

## EXAMPLE: How to build a dYdX Orderbook

Author: Lawrence Chiu ([lawrence@dydx.exchange](mailto:lawrence@dydx.exchange))

Twitter: <https://twitter.com/LawrenceChiu14>

Date: 4/8/2022 (updated 4/18/2022)

Tested on: Ubuntu Server 20.04.4 LTS, Windows 10 (See Appendix A), MacOS 12.3.1 (Monterey) (See Appendix B)

We will build the dYdX orderbook with 3 programs:

- 1) dydxob.py (the orderbook builder).
- 2) dydxtrades (the last-trade monitor), and
- 3) dydxob2.py (the orderbook displayer)

All programs are available at <https://github.com/chiwalfm/dydxexamples>

### PART 1: dydxob.py

1. First, create a 1GB ramdisk and mount it as /mnt/ramdisk/ as this program will write a LOT of files to it continuously. Set the permissions mask to 777.

```
$ sudo mkdir -p /mnt/ramdisk
$ sudo mount -t tmpfs -o rw,size=1G tmpfs /mnt/ramdisk
$ sudo chmod 777 /mnt/ramdisk
```

2. Next, start the dydxob.py program. Invoke the program with the <MARKET> parameter (if omitted, it defaults to 'BTC-USD'). You can get a list of markets from <https://api.dydx.exchange/v3/markets> Markets with status = 'POST\_ONLY' are not open for trading, and this program will not do anything. The -u option tells python not to buffer.

-u Force the stdout and stderr streams to be unbuffered. This option has no effect on the stdin stream.

```
$ nohup python3 -u dydxob.py BTC-USD > /mnt/ramdisk/dydxobBTC-USD.stdout 2>&1 &
```

3. This creates and continuously updates 'price' files in the directories <MARKET>/asks/ and <MARKET>/bids/. These price files each contain four elements: the 'offset' of the price, the 'size' at that price, and the 'date' (yyyy-mm-dd) and 'time' (hh:mm:ss) of the last update. The program logs to and automatically rotates the logfiles /mnt/ramdisk/dydxob<MARKET>.log so that each <MARKET> uses less than 5 MB.

```
vmware@ubuntu20041a:~$ cat /mnt/ramdisk/BTC-USD/asks/42863
12240989790 0 2022-04-10 12:17:37
vmware@ubuntu20041a:~$
```

## PART 2: dydxtrades.py

1. This step is optional. If you run this, it adds a “Last trade” line and if the last trade price matches the bid or ask, it will highlight it in red/green color. If you don’t care about this, you can skip this part.

```
vmware@ubuntu20041a:~/extra$ python3 -u dydxob2.py BTC-USD 10
2022-04-18 22:55:23 dydxob2.py
2022-04-18 22:55:23 Last trade: 2022-04-19T03:55:20.761Z 40687 SELL (0.0147)
BUY | ASK
40687.0 (1.7306) 12838168049 2022-04-18 22:55:22 | 40689.0 (6.0309) 12838167800 2022-04-18 22:55:21
40685.0 (1.1423) 12838165421 2022-04-18 22:55:14 | 40690.0 (0.4247) 12838167794 2022-04-18 22:55:21
40684.0 (1.1322) 12838165558 2022-04-18 22:55:14 | 40691.0 (1.4886) 12838166990 2022-04-18 22:55:18
40683.0 (1.4919) 12838168080 2022-04-18 22:55:22 | 40694.0 (0.6403) 12838164602 2022-04-18 22:55:11
40682.0 (0.3947) 12838168091 2022-04-18 22:55:22 | 40696.0 (16.7594) 12838167463 2022-04-18 22:55:20
40681.0 (0.7081) 12838167458 2022-04-18 22:55:20 | 40699.0 (27.9146) 12838164856 2022-04-18 22:55:12
40680.0 (0.5852) 12838163911 2022-04-18 22:55:10 | 40701.0 (0.8711) 12838166608 2022-04-18 22:55:18
40679.0 (5.4861) 12838167464 2022-04-18 22:55:20 | 40702.0 (0.1772) 12838168054 2022-04-18 22:55:22
40678.0 (1.8285) 12838165261 2022-04-18 22:55:13 | 40704.0 (5.8664) 12838157546 2022-04-18 22:54:54
40677.0 (0.1353) 12838164131 2022-04-18 22:55:10 | 40705.0 (3.2297) 12838164776 2022-04-18 22:55:12
maxbid : 40687.0
minask : 40689.0 (+2.0000) 0.0049%
bidvolume: 14.634900000000004
askvolume: 63.402899999999995
minoffset: 12838157546
maxoffset: 12838168091 (+10545)
```

2. Start the dydxtrades.py program. Invoke the program with the <MARKET> parameter (if omitted, it defaults to ‘BTC-USD’).

```
$ nohup python3 -u dydxtrades.py BTC-USD > /mnt/ramdisk/dydxtradesBTC-USD.stdout 2>&1 &
```

3. This creates and continuously updates the file <MARKET>/lasttrade with the ‘date/timestamp’, ‘price’ and ‘side’ (BUY or SELL) of the last trade. The program logs to and automatically rotates the logfiles /mnt/ramdisk/dydxtrades<MARKET>.log so that each <MARKET> uses less than 5 MB.

```
vmware@ubuntu20041a:~/extra/f$ cat /mnt/ramdisk/BTC-USD/lasttrade
2022-04-10T17:51:15.687Z 43056 SELL (0.0229)
vmware@ubuntu20041a:~/extra/f$
```

## PART 3: dydxob2.py

1. Finally, start the dydxob2.py program. Invoke the program with the <MARKET> and <DEPTH> parameters. <MARKET> must be first parameter, and <DEPTH> must be second. If <DEPTH> is omitted, it defaults to '10'. If <MARKET> and <DEPTH> are both omitted, it defaults to 'BTC-USD' and '10'. This requires a 103-column terminal to look good.

```
$ python3 -u dydxob2.py BTC-USD 10
```

