BITCOIN

Bitcoin is a decentralized digital currency that you can buy, sell and exchange directly, without an intermediary like a bank. Bitcoin's creator, Satoshi Nakamoto, originally described the need for "an electronic payment system based on cryptographic proof instead of trust."

Each and every Bitcoin transaction that's ever been made exists on a public ledger accessible to everyone, making transactions hard to reverse and difficult to fake. That's by design: Core to their decentralized nature, Bitcoins aren't backed by the government or any issuing institution, and there's nothing to guarantee their value besides the proof baked in the heart of the system.

Bitcoin is built on a distributed digital record called a blockchain. As the name implies, blockchain is a linked body of data, made up of units called blocks that contain information about each and every transaction, including date and time, total value, buyer and seller, and a unique identifying code for each exchange. Entries are strung together in chronological order, creating a digital chain of blocks.

"Once a block is added to the blockchain, it becomes accessible to anyone who wishes to view it, acting as a public ledger of cryptocurrency transactions," says Stacey Harris, consultant for Pelicoin, a network of cryptocurrency ATMs.

Blockchain is decentralized, which means it's not controlled by any one organization. "It's like a Google Doc that anyone can work on," says Buchi Okoro, CEO and co-founder of African cryptocurrency exchange

Quidax. “Nobody owns it, but anyone who has a link can contribute to it. And as different people update it, your copy also gets updated.”

While the idea that anyone can edit the blockchain might sound risky, it’s actually what makes Bitcoin trustworthy and secure. In order for a transaction block to be added to the Bitcoin blockchain, it must be verified by the majority of all Bitcoin holders, and the unique codes used to recognize users’ wallets and transactions must conform to the right encryption pattern.

These codes are long, random numbers, making them incredibly difficult to fraudulently produce. In fact, a fraudster guessing the key code to your Bitcoin wallet has roughly the same odds as someone winning a Powerball lottery nine times in a row, according to Bryan Lotti of Crypto Aquarium. This level of statistical randomness blockchain verification codes, which are needed for every transaction, greatly reduces the risk anyone can make fraudulent Bitcoin transactions.

BLOCK (BITCOIN BLOCK)

Blocks are files where data pertaining to the Bitcoin network are permanently recorded. A block records some or all of the most recent Bitcoin transactions that have not yet entered any prior blocks. Thus, a block is like a page of a ledger or record book. Each time a block is ‘completed’, it gives way to the next block in the blockchain. A block is thus a permanent store of records which, once written, cannot be altered or removed.

BLOCKCHAIN

A blockchain is a distributed database that is shared among the nodes of a computer network. As a database, a blockchain stores information electronically in digital format. Blockchains are best known for their crucial role in cryptocurrency systems, such as Bitcoin, for maintaining a secure and decentralized record of transactions. The innovation with a blockchain is that it guarantees the fidelity and security of a record of data and generates trust without the need for a trusted third party.

One key difference between a typical database and a blockchain is the way the data is structured. A blockchain collects information together in groups, known as "blocks" that hold sets of information. Blocks have certain storage capacities and, when filled, are closed and linked to the previously filled block, forming a chain of data known as the “blockchain.” All new information that follows that freshly added block is compiled into a newly formed block that will then also be added to the chain once filled.

A database usually structures its data into tables whereas a blockchain, like its name implies, structures its data into chunks (blocks) that are strung together. This data structure inherently makes an irreversible timeline of data when implemented in a decentralized nature. When a block is filled it is set in stone and becomes a part of this timeline. Each block in the chain is given an exact timestamp when it is added to the chain.

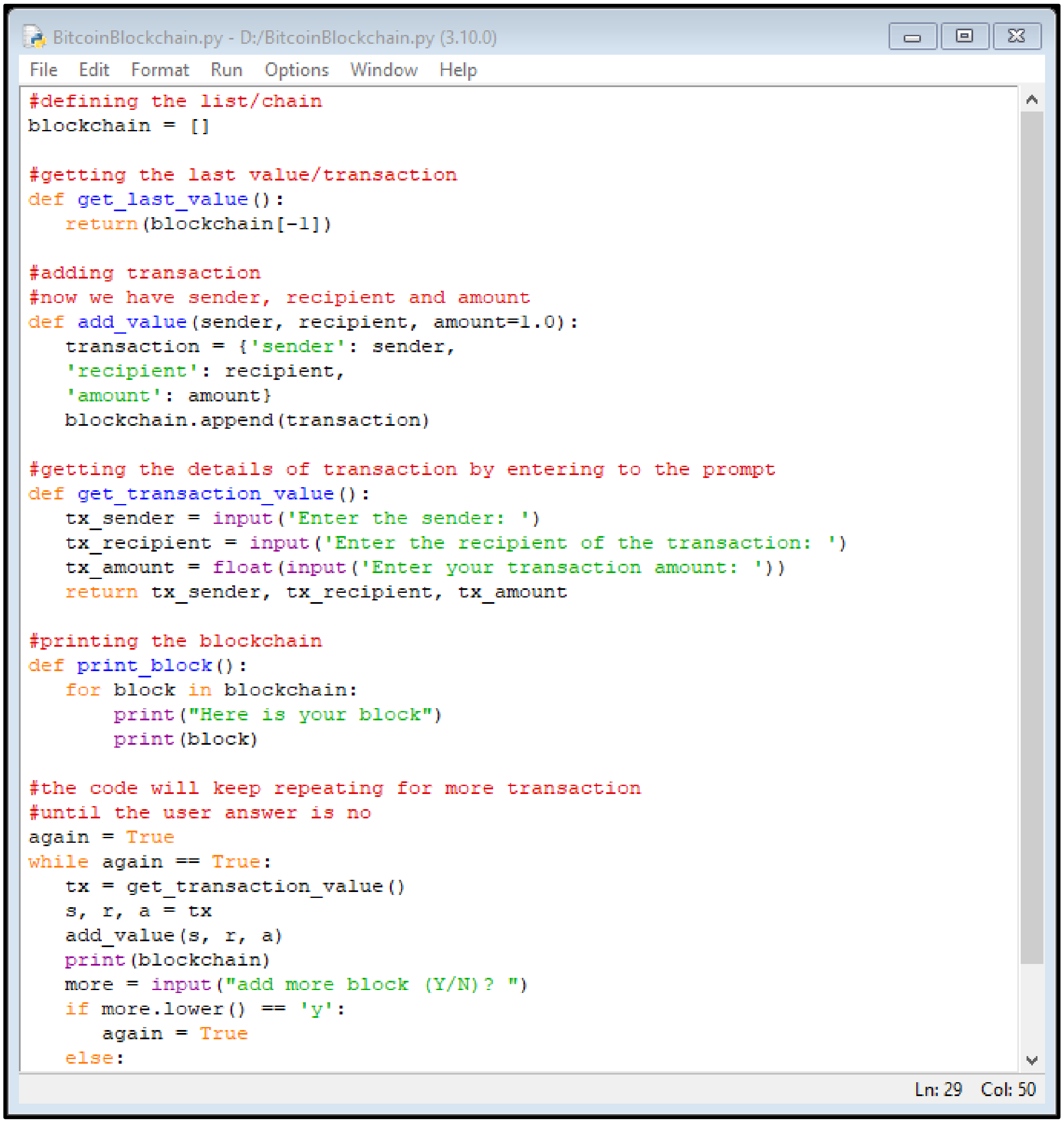
Program:

Implement the creation of Bitcoin Block/Blockchain (Genesis Block)

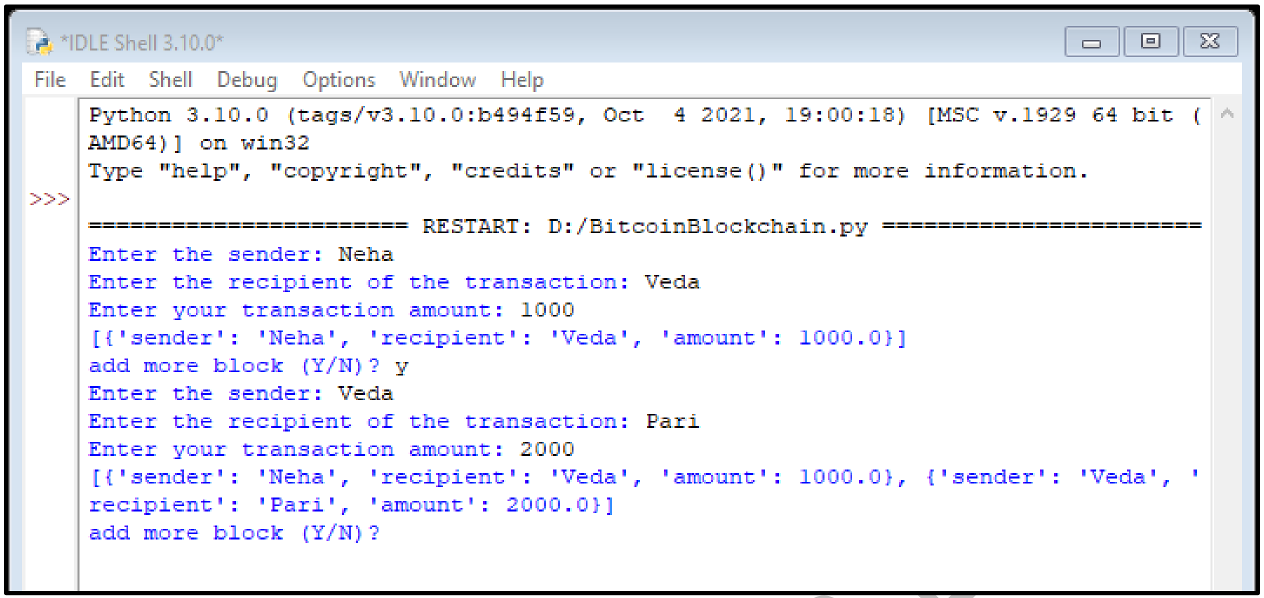
```
#defining the list/chain
blockchain = []

#getting the last value/transaction
def get_last_value():
   return(blockchain[-1])

#adding transaction
#now we have sender, recipient and amount
def add_value(sender, recipient, amount=1.0):
   transaction = {'sender': sender,
   'recipient': recipient,
   'amount': amount}
   blockchain.append(transaction)

#getting the details of transaction by entering to the prompt
def get_transaction_value():
   tx_sender = input('Enter the sender: ')
   tx_recipient = input('Enter the recipient of the transaction: ')
   tx_amount = float(input('Enter your transaction amount: '))
   return tx_sender, tx_recipient, tx_amount
```

```
#printing the blockchain
def print_block():
   for block in blockchain:
       print("Here is your block")
       print(block)

#the code will keep repeating for more transaction
#until the user answer is no
again = True
while again == True:
   tx = get_transaction_value()
   s, r, a = tx
   add_value(s, r, a)
   print(blockchain)
   more = input("add more block (Y/N)? ")
   if more.lower() == 'y':
      again = True
   else:
      again = False
```

