School of Computer Engineering and Science SHU

Blockchain Technology and Data Economics

(08696017 1000)

Practical Report

Practise 2

Blockchain Programming with Python

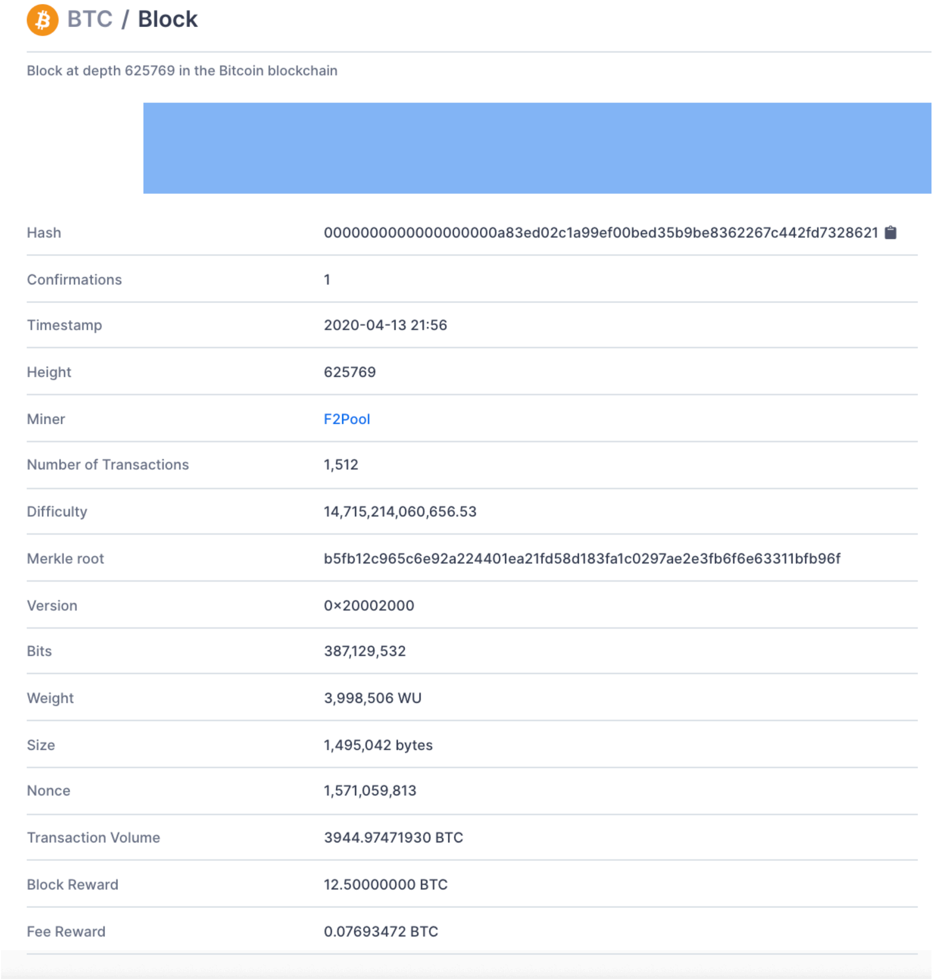
Name: Lin Yijun ID: 18120189

Date: 15 Apr. 2020

Contents

1 Introduction

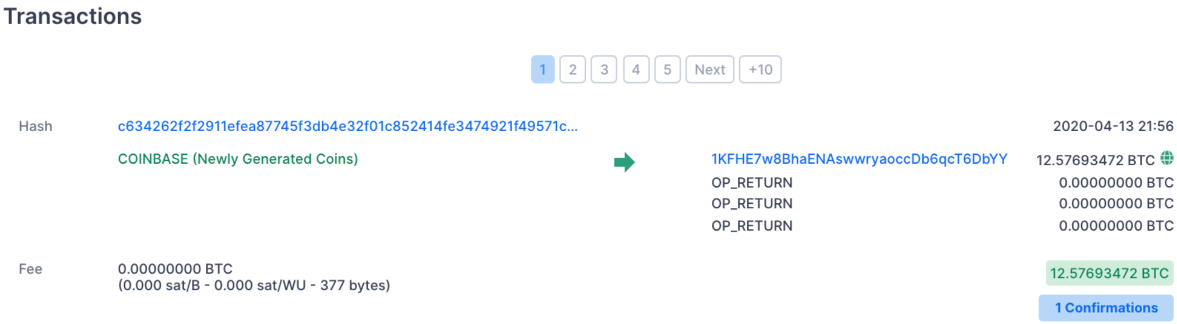
Blockchain.info is one of the most popular blockchain and bitcoin network explorers and wallet providers. From the web, you can view the block level and see all the transactions that have happened. For example, by going to this particular block—that is, Block #625769—you can see all the transactions, as well as some other information, as shown in the following screenshot:



Block at depth 625769 in the Bitcoin blockchain

Hash	0000000000000000000a83ed02c1a99ef00bed35b9be8362267c442fd7328621
Confirmations	1
Timestamp	2020-04-13 21:56
Height	625769
Miner	F2Pool
Number of Transactions	1,512
Difficulty	14,715,214,060,656.53