2. This will display and continuously update the order book on your screen.

```
vmware@ubuntu20041a:~/extra/f$ python3 -u dydxob2.py BTC-USD 10
2022-04-10 12:59:42 Last trade: 2022-04-10T17:59:41.580Z 43119 BUY (0.0463)
Bid | Ask
43118 (5.2829) 12243272344 2022-04-10 12:59:40 | 43117 (1.1615) 12243272866 2022-04-10 12:59:41
43117 (1.6726) 12243270829 2022-04-10 12:59:38 | 43120 (1.1842) 12243272958 2022-04-10 12:59:41
43116 (1.141) 12243271365 2022-04-10 12:59:39 | 43121 (2.3684) 12243272795 2022-04-10 12:59:41
43115 (2.2525) 12243272118 2022-04-10 12:59:40 | 43122 (1.4324) 12243272816 2022-04-10 12:59:41
43114 (1.142) 12243270741 2022-04-10 12:59:38 | 43123 (2.1321) 12243272376 2022-04-10 12:59:40
43113 (0.4638) 12243270818 2022-04-10 12:59:38 | 43124 (1.6613) 12243272130 2022-04-10 12:59:40
43112 (0.1005) 12243270846 2022-04-10 12:59:38 | 43125 (0.1848) 12243272750 2022-04-10 12:59:41
43110 (0.3756) 12243272381 2022-04-10 12:59:40 | 43127 (0.1403) 12243272139 2022-04-10 12:59:40
43107 (3.807) 12243272572 2022-04-10 12:59:40 | 43129 (0.3062) 12243256096 2022-04-10 12:59:35
43106 (24.1628) 12243272961 2022-04-10 12:59:41 | 43130 (2.1548) 12243268747 2022-04-10 12:59:37
bidvolume: 40.4429
askvolume: 13.536
minoffset: 12243256096
maxoffset: 12243272961 (+16865)
2022-04-10 12:59:43 Last trade: 2022-04-10T17:59:41.580Z 43119 BUY (0.0463)
Bid | Ask
43118 (4.5509) 12243273306 2022-04-10 12:59:42 | 43119 (2.3457) 12243273472 2022-04-10 12:59:43
43117 (1.6726) 12243270829 2022-04-10 12:59:38 | 43120 (1.1842) 12243273299 2022-04-10 12:59:42
43116 (1.141) 12243271365 2022-04-10 12:59:39 | 43121 (1.1842) 12243273164 2022-04-10 12:59:42
43115 (2.2525) 12243272118 2022-04-10 12:59:40 | 43122 (1.4324) 12243272816 2022-04-10 12:59:41
43114 (1.1842) 12243270741 2022-04-10 12:59:38 | 43123 (1.7579) 12243273438 2022-04-10 12:59:43
43113 (1.1295) 12243273315 2022-04-10 12:59:42 | 43124 (0.4771) 12243273170 2022-04-10 12:59:42
43112 (0.1005) 12243270846 2022-04-10 12:59:38 | 43125 (0.1848) 12243272750 2022-04-10 12:59:41
43110 (0.3756) 12243272381 2022-04-10 12:59:40 | 43127 (0.1403) 12243272139 2022-04-10 12:59:40
43107 (2.727) 12243273262 2022-04-10 12:59:42 | 43129 (0.4593) 12243273700 2022-04-10 12:59:44
43106 (24.1628) 12243272961 2022-04-10 12:59:41 | 43130 (2.1548) 12243268747 2022-04-10 12:59:37
bidvolume: 39.286600000000005
askvolume: 11.167599999999999
minoffset: 12243256096
maxoffset: 12243273472 (+17376)
2022-04-10 12:59:44 Last trade: 2022-04-10T17:59:44.240Z 43118 SELL (0.2)
Bid | Ask
43118 (4.0754) 12243273786 2022-04-10 12:59:44 | 43119 (4.2246) 12243273729 2022-04-10 12:59:44
43117 (1.6726) 12243270829 2022-04-10 12:59:38 | 43120 (1.1842) 12243273299 2022-04-10 12:59:42
43116 (1.0276) 12243273815 2022-04-10 12:59:44 | 43121 (1.1842) 12243273164 2022-04-10 12:59:42
43115 (2.2525) 12243272118 2022-04-10 12:59:40 | 43122 (1.4324) 12243272816 2022-04-10 12:59:41
43114 (1.1842) 12243270741 2022-04-10 12:59:38 | 43123 (2.9421) 12243273674 2022-04-10 12:59:43
43113 (1.1295) 12243273315 2022-04-10 12:59:42 | 43124 (0.4771) 12243273170 2022-04-10 12:59:42
43112 (0.1005) 12243270846 2022-04-10 12:59:38 | 43125 (0.1848) 12243272750 2022-04-10 12:59:41
43110 (0.3756) 12243272381 2022-04-10 12:59:40 | 43127 (0.1403) 12243272139 2022-04-10 12:59:40
43107 (2.727) 12243273262 2022-04-10 12:59:42 | 43129 (0.4593) 12243273700 2022-04-10 12:59:44
43106 (24.1628) 12243272961 2022-04-10 12:59:41 | 43130 (2.1548) 12243268747 2022-04-10 12:59:37
bidvolume: 38.7077
askvolume: 14.383799999999999
minoffset: 12243268747
maxoffset: 12243273815 (+5068)
2022-04-10 12:59:45 Last trade: 2022-04-10T17:59:45.165Z 43119 BUY (0.0232)
Bid | Ask
43118 (0.5349) 12243274291 2022-04-10 12:59:45 | 43119 (6.9283) 12243274307 2022-04-10 12:59:45
43117 (1.6726) 12243270829 2022-04-10 12:59:38 | 43120 (1.1842) 12243273299 2022-04-10 12:59:42
43116 (2.2118) 12243274164 2022-04-10 12:59:44 | 43121 (1.1842) 12243273164 2022-04-10 12:59:42
43115 (2.2525) 12243272118 2022-04-10 12:59:40 | 43122 (1.452) 12243274362 2022-04-10 12:59:45
43114 (2.3684) 12243274028 2022-04-10 12:59:44 | 43123 (0.4771) 12243273170 2022-04-10 12:59:42
43113 (1.1295) 12243274358 2022-04-10 12:59:45 | 43124 (0.4771) 12243273170 2022-04-10 12:59:42
43110 (0.3756) 12243272381 2022-04-10 12:59:40 | 43125 (0.1848) 12243272750 2022-04-10 12:59:41
43107 (3.967) 12243274089 2022-04-10 12:59:44 | 43127 (0.1403) 12243272139 2022-04-10 12:59:40
43106 (24.1628) 12243272961 2022-04-10 12:59:41 | 43129 (0.4593) 12243273700 2022-04-10 12:59:44
43105 (1.3838) 12243271774 2022-04-10 12:59:39 | 43130 (2.1548) 12243268747 2022-04-10 12:59:37
bidvolume: 40.0689
askvolume: 17.107100000000003
minoffset: 12243268747
maxoffset: 12243274362 (+5615)
last trade
bid/ask is highlighted if it matches the last trade
bidvolume = sum of all bid sizes
askvolume = ask sizes
minoffset = lowest offset
maxoffset = highest offset and delta between highest and lowest
```