Output:


BitcoinBlockchain.py - D:/BitcoinBlockchain.py (3.10.0)
File Edit Format Run Options Window Help
Ln: 29 Col: 50

```python
#defining the list/chain
blockchain = []

#getting the last value/transaction
def get_last_value():
   return(blockchain[-1])

#adding transaction
#now we have sender, recipient and amount
def add_value(sender, recipient, amount=1.0):
   transaction = {'sender': sender,
   'recipient': recipient,
   'amount': amount}
   blockchain.append(transaction)

#getting the details of transaction by entering to the prompt
def get_transaction_value():
   tx_sender = input('Enter the sender: ')
   tx_recipient = input('Enter the recipient of the transaction: ')
   tx_amount = float(input('Enter your transaction amount: '))
   return tx_sender, tx_recipient, tx_amount

#printing the blockchain
def print_block():
   for block in blockchain:
       print("Here is your block")
       print(block)

#the code will keep repeating for more transaction
#until the user answer is no
again = True
while again == True:
   tx = get_transaction_value()
   s, r, a = tx
   add_value(s, r, a)
   print(blockchain)
   more = input("add more block (Y/N)? ")
   if more.lower() == 'y':
      again = True
   else:
```



```
Python 3.10.0 (tags/v3.10.0:b494f59, Oct  4 2021, 19:00:18) [MSC v.1929 64 bit (
AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
======================= RESTART: D:/BitcoinBlockchain.py =======================
Enter the sender: Neha
Enter the recipient of the transaction: Veda
Enter your transaction amount: 1000
[{'sender': 'Neha', 'recipient': 'Veda', 'amount': 1000.0}]
add more block (Y/N)? y
Enter the sender: Veda
Enter the recipient of the transaction: Pari
Enter your transaction amount: 2000
[{'sender': 'Neha', 'recipient': 'Veda', 'amount': 1000.0}, {'sender': 'Veda', '
recipient': 'Pari', 'amount': 2000.0}]
add more block (Y/N)?
```

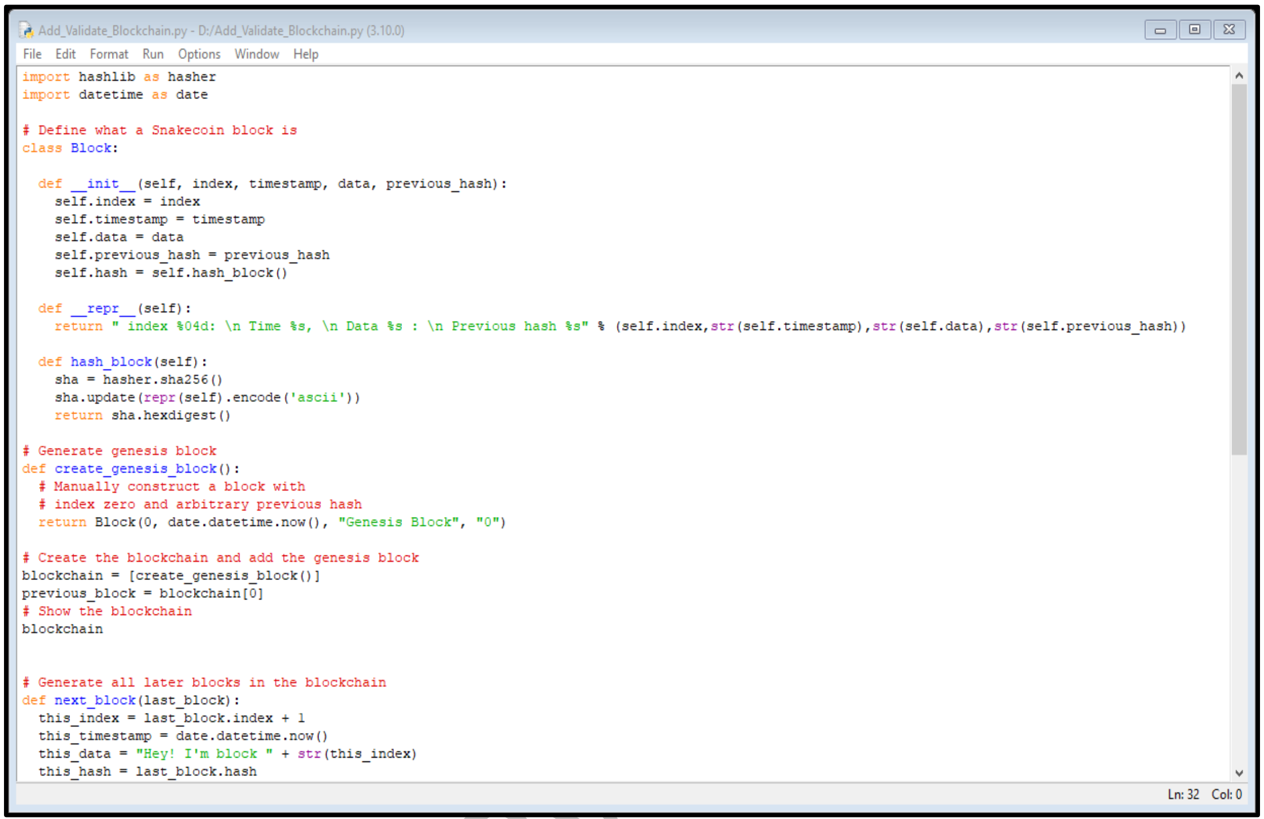
Program:

Implement the creation of a Blockchain (Adding the blocks to the chain and validating)

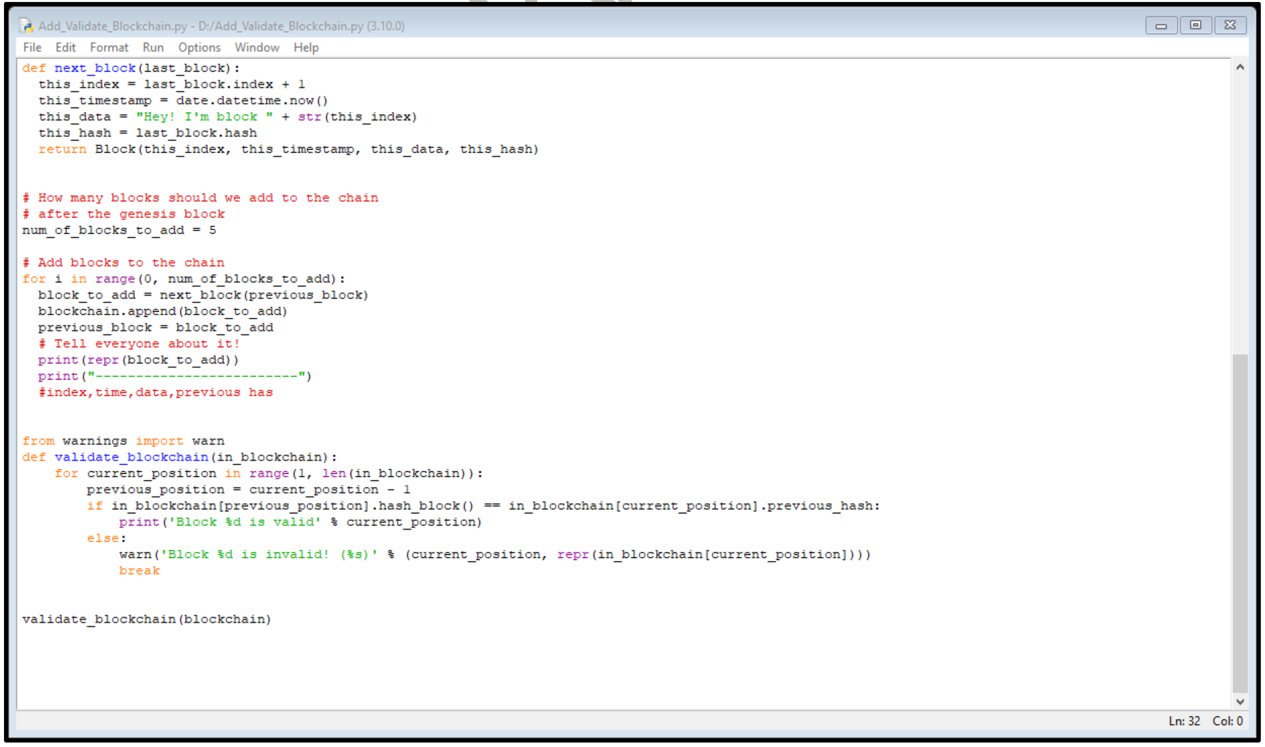
```
import hashlib as hasher

import datetime as date


# Define what a Snakecoin block is

class Block:


  def __init__(self, index, timestamp, data, previous_hash):

    self.index = index

    self.timestamp = timestamp

    self.data = data
    self.previous_hash = previous_hash
    self.hash = self.hash_block()

  def __repr__(self):
    return " index %04d: \n Time %s, \n Data %s : \n Previous hash %s" %
(self.index,str(self.timestamp),str(self.data),str(self.previous_hash))

  def hash_block(self):
    sha = hasher.sha256()
    sha.update(repr(self).encode('ascii'))
    return sha.hexdigest()

# Generate genesis block
def create_genesis_block():
  # Manually construct a block with
  # index zero and arbitrary previous hash
  return Block(0, date.datetime.now(), "Genesis Block", "0")

# Create the blockchain and add the genesis block
blockchain = [create_genesis_block()]
previous_block = blockchain[0]
# Show the blockchain
blockchain
```

```
# Generate all later blocks in the blockchain
def next_block(last_block):
  this_index = last_block.index + 1
  this_timestamp = date.datetime.now()
  this_data = "Hey! I'm block " + str(this_index)
  this_hash = last_block.hash
  return Block(this_index, this_timestamp, this_data, this_hash)


# How many blocks should we add to the chain
# after the genesis block
num_of_blocks_to_add = 5

# Add blocks to the chain
for i in range(0, num_of_blocks_to_add):
  block_to_add = next_block(previous_block)
  blockchain.append(block_to_add)
  previous_block = block_to_add
  # Tell everyone about it!
  print(repr(block_to_add))
  print("-------------------------")
  #index,time,data,previous has



from warnings import warn
def validate_blockchain(in_blockchain):
    for current_position in range(1, len(in_blockchain)):
        previous_position = current_position - 1
        if            in_blockchain[previous_position].hash_block()            ==
in_blockchain[current_position].previous_hash:
            print('Block %d is valid' % current_position)
        else:
            warn('Block    %d    is    invalid!    (%s)'    %    (current_position,
repr(in_blockchain[current_position])))
            break


validate_blockchain(blockchain)
```

