wells in a trough to each row of a plate:

```
# distribute 20uL from trough:A1 -> plate:row:1
# distribute 20uL from trough:A2 -> plate:row:2
# etc...

# ranges() starts at 0 and stops at 12, creating a range of 0-11
for i in range(12):
  p200.distribute(20, trough.wells(i), plate.rows(i))
```

and creating air gaps within the same tip. The below example will aspirate from five wells in the trough, while creating a air gap between each sample.

```
p200.pick_up_tip()

for well in trough.wells():
  p200.aspirate(5, well).air_gap(10)

p200.dispense(plate.wells('A1'))

p200.return_tip()
```

Dilution

This example first spreads a dilutent to all wells of a plate. It then dilutes 8 samples from the trough across the 8 columns of the plate.

```
p200.distribute(50, trough.wells('A12'), plate.wells())  # dilutent

# loop through each column
for i in range(8):

  # save the source well and destination column to variables
  source = trough.wells(i)
  column = plate.cols(i)

  # transfer 10uL of source to first well in column
  p200.transfer(10, source, column.wells('1'))

  # dilute the sample down the column
  p200.transfer(
    10, column.wells('1', to='11'), column.wells('2', to='12'),
    mix_after=(3, 25))
```

Plate Mapping

Deposit various volumes of liquids into the same plate of wells, and automatically refill the tip volume when it runs out.

```
# these uL values were created randomly for this example
water_volumes = [
  1,  2,  3,  4,  5,  6,  7,  8,
  9,  10, 11, 12, 13, 14, 15, 16,
  17, 18, 19, 20, 21, 22, 23, 24,
  25, 26, 27, 28, 29, 30, 31, 32,
  33, 34, 35, 36, 37, 38, 39, 40,
  41, 42, 43, 44, 45, 46, 47, 48,
  49, 50, 51, 52, 53, 54, 55, 56,
  57, 58, 59, 60, 61, 62, 63, 64,
  65, 66, 67, 68, 69, 70, 71, 72,
  73, 74, 75, 76, 77, 78, 79, 80,
  81, 82, 83, 84, 85, 86, 87, 88,
  89, 90, 91, 92, 93, 94, 95, 96
]

p200.distribute(water_volumes, trough.wells('A12'), plate)
```

The final volumes can also be read from a CSV, and opened by your protocol.

```
'''
  This example uses a CSV file saved on the same computer, formatted as follows,
  where the columns in the file represent the 8 columns of the plate,
  and the rows in the file represent the 12 rows of the plate,
  and the values represent the uL that must end up at that location
  41, 42, 43, 44, 45, 46, 47, 48,
  49, 50, 51, 52, 53, 54, 55, 56,
  57, 58, 59, 60, 61, 62, 63, 64,
  65, 66, 67, 68, 69, 70, 71, 72,
  73, 74, 75, 76, 77, 78, 79, 80,
  81, 82, 83, 84, 85, 86, 87, 88,
  89, 90, 91, 92, 93, 94, 95, 96,

'''

# open file with absolute path (will be different depending on operating system)
# file paths on Windows look more like 'C:\\path\\to\\your\\csv_file.csv'
with open('/path/to/your/csv_file.csv') as my_file:

    # save all volumes from CSV file into a list
    volumes = []

    # loop through each line (the plate's columns)
    for l in my_file.read().splitlines():
        # loop through each comma-separated value (the plate's rows)
        for v in l.split(','):
            volumes.append(float(v))  # save the volume

    # distribute those volumes to the plate
    p200.distribute(volumes, trough.wells('A1'), plate.wells())
```

Precision Pipetting

This example shows how to deposit liquid around the edge of a well.

```
p200.pick_up_tip()

# rotate around the edge of the well, dropping 10ul at a time
theta = 0.0
while p200.current_volume > 0:
    # we can move around a circle with radius (r) and theta (degrees)
    well_edge = plate.wells('B1').from_center(r=1.0, theta=theta, h=0.9)

    # combine a Well with a Vector in a tuple
    destination = (plate.wells('B1'), well_edge)
    p200.move_to(destination, strategy='direct')  # move straight there
    p200.dispense(10)

    theta += 0.314

p200.drop_tip()
```