3. (Optional) There are a few optional parameters to customize the display to your preference.
  - a. compact – Removes 'offset' values from the display (67-column terminal required)
  - b. ultracompact – Removes 'offset' and year part of the 'date' values (39-column terminal required)

- c. noansi – Removes colored values (this must be the last parameter)

```
vmware@ubuntu20041a:~/extra/f$ python3 -u dydxob2.py BTC-USD 10 compact
2022-04-10 13:28:18 2022-04-10T18:28:18.381Z 43206 SELL (0.2)
Bid | Ask
43206 (4.6253) 04-10 13:28:18 | 43207 (1.5398) 04-10 13:28:18
43205 (1.7342) 04-10 13:28:16 | 43209 (1.1842) 04-10 13:28:17
43204 (1.1842) 04-10 13:28:16 | 43210 (3.2944) 04-10 13:28:18
43203 (2.4384) 04-10 13:28:18 | 43211 (1.4844) 04-10 13:28:17
43202 (1.16) 04-10 13:28:16 | 43212 (0.2636) 04-10 13:28:17
43201 (1.0909) 04-10 13:28:17 | 43213 (0.2179) 04-10 13:28:17
43200 (2.2496) 04-10 13:28:16 | 43214 (0.7333) 04-10 13:28:17
43199 (3.6899) 04-10 13:28:17 | 43216 (0.6727) 04-10 13:28:18
43196 (0.7406) 04-10 13:28:17 | 43217 (5.3617) 04-10 13:28:18
43195 (0.6412) 04-10 13:28:15 | 43218 (0.3056) 04-10 13:28:16
bidvolume: 19.554300000000005
askvolume: 15.057600000000003
minoffset: 12244538054
maxoffset: 12244540983 (+2929)
```

```
vmware@ubuntu20041a:~/extra/f$ python3 -u dydxob2.py BTC-USD 10 ultracompact
04-10T18:28:58.481Z 43197 BUY (0.0462)
Bid | Ask
43196 (6.932) | 43197 (3.9598)
43195 (2.9314) | 43198 (1.1842)
43194 (3.4008) | 43199 (1.1842)
43193 (1.1842) | 43200 (0.7081)
43192 (0.1274) | 43201 (0.55)
43191 (1.1842) | 43207 (0.5206)
43190 (0.6459) | 43208 (0.1389)
43187 (27.5577) | 43210 (0.463)
43186 (2.7433) | 43211 (0.3002)
43185 (29.2198) | 43212 (0.6019)
bidvolume: 75.9267
askvolume: 9.610899999999999
minoffset: 12244551304
maxoffset: 12244560171 (+8867)
```

```
vmware@ubuntu20041a:~/extra/f$ python3 -u dydxob2.py BTC-USD 10 ultracompact noansi
04-10T18:29:23.831Z 43203 SELL (0.1)
Bid | Ask
43203 (4.1436) | 43204 (0.6842)
43202 (2.0784) | 43205 (1.1311)
43201 (1.6726) | 43206 (1.1842)
43200 (4.7994) | 43207 (0.4528)
43199 (1.1842) | 43210 (0.6158)
43198 (4.3342) | 43211 (0.3002)
43195 (0.563) | 43212 (3.6438)
43194 (2.3556) | 43213 (0.4957)
43192 (1.1508) | 43217 (1.7619)
43191 (0.278) | 43218 (2.0317)
bidvolume: 22.559799999999996
askvolume: 12.301400000000001
minoffset: 12244558946
maxoffset: 12244572834 (+13888)
```

## PART 4: (Optional) Launching multiple orderbooks

1. You can launch this program for multiple dYdX markets with this simple command.

```
for s1 in 1INCH-USD AAVE-USD ADA-USD ALGO-USD ATOM-USD AVAX-USD BCH-USD BTC-USD COMP-USD \  
    CRV-USD DOGE-USD DOT-USD ENJ-USD EOS-USD ETC-USD ETH-USD FIL-USD LINK-USD LTC-USD \  
    MATIC-USD MKR-USD SNX-USD SOL-USD SUSHI-USD UMA-USD UNI-USD XLM-USD XMR-USD YFI-USD \  
    ZEC-USD ZRX-USD  
do  
    nohup python3 -u dydxob.py $s1 > /mnt/ramdisk/dydxob$s1.stdout 2>&1 &  
    nohup python3 -u dydxtrades.py $s1 > /mnt/ramdisk/dydxob$s1.stdout 2>&1 &  
    sleep 1  
done
```