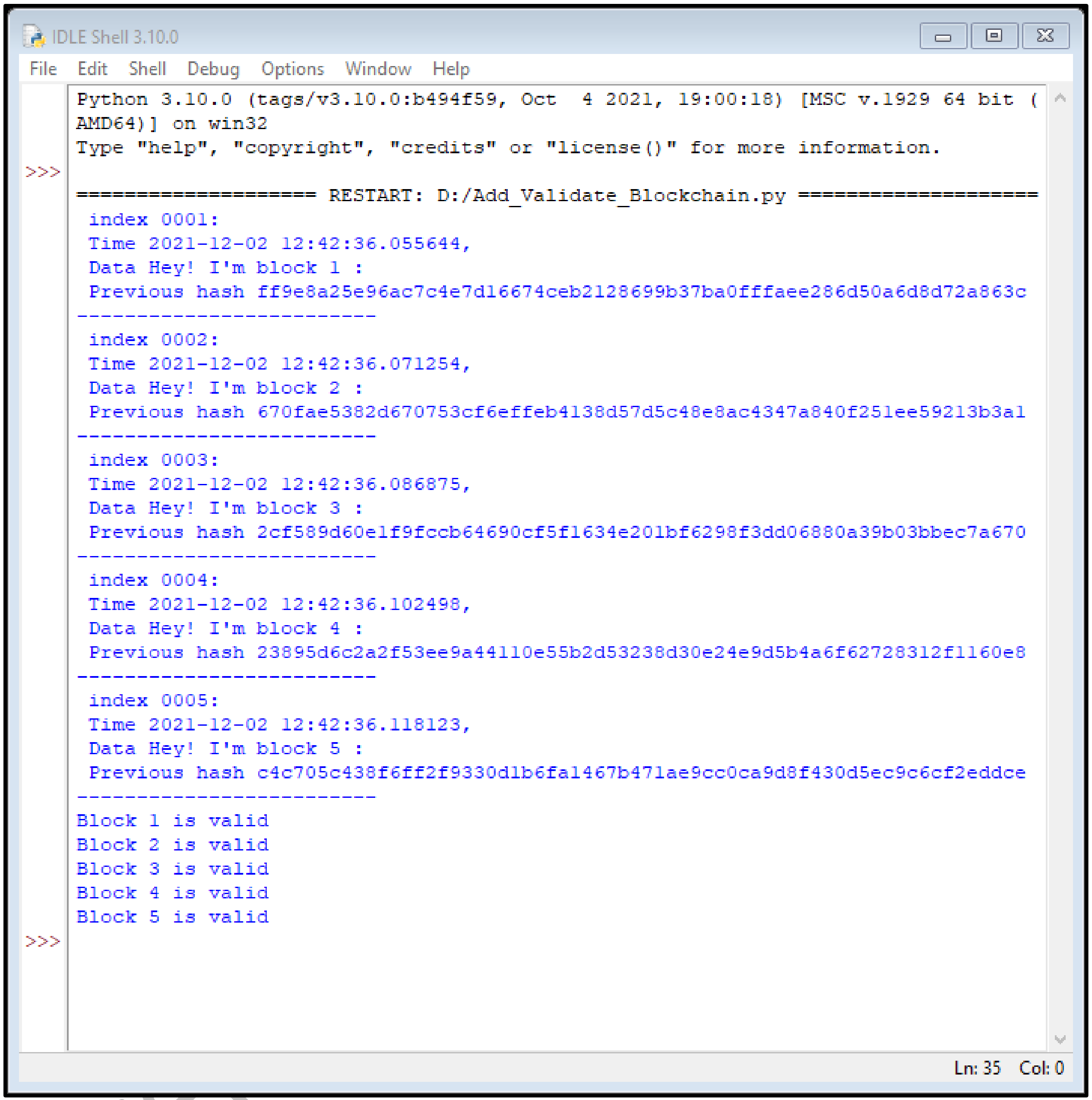
Output:


Add_Validate_Blockchain.py - D:/Add_Validate_Blockchain.py (3.10.0)
File Edit Format Run Options Window Help
Ln: 32 Col: 0

```python
import hashlib as hasher
import datetime as date

# Define what a Snakecoin block is
class Block:

  def __init__(self, index, timestamp, data, previous_hash):
    self.index = index
    self.timestamp = timestamp
    self.data = data
    self.previous_hash = previous_hash
    self.hash = self.hash_block()

  def __repr__(self):
    return " index %04d: \n Time %s, \n Data %s : \n Previous hash %s" % (self.index,str(self.timestamp),str(self.data),str(self.previous_hash))

  def hash_block(self):
    sha = hasher.sha256()
    sha.update(repr(self).encode('ascii'))
    return sha.hexdigest()

# Generate genesis block
def create_genesis_block():
  # Manually construct a block with
  # index zero and arbitrary previous hash
  return Block(0, date.datetime.now(), "Genesis Block", "0")

# Create the blockchain and add the genesis block
blockchain = [create_genesis_block()]
previous_block = blockchain[0]
# Show the blockchain
blockchain


# Generate all later blocks in the blockchain
def next_block(last_block):
  this_index = last_block.index + 1
  this_timestamp = date.datetime.now()
  this_data = "Hey! I'm block " + str(this_index)
  this_hash = last_block.hash
```


Add_Validate_Blockchain.py - D:/Add_Validate_Blockchain.py (3.10.0)
File Edit Format Run Options Window Help
Ln: 32 Col: 0

```python
def next_block(last_block):
  this_index = last_block.index + 1
  this_timestamp = date.datetime.now()
  this_data = "Hey! I'm block " + str(this_index)
  this_hash = last_block.hash
  return Block(this_index, this_timestamp, this_data, this_hash)


# How many blocks should we add to the chain
# after the genesis block
num_of_blocks_to_add = 5

# Add blocks to the chain
for i in range(0, num_of_blocks_to_add):
  block_to_add = next_block(previous_block)
  blockchain.append(block_to_add)
  previous_block = block_to_add
  # Tell everyone about it!
  print(repr(block_to_add))
  print("-------------------------")
  #index,time,data,previous has


from warnings import warn
def validate_blockchain(in_blockchain):
    for current_position in range(1, len(in_blockchain)):
        previous_position = current_position - 1
        if in_blockchain[previous_position].hash_block() == in_blockchain[current_position].previous_hash:
            print('Block %d is valid' % current_position)
        else:
            warn('Block %d is invalid! (%s)' % (current_position, repr(in_blockchain[current_position])))
            break


validate_blockchain(blockchain)
```



```
Python 3.10.0 (tags/v3.10.0:b494f59, Oct  4 2021, 19:00:18) [MSC v.1929 64 bit (
AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>> 
==================== RESTART: D:/Add_Validate_Blockchain.py ====================
 index 0001:
 Time 2021-12-02 12:42:36.055644,
 Data Hey! I'm block 1 :
 Previous hash ff9e8a25e96ac7c4e7d16674ceb2128699b37ba0fffaee286d50a6d8d72a863c
-------------------------
 index 0002:
 Time 2021-12-02 12:42:36.071254,
 Data Hey! I'm block 2 :
 Previous hash 670fae5382d670753cf6effeb4138d57d5c48e8ac4347a840f251ee59213b3a1
-------------------------
 index 0003:
 Time 2021-12-02 12:42:36.086875,
 Data Hey! I'm block 3 :
 Previous hash 2cf589d60e1f9fccb64690cf5f1634e201bf6298f3dd06880a39b03bbec7a670
-------------------------
 index 0004:
 Time 2021-12-02 12:42:36.102498,
 Data Hey! I'm block 4 :
 Previous hash 23895d6c2a2f53ee9a44110e55b2d53238d30e24e9d5b4a6f62728312f1160e8
-------------------------
 index 0005:
 Time 2021-12-02 12:42:36.118123,
 Data Hey! I'm block 5 :
 Previous hash c4c705c438f6ff2f9330d1b6fa1467b471ae9cc0ca9d8f430d5ec9c6cf2eddce
-------------------------
Block 1 is valid
Block 2 is valid
Block 3 is valid
Block 4 is valid
Block 5 is valid
>>> 
```

PRIVATE BLOCKCHAIN USING GETH

(Implement the creation of a public/private Blockchain)

Ethereum node is any device that is running the Ethereum protocol (blockchain).

When we connect to the Ethereum protocol we are on the Ethereum blockchain network.

By running an Ethereum node we can connect to other nodes in the network, have direct access to the blockchain, and even do things like mine blocks, send transactions, and deploy smart contracts.