Merkle root	b5fb12c965c6e92a224401ea21fd58d183fa1c0297ae2e3fb6f6e63311bfb96f
Version	0×20002000
Bits	387,129,532
Weight	3,998,506 WU
Size	1,495,042 bytes
Nonce	1,571,059,813
Transaction Volume	3944.97471930 BTC
Block Reward	12.50000000 BTC
Fee Reward	0.07693472 BTC

The following screenshot shows the statistical data (DATA | Stats). This is great, and useful as well; however, for a developer building applications or performing analysis based on this data, it is important to get this data programmatically:



Hash c634262f2f2911efea87745f3db4e32f01c852414fe3474921f49571c... 2020-04-13 21:56

COINBASE (Newly Generated Coins) → 1KFHE7w8BhaENAswwryaoccDb6qcT6DbYY 12.57693472 BTC

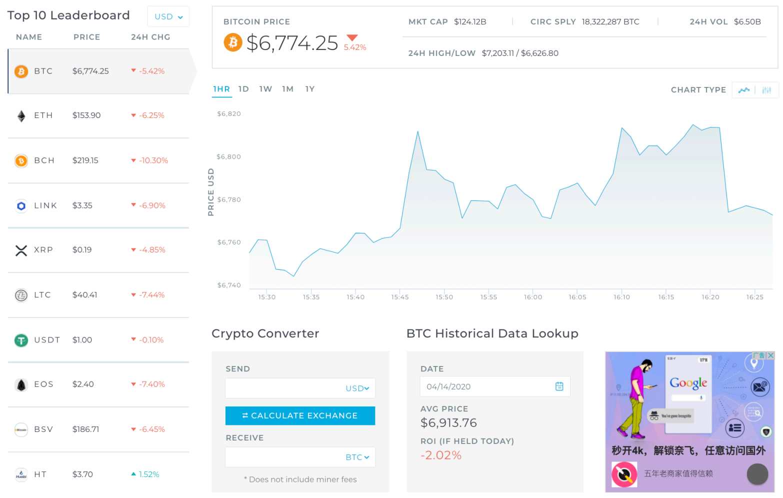
OP_RETURN 0.00000000 BTC
OP_RETURN 0.00000000 BTC
OP_RETURN 0.00000000 BTC

Fee 0.00000000 BTC
(0.000 sat/B - 0.000 sat/WU - 377 bytes)

12.57693472 BTC

1 Confirmations

The following screenshot shows the market data (DATA | Markets):