## PART 5: (Optional) Terminating all programs and unmount the ramdisk

1. You can kill all dydxob.py processes with this command:

```
$ kill -TERM `ps -ef | grep ' dydxob.py ' | grep -v ' grep '`
```

2. Likewise for dydxtrades.py:

```
$ kill -TERM `ps -ef | grep ' dydxtrades.py ' | grep -v ' grep '`
```

3. Finally, once all processes are terminated, you can unmount the ramdisk and reclaim the memory with this command. See Appendix B for MacOS equivalent.

```
$ sudo umount /mnt/ramdisk
```

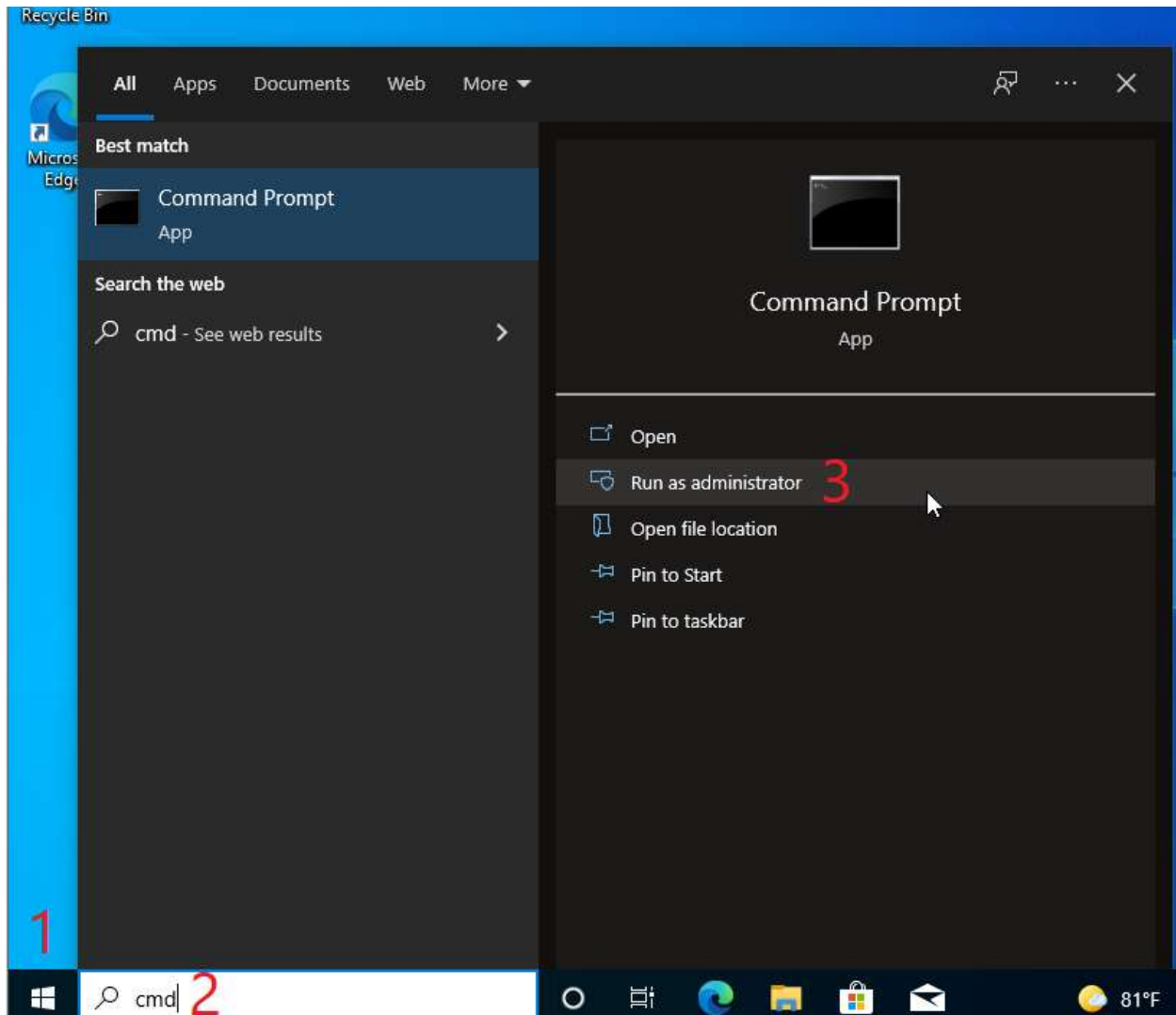


## APPENDIX A: Windows 10

### (Win10) PART 1: Install Windows Subsystem for Linux (WSL)

#### 1) Start Command Prompt as Administrator:

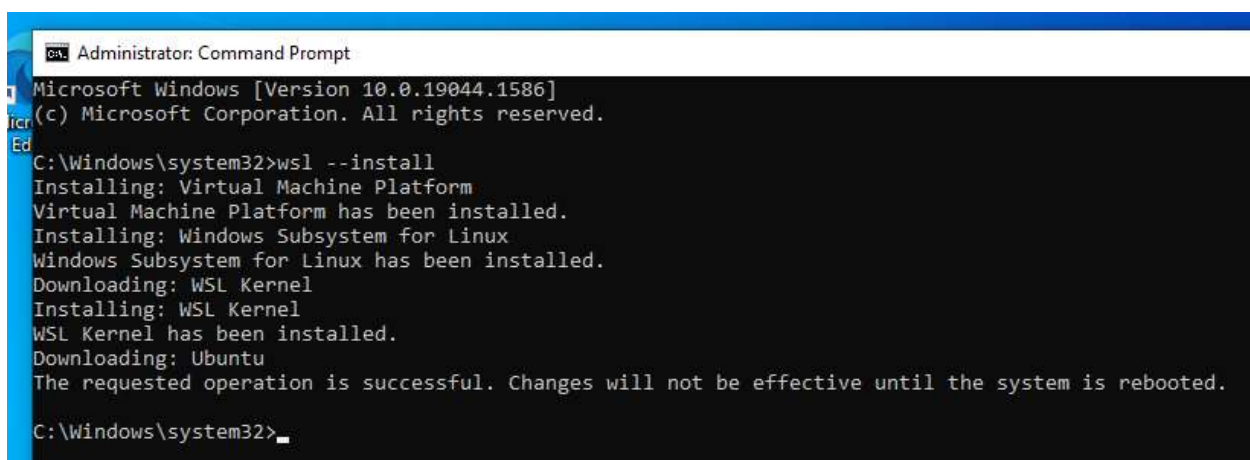
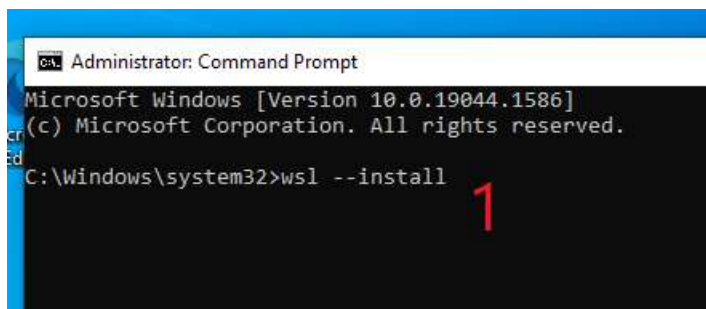
- Click the Windows button.
- Type 'cmd'.
- Click on 'Run as administrator'.



d. Click Yes.



2) Type `wsl --install` and wait.

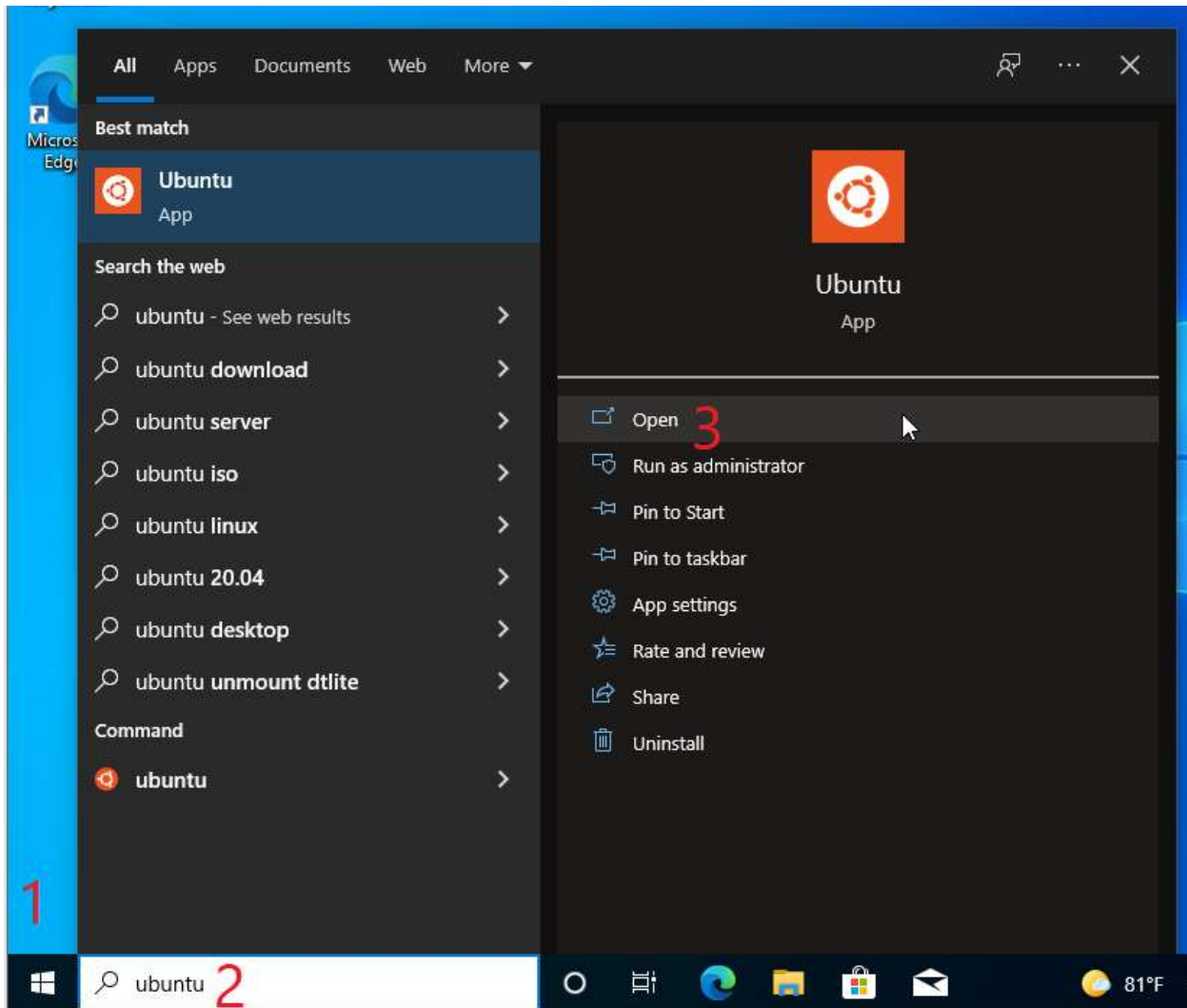


3) Reboot.