Step 1

Download and Install NodeJs

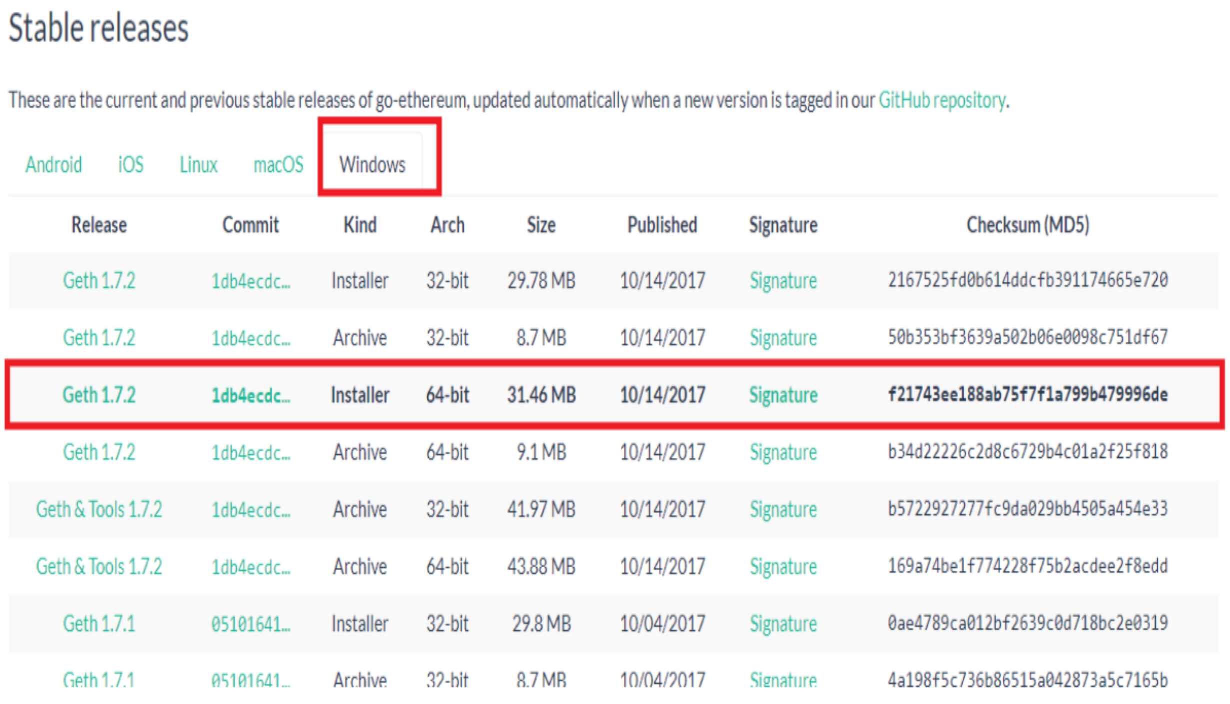
https://nodejs.org/en/download/

Step 2

- **Installation:**

1. Visit the Go Ethereum website and install Geth
   Visit here: https://geth.ethereum.org/downloads/

2. Download the latest release of Geth for Windows, make sure you download the 64-bit version



Stable releases

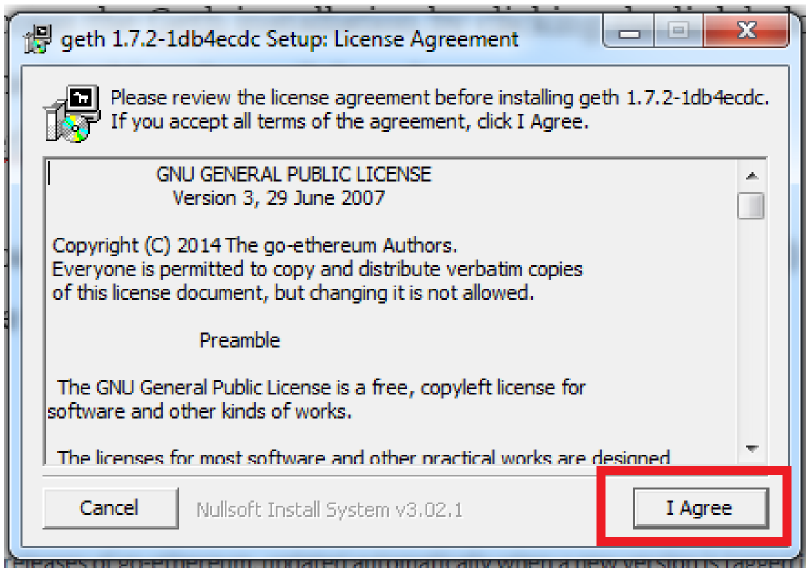
These are the current and previous stable releases of go-ethereum, updated automatically when a new version is tagged in our GitHub repository.

Android iOS Linux macOS Windows

Release	Commit	Kind	Arch	Size	Published	Signature	Checksum (MD5)
Geth 1.7.2	1db4ecdc...	Installer	32-bit	29.78 MB	10/14/2017	Signature	2167525fd0b614ddcfb391174665e720
Geth 1.7.2	1db4ecdc...	Archive	32-bit	8.7 MB	10/14/2017	Signature	50b353bf3639a502b06e0098c751df67
Geth 1.7.2	**1db4ecdc...**	**Installer**	**64-bit**	**31.46 MB**	**10/14/2017**	**Signature**	**f21743ee188ab75f7f1a799b479996de**
Geth 1.7.2	1db4ecdc...	Archive	64-bit	9.1 MB	10/14/2017	Signature	b34d22226c2d8c6729b4c01a2f25f818
Geth & Tools 1.7.2	1db4ecdc...	Archive	32-bit	41.97 MB	10/14/2017	Signature	b5722927277fc9da029bb4505a454e33
Geth & Tools 1.7.2	1db4ecdc...	Archive	64-bit	43.88 MB	10/14/2017	Signature	169a74be1f774228f75b2acdee2f8edd
Geth 1.7.1	05101641...	Installer	32-bit	29.8 MB	10/04/2017	Signature	0ae4789ca012bf2639c0d718bc2e0319
Geth 1.7.1	05101641...	Archive	32-bit	8.7 MB	10/04/2017	Signature	4a198f5c736b86515a042873a5c7165b

Geth Ethereum Downloadable Releases

3. Once your download is complete, open the installer and click "I Agree"

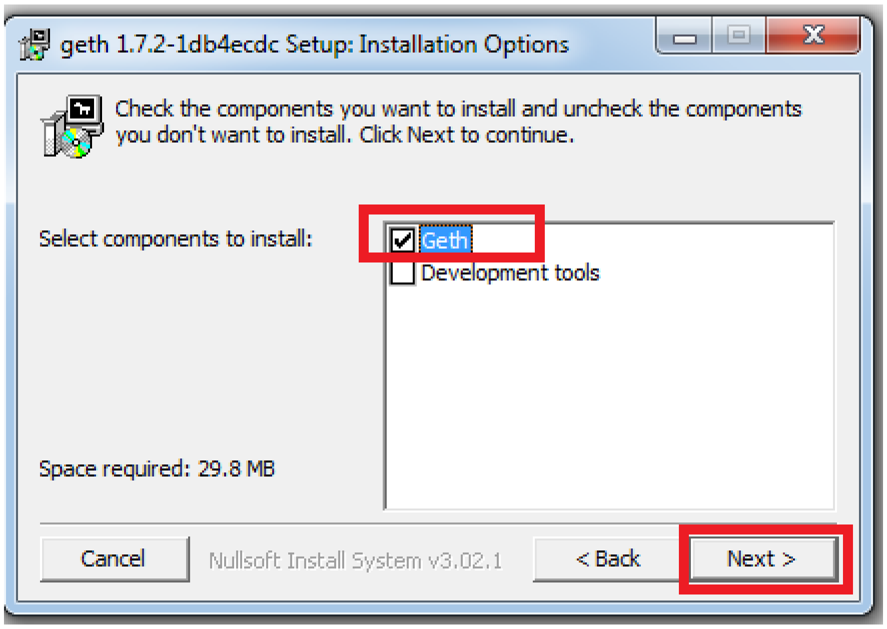


geth 1.7.2-1db4ecdc Setup: License Agreement

Please review the license agreement before installing geth 1.7.2-1db4ecdc. If you accept all terms of the agreement, click I Agree.

```
                GNU GENERAL PUBLIC LICENSE
                  Version 3, 29 June 2007

 Copyright (C) 2014 The go-ethereum Authors.
 Everyone is permitted to copy and distribute verbatim copies
 of this license document, but changing it is not allowed.

                    Preamble

  The GNU General Public License is a free, copyleft license for
software and other kinds of works.

  The licenses for most software and other practical works are designed
```

4. Make sure the Geth box is checked and click “Next”


geth 1.7.2-1db4ecdc Setup: Installation Options

Check the components you want to install and uncheck the components you don't want to install. Click Next to continue.

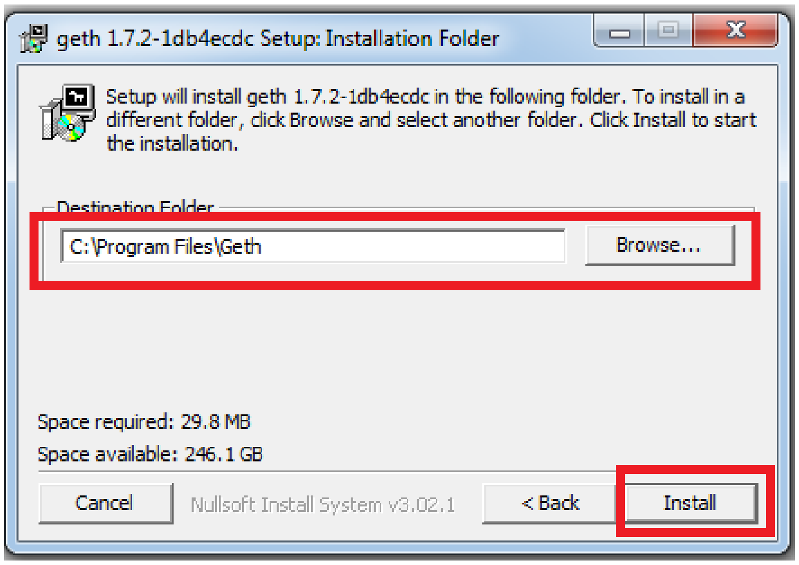
Select components to install:

[✓] Geth

[ ] Development tools

Space required: 29.8 MB

5. You’ll be prompted to select a destination folder for your download. By default, Geth will install under C:\Program Files\Geth


geth 1.7.2-1db4ecdc Setup: Installation Folder

Setup will install geth 1.7.2-1db4ecdc in the following folder. To install in a different folder, click Browse and select another folder. Click Install to start the installation.