Top 10 Leaderboard USD

NAME	PRICE	24H CHG
BTC	$6,774.25	▼ -5.42%
ETH	$153.90	▼ -6.25%
BCH	$219.15	▼ -10.30%
LINK	$3.35	▼ -6.90%
XRP	$0.19	▼ -4.85%
LTC	$40.41	▼ -7.44%
USDT	$1.00	▼ -0.10%
EOS	$2.40	▼ -7.40%
BSV	$186.71	▼ -6.45%
HT	$3.70	▲ 1.52%

BITCOIN PRICE $6,774.25 ▼ 5.42%

MKT CAP $124.12B | CIRC SPLY 18,322,287 BTC | 24H VOL $6.50B

24H HIGH/LOW $7,203.11 / $6,626.80

Crypto Converter

SEND USD

CALCULATE EXCHANGE

RECEIVE BTC

* Does not include miner fees

BTC Historical Data Lookup

DATE 04/14/2020

AVG PRICE $6,913.76

ROI (IF HELD TODAY) -2.02%

2 Materials and Methods

2.1 Installing the Blockchain.info Python library

The following are the steps to install the blockchain Python library:

1. Open the command-line program on your computer.
2. Run the *pip install blockchain* command to install the blockchain library.

```
pip3 install blockchain
```

2.2 Getting bitcoin exchange rates from Blockchain.info

The following steps shows the method for bitcoin exchange rates:

1. First, import the exchangerates classes from the blockchain library:

```
#import blockchain library
from blockchain import exchangerates
```

2. Exchange rates define a get_ticker method, which returns the exchange rates data in a dictionary object. Call this method and save the resulting object. The ticker dictionary object that we have has currency symbols as keys:

```
# get the Bitcoin rates in various currencies
ticker = exchangerates.get_ticker()
```

3. By running over these keys, data about the various rates can be pulled. For example, the latest bitcoin rates can be obtained in each currency by getting the p15min minimum value:

```
# print the Bitcoin price for every currency
print(”15 minutes delayed market Bitcoin Prices in various currencies:”)
for k in ticker:
    print(k, ticker[k].p15min)
```

The following code shows how to do this for the list of currencies and the equivalent bitcoin rate for those currencies at that moment or from the last 15 minutes:

```
# print the Bitcoin price for every currency
print(”the most recent market Bitcoin Latest Prices in various currencies:”)
for k in ticker:
    print(k, ticker[k].last)
```

4. A particular currency can also be converted to bitcoin. For example, you can pass the to_btc method and pass in the currency and the amount that we want to convert to btc, and get the result as bitcoin. The following code shows how to do this for a value of 100 euros:

```
# getting Bitcoin value for a particular amount and currency
btc = exchangerates.to_btc(’EUR’, 100)
print(”\n100 euros in Bitcoin: %s” % btc)
```

The following code shows how to do this for a value of 100 CNY:

```
# getting Bitcoin value for a particular amount and currency
btc = exchangerates.to_btc(’CNY’, 100)
print(”\n100 CNY in Bitcoin: %s” % btc)
```

2.3 Statistics

The next class from the bitcoin blockchain library is called statistics. There are a number of methods that can be called to get a wide variety of blockchain stats data.

You can call the different methods as follows:

1. Import the relevant class, call the get method on statistics, and save that object. For example, to get the bitcoin trade volume, we should get the trade_volume_btc property from the stats object that was created, as shown in the following code:

```
# import blockchain library
from blockchain import statistics
# get the stats object
stats = statistics.get()
# get and print Bitcoin trade volume
print(”Bitcoin Trade Volume: %s\n” % stats.trade_volume_btc)
```

2. To get the total bitcoins mined, call the btc_mined property on stats object, as shown here:

```
# get and print Bitcoin mined
print(”Bitcoin mined: %s\n” % stats.btc_mined)
```

3. To get the bitcoin market price, use the stats class, call the market price and append that with the particular currency:

```
# get and print Bitcoin market price i®n usd
print(”Bitcoin market price: %s\n” % stats.market_price_usd)
```

2.4 Block explorer methods

For block explorer methods, start by importing the relevant classes from the blockchain library. To get a particular block, call the get_block method as shown in the following code. It expects a block to be passed in as the parameter.