## (Win10) PART 2: Launch WSL

### 1) Start Ubuntu:

- a. Click the Windows button.
- b. Type 'ubuntu'.
- c. Click on 'Open'.





- 2) (First time) Enter username and password (twice).

```
lawrencedydx@DESKTOP-E67DGDB: ~  
Installing, this may take a few minutes...  
Please create a default UNIX user account. The username does not need to match your Windows username.  
For more information visit: https://aka.ms/wslusers  
Enter new UNIX username: lawrencedydx  
New password:  
Retype new password:  
passwd: password updated successfully  
Installation successful!  
  
To run a command as administrator (user "root"), use "sudo <command>".  
See "man sudo_root" for details.  
  
Welcome to Ubuntu 20.04 LTS (GNU/Linux 5.10.16.3-microsoft-standard-WSL2 x86_64)  
  
* Documentation:  https://help.ubuntu.com  
* Management:    https://landscape.canonical.com  
* Support:        https://ubuntu.com/advantage  
  
System information as of Mon Apr 11 18:57:55 CDT 2022  
  
System load:  0.31           Processes:            8  
Usage of /:   0.4% of 250.98GB Users logged in:       0  
Memory usage: 3%           IPv4 address for eth0: 172.30.252.228  
Swap usage:   0%  
  
0 updates can be installed immediately.  
0 of these updates are security updates.  
  
The list of available updates is more than a week old.  
To check for new updates run: sudo apt update  
  
This message is shown once once a day. To disable it please create the  
/home/lawrencedydx/.hushlogin file.  
lawrencedydx@DESKTOP-E67DGDB:~$
```

## (Win10) PART 3: Update WSL

- 1) Update WSL with the following commands:

```
$ sudo apt-get update  
$ sudo apt-get upgrade
```

- 2) Install required components:

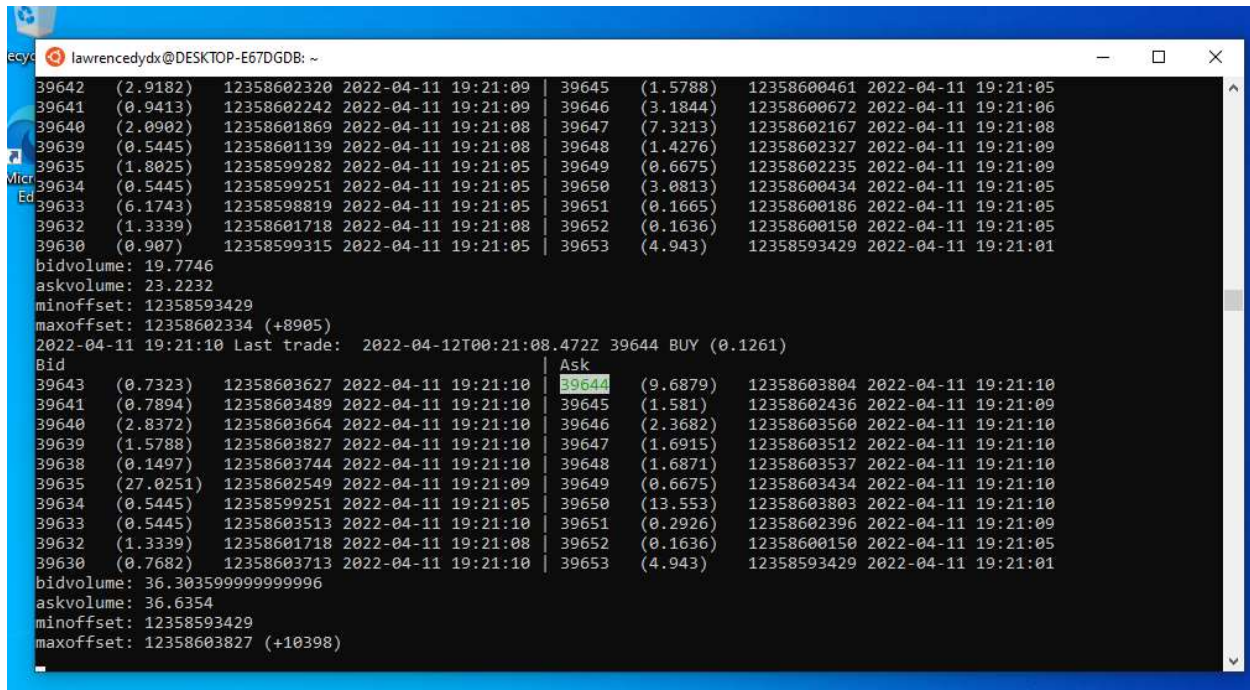
```
$ sudo apt-get install python3-pip  
$ pip3 install websocket-client
```

## (Win10) PART 4: Download dYdX order book files

- 1) You can download the 3 dYdX order book files with the following commands:

```
$ wget https://raw.githubusercontent.com/chiwalfrm/dydxexamples/main/dydxob.py  
$ wget https://raw.githubusercontent.com/chiwalfrm/dydxexamples/main/dydxtrades.py  
$ wget https://raw.githubusercontent.com/chiwalfrm/dydxexamples/main/dydxob2.py
```

2) The rest of the guide follows the Linux version. Here is a picture of it running on Windows.