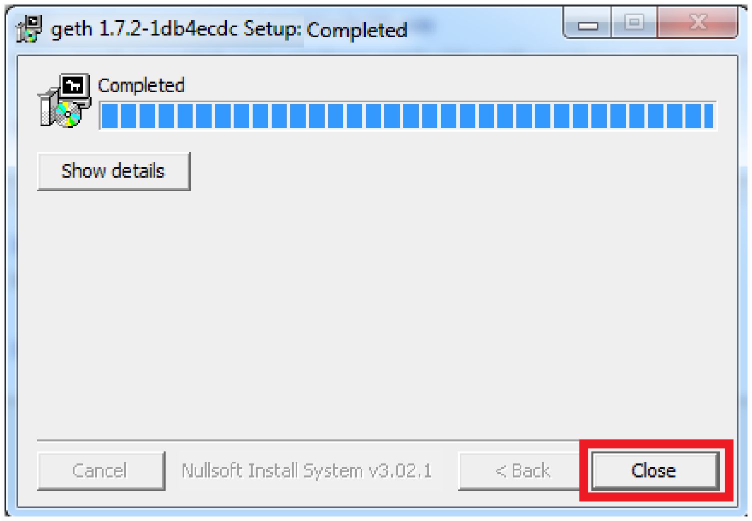
Destination Folder

C:\Program Files\Geth

Space required: 29.8 MB

Space available: 246.1 GB

6. Close installation once complete


geth 1.7.2-1db4ecdc Setup: Completed
Completed
Show details
Cancel
Nullsoft Install System v3.02.1
< Back
Close

Step 3

Establishing Our Own Private Ethereum Network

1. Create a new folder on your desktop called "Private-chain"
2. Open command prompt in this folder and create a data directory folder for our chaindata by typing "mkdir chaindata
3. Next, we need to create and save our genesis.json block in our Private-chain folder, as the genesis block will be used to initialize our private network and store data in the data directory folder "chaindata" .Initially "chaindata" folder is NULL. Once all steps done, observe the structure of "chaindata" folder.

   Open up notepad, copy & paste the code below into a new file called "genesis.json" and save this file in our Private-chain folder.

```
{
 "config": {
 "chainId": 4777,
 "homesteadBlock": 0,
 "eip150Block": 0,
 "eip155Block": 0,
 "eip158Block": 0
 },
 "alloc" : {},
 "difficulty" : "0x400",
 "extraData" : "",
 "gasLimit" : "0x7A1200",
 "parentHash" :
"0x0000000000000000000000000000000000000000000000000000000
000000000",
 "timestamp" : "0x00"
}
```

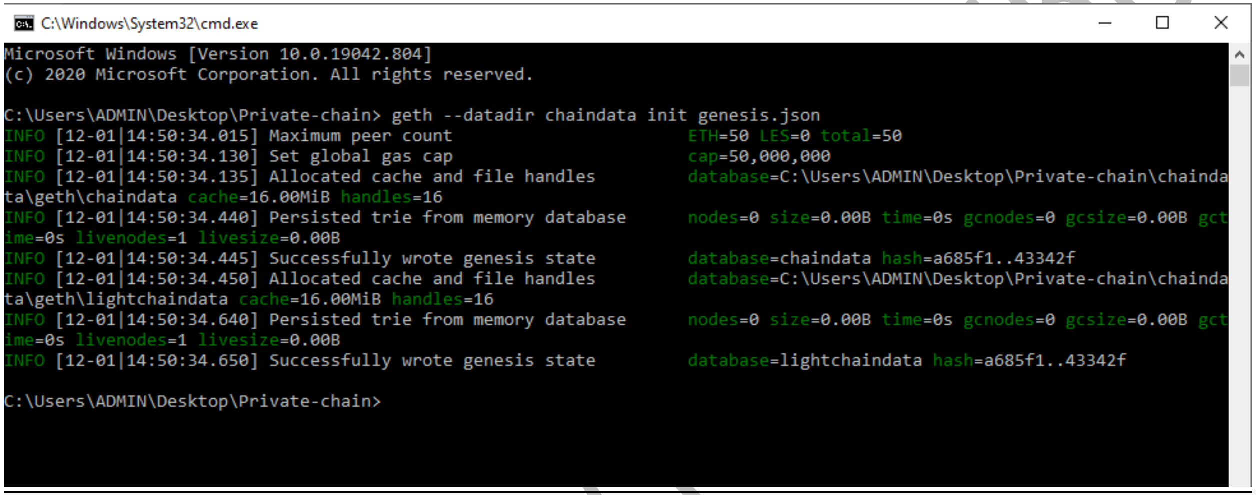
Step 4

Start the Ethereum peer node (Start the Blockchain)

Run the command:

geth --datadir chaindata init genesis.json

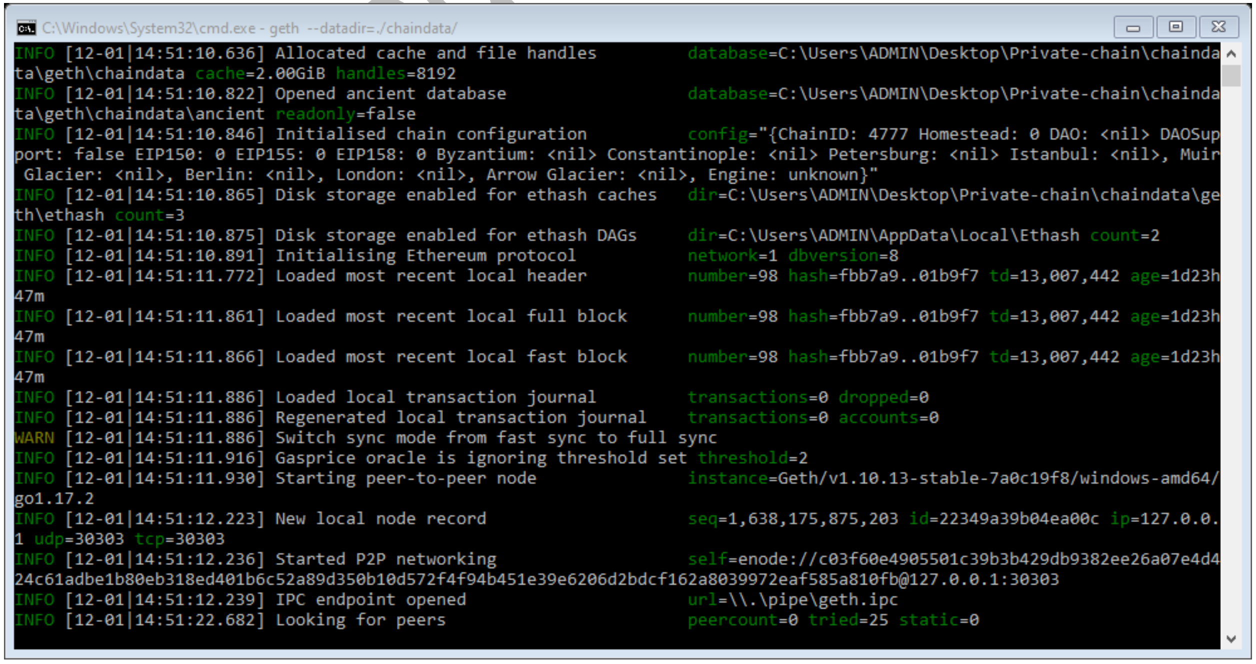
<u>It will Intialize geth into chaindata</u>



```
C:\Windows\System32\cmd.exe
Microsoft Windows [Version 10.0.19042.804]
(c) 2020 Microsoft Corporation. All rights reserved.

C:\Users\ADMIN\Desktop\Private-chain> geth --datadir chaindata init genesis.json
INFO [12-01|14:50:34.015] Maximum peer count                       ETH=50 LES=0 total=50
INFO [12-01|14:50:34.130] Set global gas cap                       cap=50,000,000
INFO [12-01|14:50:34.135] Allocated cache and file handles         database=C:\Users\ADMIN\Desktop\Private-chain\chainda
ta\geth\chaindata cache=16.00MiB handles=16
INFO [12-01|14:50:34.440] Persisted trie from memory database      nodes=0 size=0.00B time=0s gcnodes=0 gcsize=0.00B gct
ime=0s livenodes=1 livesize=0.00B
INFO [12-01|14:50:34.445] Successfully wrote genesis state         database=chaindata hash=a685f1..43342f
INFO [12-01|14:50:34.450] Allocated cache and file handles         database=C:\Users\ADMIN\Desktop\Private-chain\chainda
ta\geth\lightchaindata cache=16.00MiB handles=16
INFO [12-01|14:50:34.640] Persisted trie from memory database      nodes=0 size=0.00B time=0s gcnodes=0 gcsize=0.00B gct
ime=0s livenodes=1 livesize=0.00B
INFO [12-01|14:50:34.650] Successfully wrote genesis state         database=lightchaindata hash=a685f1..43342f

C:\Users\ADMIN\Desktop\Private-chain>
```