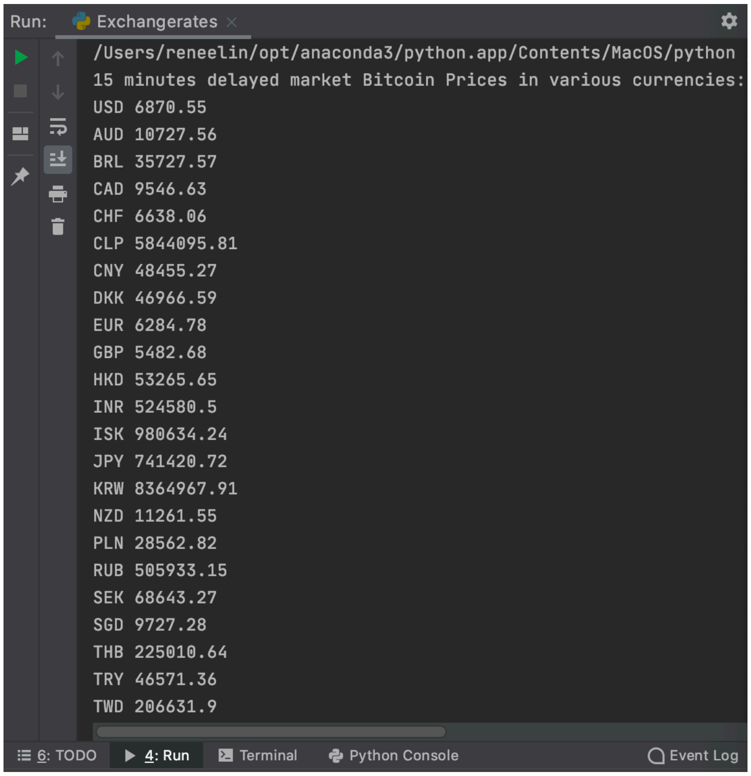
```
# import blockchain library
from blockchain import blockexplorer
# get a particular block
block = blockexplorer.get_block(’’)
```

Now let’s get some information about the block. For example, the block fee, block size, and block transactions can be obtained by using fee, size, and transactions properties respectively on the block object created, as shown in the following code:

```
# import blockchain library
from blockchain import blockexplorer
# get a particular block
block = blockexplorer.get_block(’0000000000000000000a83ed02c1a99ef00bed35b9be8362267c442fd7328621’)
print(”Block Fee: %s\n” % block.fee)
print(”Block Size: %s\n” % block.size)
print(”Block Transactions: %s\n” % block.transactions)
# get the latest block
block = blockexplorer.get_latest_block()
lbhash = block.hash
block = blockexplorer.get_block(lbhash)
print(”Latest Block Fee: %s\n” % block.fee)
print(”Latest Block Size: %s\n” % block.size)
print(”Latest Block Transactions: %s\n” % block.transactions)
```