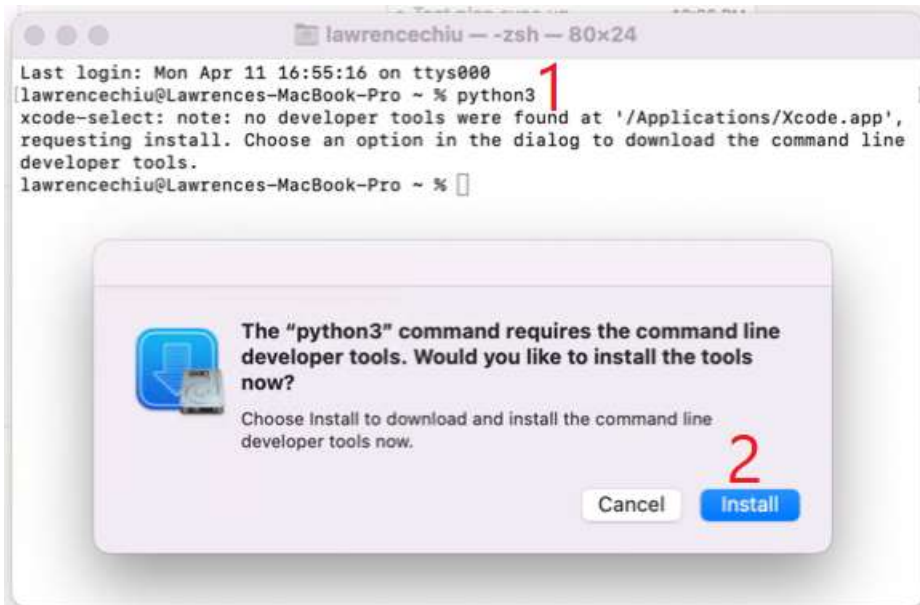


```
ecys lawrencedydx@DESKTOP-E67DGD8: ~
39642 (2.9182) 12358602320 2022-04-11 19:21:09 | 39645 (1.5788) 12358600461 2022-04-11 19:21:05
39641 (0.9413) 12358602242 2022-04-11 19:21:09 | 39646 (3.1844) 12358600672 2022-04-11 19:21:06
39640 (2.0902) 12358601869 2022-04-11 19:21:08 | 39647 (7.3213) 12358602167 2022-04-11 19:21:08
39639 (0.5445) 12358601139 2022-04-11 19:21:08 | 39648 (1.4276) 12358602327 2022-04-11 19:21:09
39635 (1.8025) 12358599282 2022-04-11 19:21:05 | 39649 (0.6675) 12358602235 2022-04-11 19:21:09
39634 (0.5445) 12358599251 2022-04-11 19:21:05 | 39650 (3.0813) 12358600434 2022-04-11 19:21:05
39633 (6.1743) 12358598819 2022-04-11 19:21:05 | 39651 (0.1665) 12358600186 2022-04-11 19:21:05
39632 (1.3339) 12358601718 2022-04-11 19:21:08 | 39652 (0.1636) 12358600150 2022-04-11 19:21:05
39630 (0.907) 12358599315 2022-04-11 19:21:05 | 39653 (4.943) 12358593429 2022-04-11 19:21:01
bidvolume: 19.7746
askvolume: 23.2232
minoffset: 12358593429
maxoffset: 12358602334 (+8905)
2022-04-11 19:21:10 Last trade: 2022-04-12T00:21:08.472Z 39644 BUY (0.1261)
Bid | Ask
39643 (0.7323) 12358603627 2022-04-11 19:21:10 | 39644 (9.6879) 12358603804 2022-04-11 19:21:10
39641 (0.7894) 12358603489 2022-04-11 19:21:10 | 39645 (1.581) 12358602436 2022-04-11 19:21:09
39640 (2.8372) 12358603664 2022-04-11 19:21:10 | 39646 (2.3682) 12358603560 2022-04-11 19:21:10
39639 (1.5788) 12358603827 2022-04-11 19:21:10 | 39647 (1.6915) 12358603512 2022-04-11 19:21:10
39638 (0.1497) 12358603744 2022-04-11 19:21:10 | 39648 (1.6871) 12358603537 2022-04-11 19:21:10
39635 (27.0251) 12358602549 2022-04-11 19:21:09 | 39649 (0.6675) 12358603434 2022-04-11 19:21:10
39634 (0.5445) 12358599251 2022-04-11 19:21:05 | 39650 (13.553) 12358603803 2022-04-11 19:21:10
39633 (0.5445) 12358603513 2022-04-11 19:21:10 | 39651 (0.2926) 12358602396 2022-04-11 19:21:09
39632 (1.3339) 12358601718 2022-04-11 19:21:08 | 39652 (0.1636) 12358600150 2022-04-11 19:21:05
39630 (0.7682) 12358603713 2022-04-11 19:21:10 | 39653 (4.943) 12358593429 2022-04-11 19:21:01
bidvolume: 36.303599999999996
askvolume: 36.6354
minoffset: 12358593429
maxoffset: 12358603827 (+10398)
```