Now we can start Geth and connect to our own private chain

geth --datadir=./chaindata/



```
C:\Windows\System32\cmd.exe - geth  --datadir=./chaindata/
INFO [12-01|14:51:10.636] Allocated cache and file handles         database=C:\Users\ADMIN\Desktop\Private-chain\chainda
ta\geth\chaindata cache=2.00GiB handles=8192
INFO [12-01|14:51:10.822] Opened ancient database                  database=C:\Users\ADMIN\Desktop\Private-chain\chainda
ta\geth\chaindata\ancient readonly=false
INFO [12-01|14:51:10.846] Initialised chain configuration          config="{ChainID: 4777 Homestead: 0 DAO: <nil> DAOSup
port: false EIP150: 0 EIP155: 0 EIP158: 0 Byzantium: <nil> Constantinople: <nil> Petersburg: <nil> Istanbul: <nil>, Muir
 Glacier: <nil>, Berlin: <nil>, London: <nil>, Arrow Glacier: <nil>, Engine: unknown}"
INFO [12-01|14:51:10.865] Disk storage enabled for ethash caches   dir=C:\Users\ADMIN\Desktop\Private-chain\chaindata\ge
th\ethash count=3
INFO [12-01|14:51:10.875] Disk storage enabled for ethash DAGs     dir=C:\Users\ADMIN\AppData\Local\Ethash count=2
INFO [12-01|14:51:10.891] Initialising Ethereum protocol           network=1 dbversion=8
INFO [12-01|14:51:11.772] Loaded most recent local header          number=98 hash=fbb7a9..01b9f7 td=13,007,442 age=1d23h
47m
INFO [12-01|14:51:11.861] Loaded most recent local full block      number=98 hash=fbb7a9..01b9f7 td=13,007,442 age=1d23h
47m
INFO [12-01|14:51:11.866] Loaded most recent local fast block      number=98 hash=fbb7a9..01b9f7 td=13,007,442 age=1d23h
47m
INFO [12-01|14:51:11.886] Loaded local transaction journal         transactions=0 dropped=0
INFO [12-01|14:51:11.886] Regenerated local transaction journal    transactions=0 accounts=0
WARN [12-01|14:51:11.886] Switch sync mode from fast sync to full sync
INFO [12-01|14:51:11.916] Gasprice oracle is ignoring threshold set threshold=2
INFO [12-01|14:51:11.930] Starting peer-to-peer node               instance=Geth/v1.10.13-stable-7a0c19f8/windows-amd64/
go1.17.2
INFO [12-01|14:51:12.223] New local node record                    seq=1,638,175,875,203 id=22349a39b04ea00c ip=127.0.0.
1 udp=30303 tcp=30303
INFO [12-01|14:51:12.236] Started P2P networking                   self=enode://c03f60e4905501c39b3b429db9382ee26a07e4d4
24c61adbe1b80eb318ed401b6c52a89d350b10d572f4f94b451e39e6206d2bdcf162a8039972eaf585a810fb@127.0.0.1:30303
INFO [12-01|14:51:12.239] IPC endpoint opened                      url=\\.\pipe\geth.ipc
INFO [12-01|14:51:22.682] Looking for peers                        peercount=0 tried=25 static=0
```

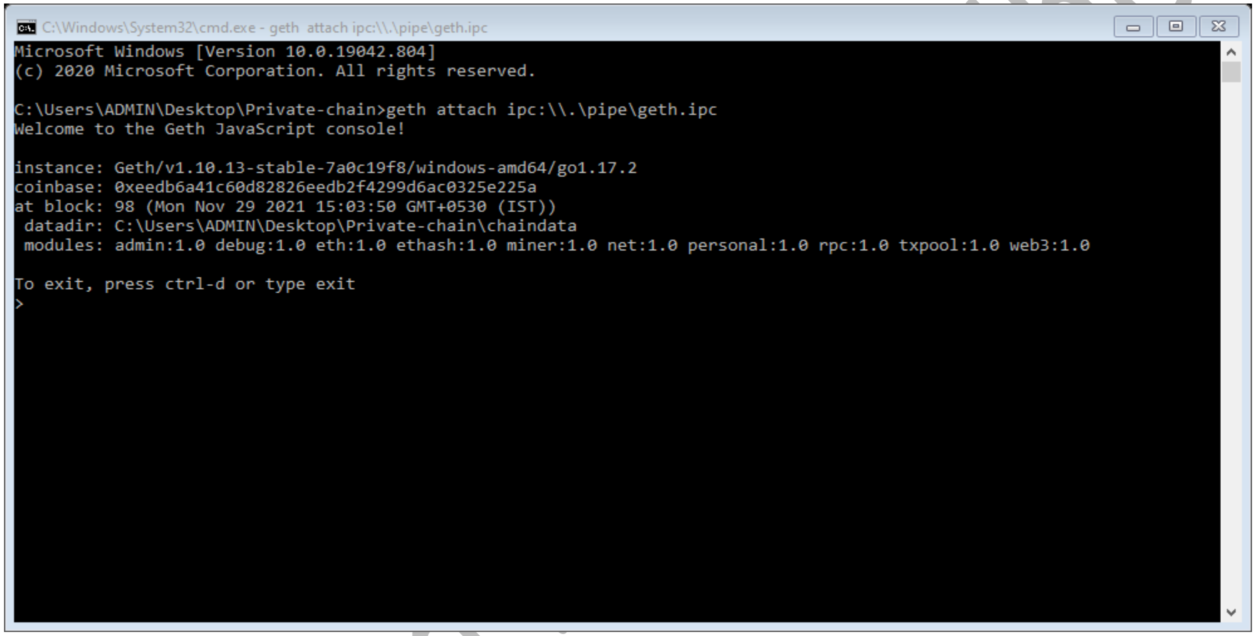
Step 5

Minimize the terminal and open a new terminal

IPC to interact with Geth:

geth attach ipc:\\.\pipe\geth.ipc

(Run this command on new terminal)



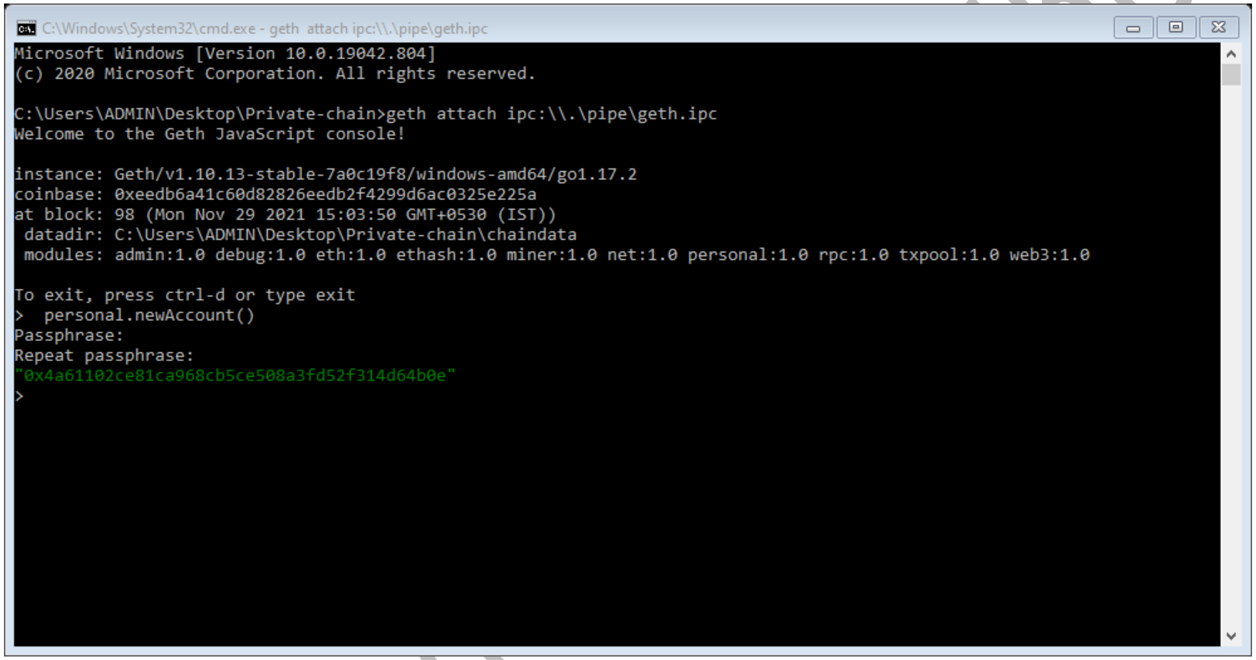
C:\Windows\System32\cmd.exe - geth attach ipc:\\.\pipe\geth.ipc

```
Microsoft Windows [Version 10.0.19042.804]
(c) 2020 Microsoft Corporation. All rights reserved.

C:\Users\ADMIN\Desktop\Private-chain>geth attach ipc:\\.\pipe\geth.ipc
Welcome to the Geth JavaScript console!

instance: Geth/v1.10.13-stable-7a0c19f8/windows-amd64/go1.17.2
coinbase: 0xeedb6a41c60d82826eedb2f4299d6ac0325e225a
at block: 98 (Mon Nov 29 2021 15:03:50 GMT+0530 (IST))
 datadir: C:\Users\ADMIN\Desktop\Private-chain\chaindata
 modules: admin:1.0 debug:1.0 eth:1.0 ethash:1.0 miner:1.0 net:1.0 personal:1.0 rpc:1.0 txpool:1.0 web3:1.0

To exit, press ctrl-d or type exit
>
```

Create Account

personal.newAccount()

(Run this command on terminal)