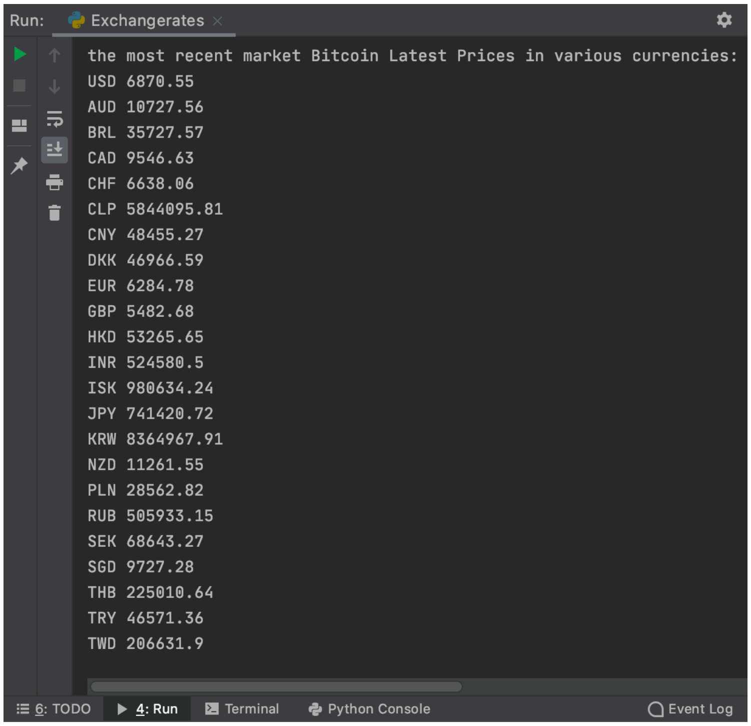
3 Results

The following screenshot shows the list of currencies and the equivalent bitcoin rate for those currencies from the last 15 minutes:



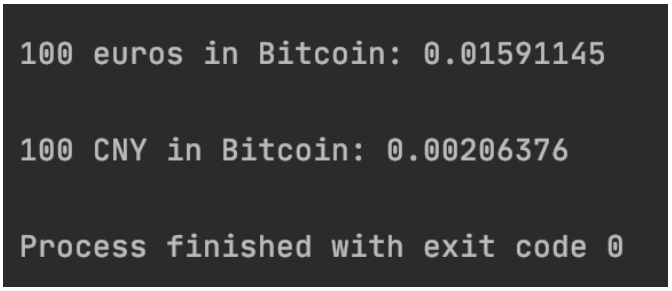
```
/Users/reneelin/opt/anaconda3/python.app/Contents/MacOS/python
15 minutes delayed market Bitcoin Prices in various currencies:
USD 6870.55
AUD 10727.56
BRL 35727.57
CAD 9546.63
CHF 6638.06
CLP 5844095.81
CNY 48455.27
DKK 46966.59
EUR 6284.78
GBP 5482.68
HKD 53265.65
INR 524580.5
ISK 980634.24
JPY 741420.72
KRW 8364967.91
NZD 11261.55
PLN 28562.82
RUB 505933.15
SEK 68643.27
SGD 9727.28
THB 225010.64
TRY 46571.36
TWD 206631.9
```

The following screenshot shows the list of currencies and the equivalent bitcoin rate for those currencies at that moment:



```
the most recent market Bitcoin Latest Prices in various currencies:
USD 6870.55
AUD 10727.56
BRL 35727.57
CAD 9546.63
CHF 6638.06
CLP 5844095.81
CNY 48455.27
DKK 46966.59
EUR 6284.78
GBP 5482.68
HKD 53265.65
INR 524580.5
ISK 980634.24
JPY 741420.72
KRW 8364967.91
NZD 11261.55
PLN 28562.82
RUB 505933.15
SEK 68643.27
SGD 9727.28
THB 225010.64
TRY 46571.36
TWD 206631.9
```

The following screenshot shows the outputs for 100 euros and 100 CNY in bitcoin:

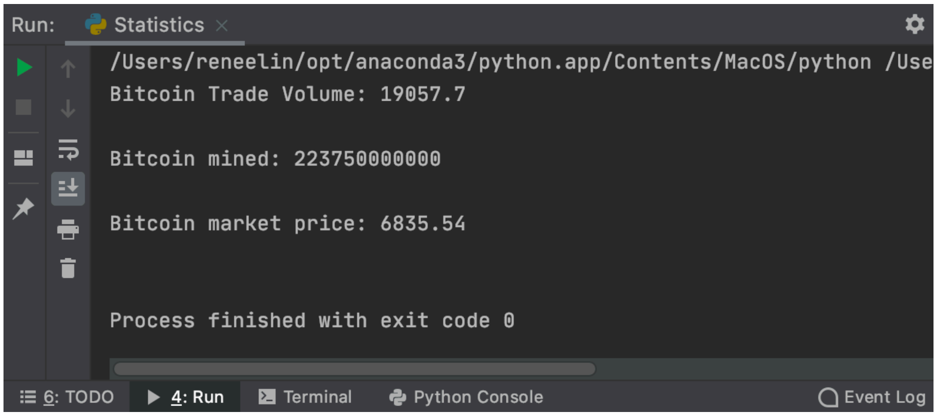


```
100 euros in Bitcoin: 0.01591145

100 CNY in Bitcoin: 0.00206376

Process finished with exit code 0
```

The following screenshot shows the bitcoin trade volume, the output of the number of bitcoins mined and the current bitcoin price in US dollars:



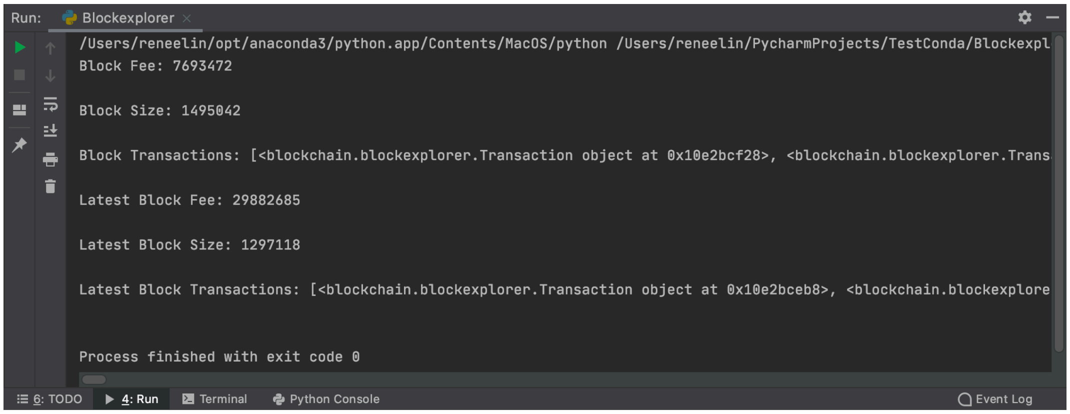
```
/Users/reneelin/opt/anaconda3/python.app/Contents/MacOS/python /Use
Bitcoin Trade Volume: 19057.7

Bitcoin mined: 223750000000

Bitcoin market price: 6835.54


Process finished with exit code 0
```

The following screenshot shows the block fee, block size, and block transactions output:



```
/Users/reneelin/opt/anaconda3/python.app/Contents/MacOS/python /Users/reneelin/PycharmProjects/TestConda/Blockexpl
Block Fee: 7693472

Block Size: 1495042

Block Transactions: [<blockchain.blockexplorer.Transaction object at 0x10e2bcf28>, <blockchain.blockexplorer.Trans

Latest Block Fee: 29882685

Latest Block Size: 1297118

Latest Block Transactions: [<blockchain.blockexplorer.Transaction object at 0x10e2bceb8>, <blockchain.blockexplore


Process finished with exit code 0
```