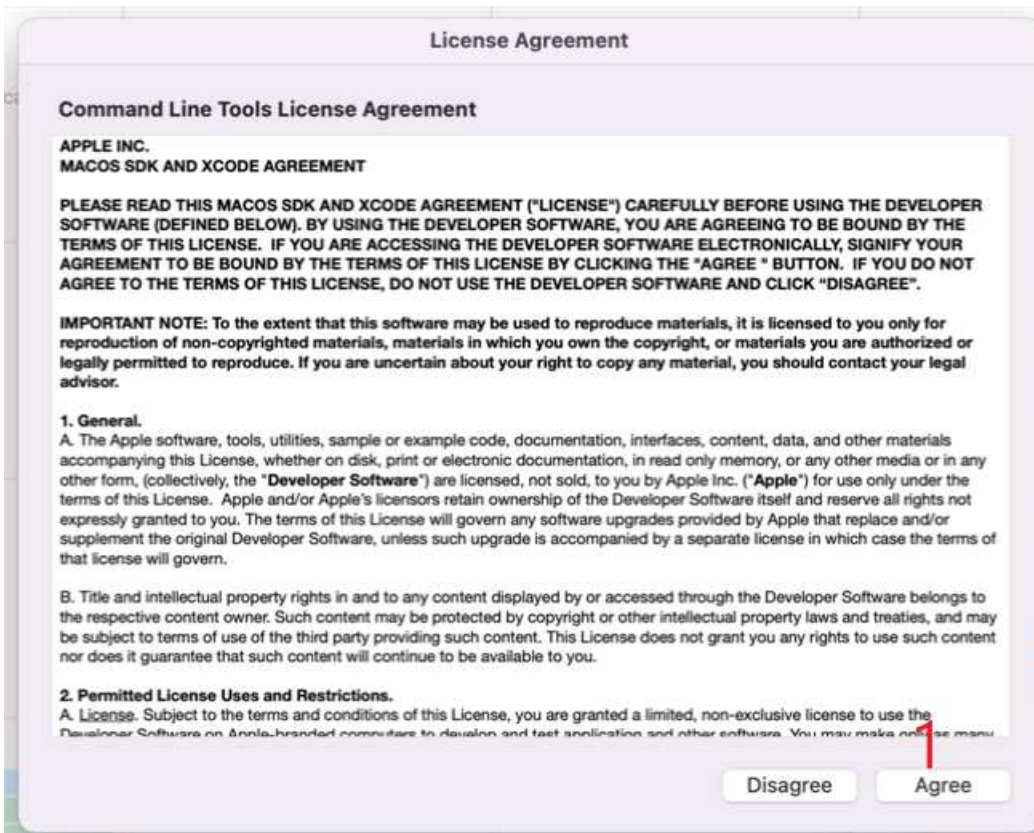
## APPENDIX B: MacOS

### (MacOS) PART 1: Install Python3

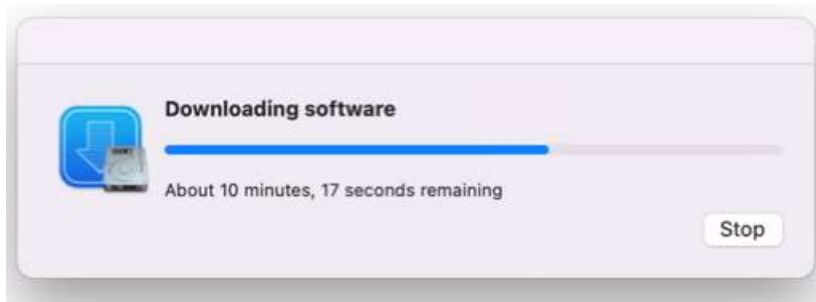
- 1) Launch Terminal and type 'python3'. Then click Install.



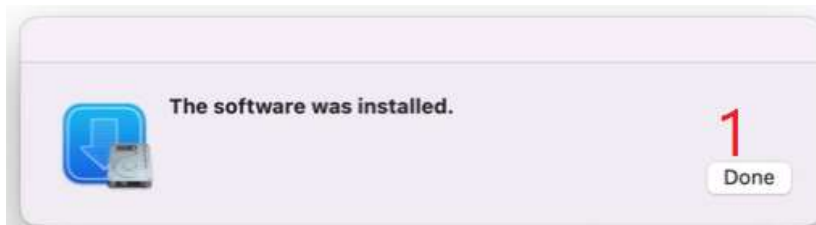
- 2) Click Agree.



3) Wait.



4) Click Done.



## (MacOS) PART 2: Install components

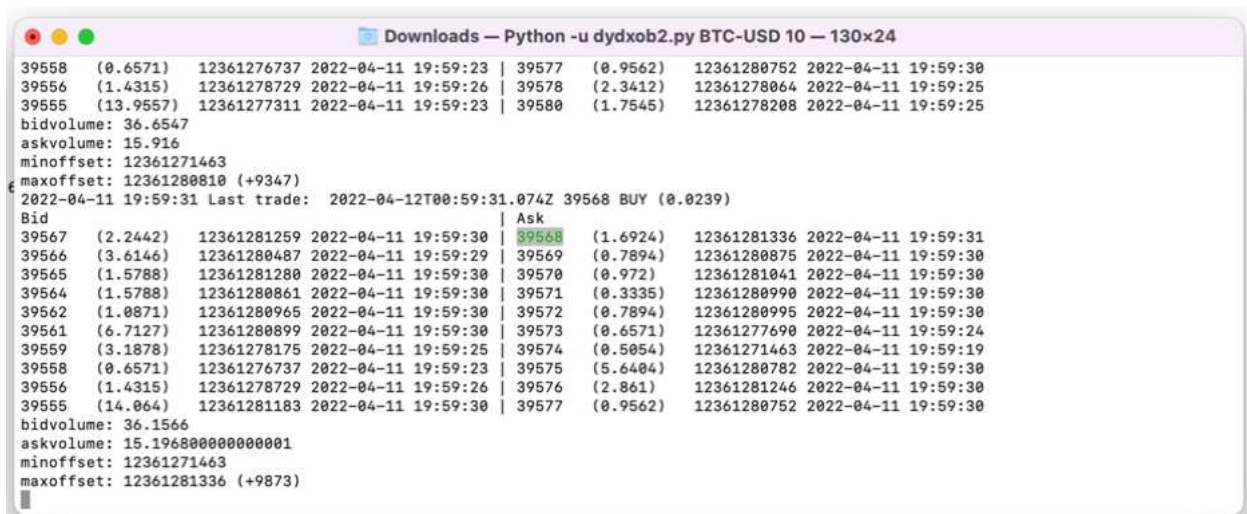
1) Install required components:

```
$ pip3 install websocket-client
```

2) The rest of the guide follows the Linux version except the command to create the ramdisk is different:

```
$ diskutil erasevolume HFS+ "RAMDisk" `hdiutil attach -nomount ram://2097152`
```

3) Here is a picture of it running on MacOS.



### (MacOS) PART 3: (Optional) Terminating all programs and unmount the ramdisk

- 1) This part follows the Linux version except the command to unmount the ramdisk is different:

```
$ hdiutil detach /Volumes/RAMdisk
```

<END>