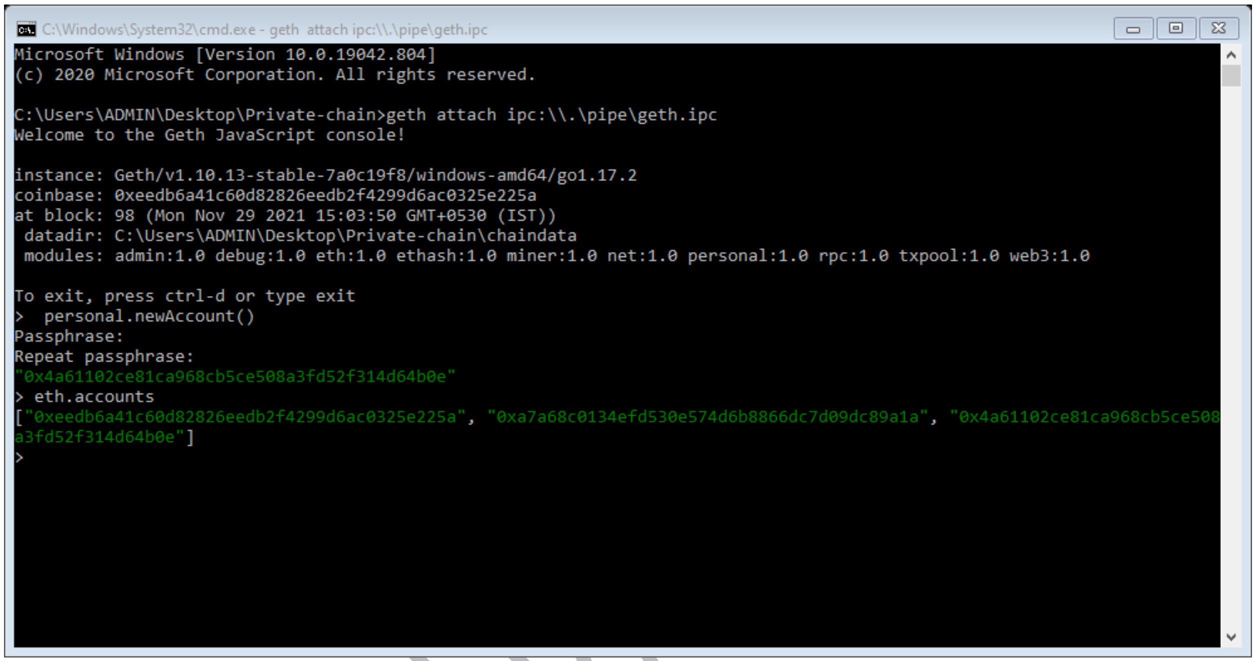


```
C:\Windows\System32\cmd.exe - geth  attach ipc:\\.\pipe\geth.ipc
Microsoft Windows [Version 10.0.19042.804]
(c) 2020 Microsoft Corporation. All rights reserved.

C:\Users\ADMIN\Desktop\Private-chain>geth attach ipc:\\.\pipe\geth.ipc
Welcome to the Geth JavaScript console!

instance: Geth/v1.10.13-stable-7a0c19f8/windows-amd64/go1.17.2
coinbase: 0xeedb6a41c60d82826eedb2f4299d6ac0325e225a
at block: 98 (Mon Nov 29 2021 15:03:50 GMT+0530 (IST))
 datadir: C:\Users\ADMIN\Desktop\Private-chain\chaindata
 modules: admin:1.0 debug:1.0 eth:1.0 ethash:1.0 miner:1.0 net:1.0 personal:1.0 rpc:1.0 txpool:1.0 web3:1.0

To exit, press ctrl-d or type exit
>  personal.newAccount()
Passphrase:
Repeat passphrase:
"0x4a61102ce81ca968cb5ce508a3fd52f314d64b0e"
>
```

(you can create multiple accounts. It will ask password. Note the password. This password required while you unlock account) eth.accounts

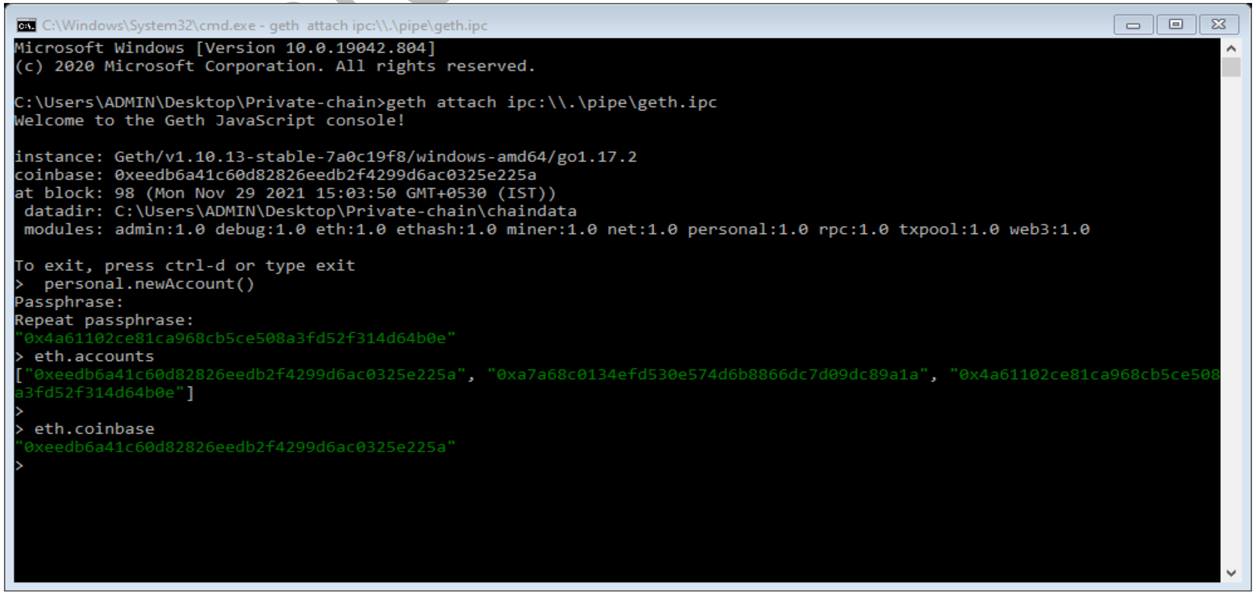
(Run this command on terminal)

It will show number of accounts created



```
C:\Windows\System32\cmd.exe - geth  attach ipc:\\.\pipe\geth.ipc
Microsoft Windows [Version 10.0.19042.804]
(c) 2020 Microsoft Corporation. All rights reserved.

C:\Users\ADMIN\Desktop\Private-chain>geth attach ipc:\\.\pipe\geth.ipc
Welcome to the Geth JavaScript console!

instance: Geth/v1.10.13-stable-7a0c19f8/windows-amd64/go1.17.2
coinbase: 0xeedb6a41c60d82826eedb2f4299d6ac0325e225a
at block: 98 (Mon Nov 29 2021 15:03:50 GMT+0530 (IST))
 datadir: C:\Users\ADMIN\Desktop\Private-chain\chaindata
 modules: admin:1.0 debug:1.0 eth:1.0 ethash:1.0 miner:1.0 net:1.0 personal:1.0 rpc:1.0 txpool:1.0 web3:1.0

To exit, press ctrl-d or type exit
>  personal.newAccount()
Passphrase:
Repeat passphrase:
"0x4a61102ce81ca968cb5ce508a3fd52f314d64b0e"
> eth.accounts
["0xeedb6a41c60d82826eedb2f4299d6ac0325e225a", "0xa7a68c0134efd530e574d6b8866dc7d09dc89a1a", "0x4a61102ce81ca968cb5ce508
a3fd52f314d64b0e"]
>
```

eth.coinbase

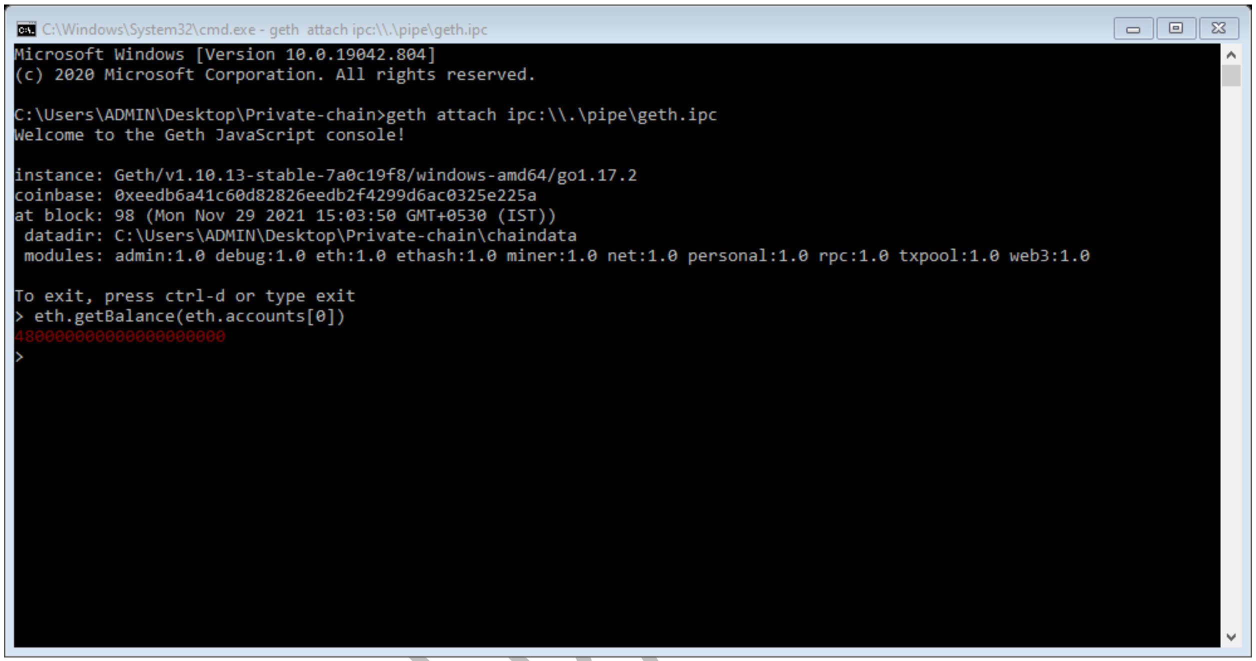


```
C:\Windows\System32\cmd.exe - geth  attach ipc:\\.\pipe\geth.ipc
Microsoft Windows [Version 10.0.19042.804]
(c) 2020 Microsoft Corporation. All rights reserved.

C:\Users\ADMIN\Desktop\Private-chain>geth attach ipc:\\.\pipe\geth.ipc
Welcome to the Geth JavaScript console!

instance: Geth/v1.10.13-stable-7a0c19f8/windows-amd64/go1.17.2
coinbase: 0xeedb6a41c60d82826eedb2f4299d6ac0325e225a
at block: 98 (Mon Nov 29 2021 15:03:50 GMT+0530 (IST))
 datadir: C:\Users\ADMIN\Desktop\Private-chain\chaindata
 modules: admin:1.0 debug:1.0 eth:1.0 ethash:1.0 miner:1.0 net:1.0 personal:1.0 rpc:1.0 txpool:1.0 web3:1.0

To exit, press ctrl-d or type exit
>  personal.newAccount()
Passphrase:
Repeat passphrase:
"0x4a61102ce81ca968cb5ce508a3fd52f314d64b0e"
> eth.accounts
["0xeedb6a41c60d82826eedb2f4299d6ac0325e225a", "0xa7a68c0134efd530e574d6b8866dc7d09dc89a1a", "0x4a61102ce81ca968cb5ce508
a3fd52f314d64b0e"]
>
> eth.coinbase
"0xeedb6a41c60d82826eedb2f4299d6ac0325e225a"
>
```

eth.getBalance(eth.accounts[0])

(Run this command on terminal)

It will show balance of account.