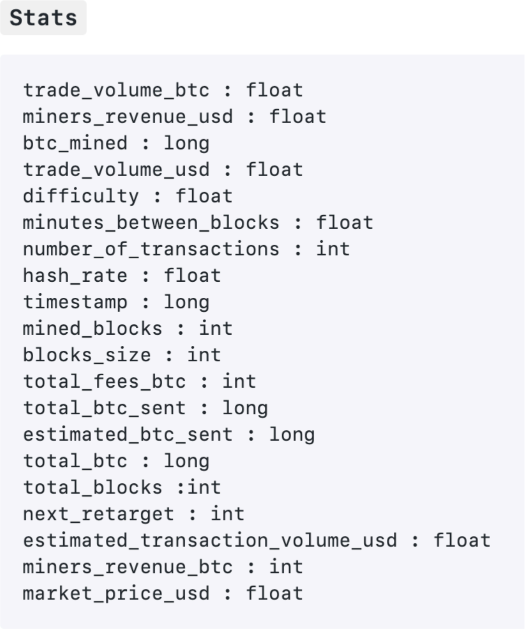
4 Conclusion

4.1 Functions Used

get_ticker Call the 'ticker' method and return a dictionary of Currency objects. Keys are currency symbols (str) and values are Currency objects.

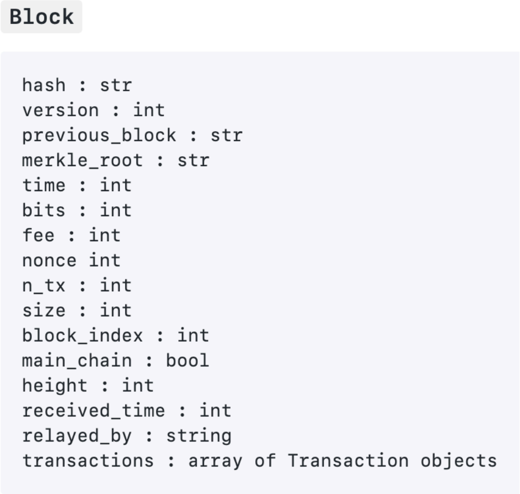
to_btc Call the 'tobtc' method and convert x value in the provided currency to BTC. Returns a float.

get Get network statistics. Returns a Stats object.



```
Stats

trade_volume_btc : float
miners_revenue_usd : float
btc_mined : long
trade_volume_usd : float
difficulty : float
minutes_between_blocks : float
number_of_transactions : int
hash_rate : float
timestamp : long
mined_blocks : int
blocks_size : int
total_fees_btc : int
total_btc_sent : long
estimated_btc_sent : long
total_btc : long
total_blocks :int
next_retarget : int
estimated_transaction_volume_usd : float
miners_revenue_btc : int
market_price_usd : float
```

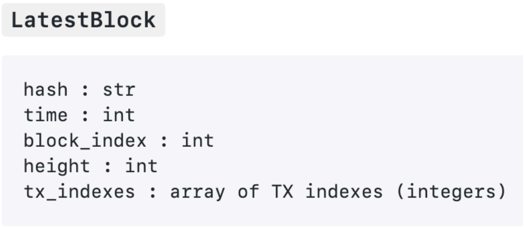


```
Block

hash : str
version : int
previous_block : str
merkle_root : str
time : int
bits : int
fee : int
nonce int
n_tx : int
size : int
block_index : int
main_chain : bool
height : int
received_time : int
relayed_by : string
transactions : array of Transaction objects
```

get_block Get a single block based on a block hash. Returns a Block object.

get_latest_block Get the latest block on the main chain. Returns a LatestBlock object.



`LatestBlock`

```
hash : str
time : int
block_index : int
height : int
tx_indexes : array of TX indexes (integers)
```

4.2 Practise Conclusion

In this practise, I have learnt to install blockchain library and tried Pycharm this time. Meanwhile, I get familiar with blockchain library.

Using the functions in the library, I get the list of currencies and the equivalent bitcoin rate for those currencies and the outputs for 100 euros and 100 CNY in bitcoin.

Then I knew how the website 'blockchain.info' works in the section 'blockchain explorer' and knew the data structure of those organization.

In this practise, I get familiar with blockchain library. By the way, set up the new environment for Pycharm. Successfully complete the practical section this week.

5 References

PDF document: Practical 2 Guide File.
Website: Pypi.org/project/blockchain.
PDF document: How to Write a Practical/Laboratory Report—Learning Guide.