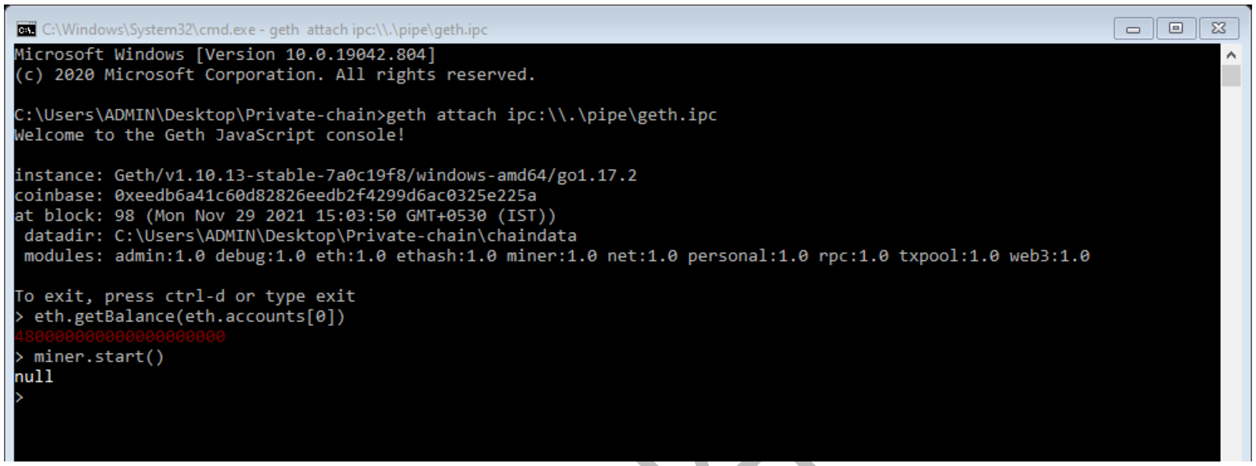


```
C:\Windows\System32\cmd.exe - geth  attach ipc:\\.\pipe\geth.ipc
Microsoft Windows [Version 10.0.19042.804]
(c) 2020 Microsoft Corporation. All rights reserved.

C:\Users\ADMIN\Desktop\Private-chain>geth attach ipc:\\.\pipe\geth.ipc
Welcome to the Geth JavaScript console!

instance: Geth/v1.10.13-stable-7a0c19f8/windows-amd64/go1.17.2
coinbase: 0xeedb6a41c60d82826eedb2f4299d6ac0325e225a
at block: 98 (Mon Nov 29 2021 15:03:50 GMT+0530 (IST))
 datadir: C:\Users\ADMIN\Desktop\Private-chain\chaindata
 modules: admin:1.0 debug:1.0 eth:1.0 ethash:1.0 miner:1.0 net:1.0 personal:1.0 rpc:1.0 txpool:1.0 web3:1.0

To exit, press ctrl-d or type exit
> eth.getBalance(eth.accounts[0])
480000000000000000000
>
```

miner.start()

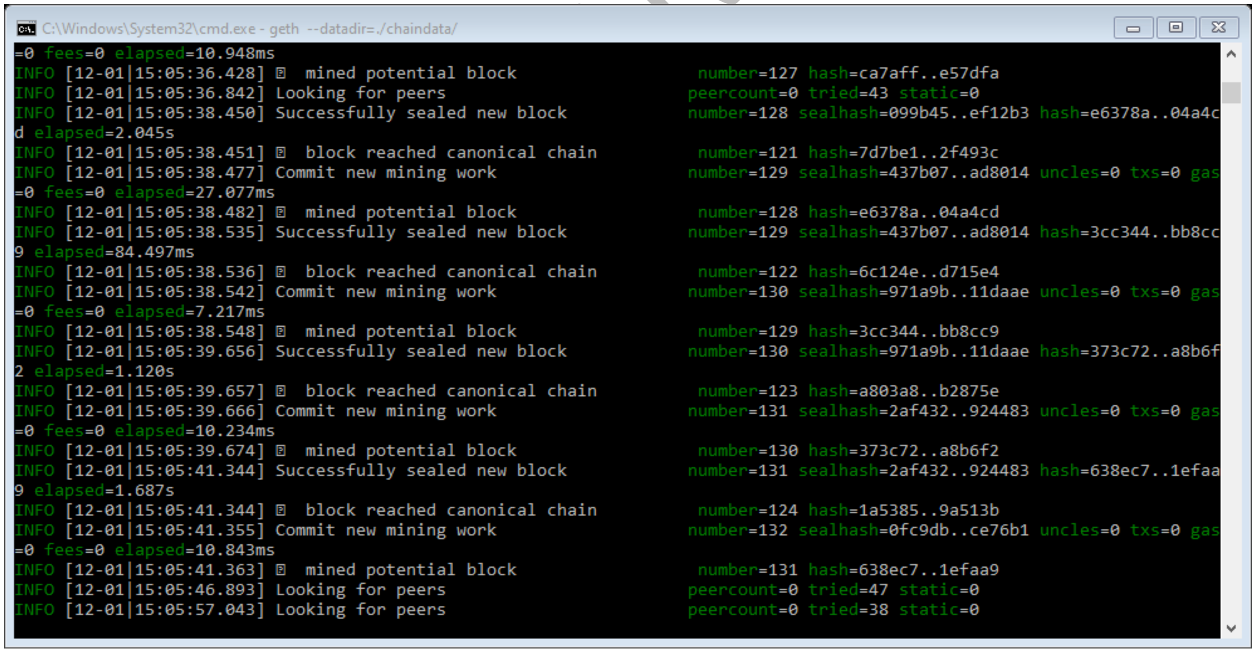
(Run this command on terminal)

Mining is the process of creating a block of transactions to be added to the Ethereum blockchain.



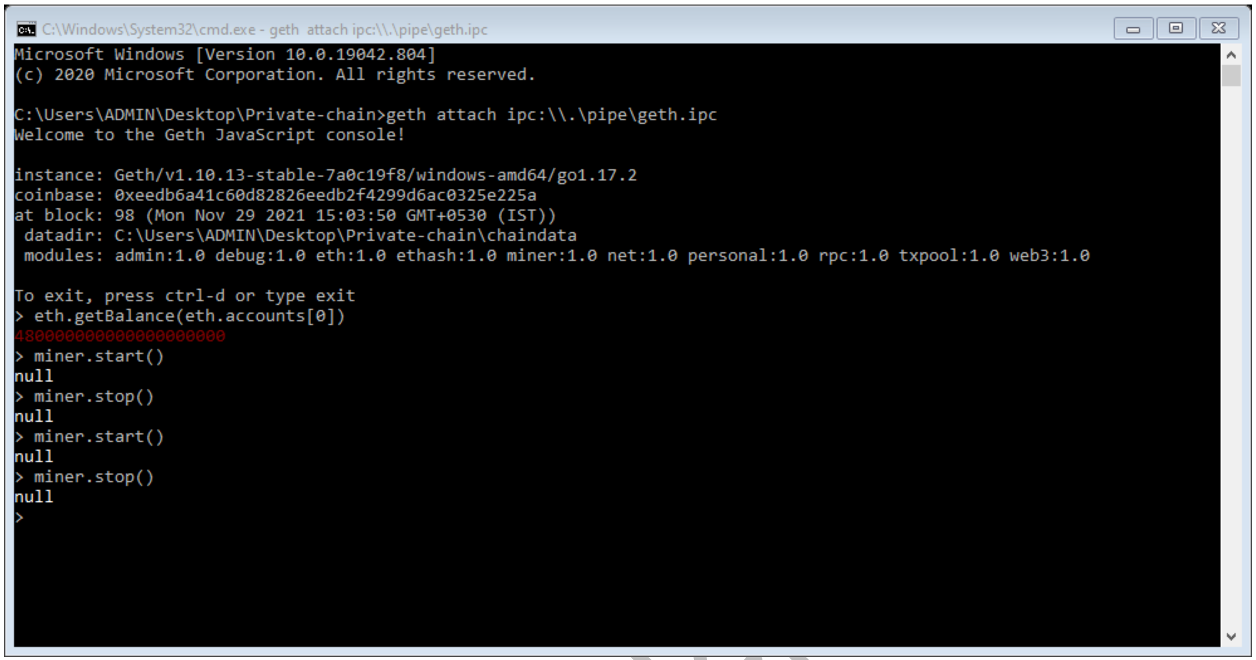
```
Microsoft Windows [Version 10.0.19042.804]
(c) 2020 Microsoft Corporation. All rights reserved.

C:\Users\ADMIN\Desktop\Private-chain>geth attach ipc:\\.\pipe\geth.ipc
Welcome to the Geth JavaScript console!

instance: Geth/v1.10.13-stable-7a0c19f8/windows-amd64/go1.17.2
coinbase: 0xeedb6a41c60d82826eedb2f4299d6ac0325e225a
at block: 98 (Mon Nov 29 2021 15:03:50 GMT+0530 (IST))
 datadir: C:\Users\ADMIN\Desktop\Private-chain\chaindata
 modules: admin:1.0 debug:1.0 eth:1.0 ethash:1.0 miner:1.0 net:1.0 personal:1.0 rpc:1.0 txpool:1.0 web3:1.0

To exit, press ctrl-d or type exit
> eth.getBalance(eth.accounts[0])
480000000000000000000
> miner.start()
null
>
```

(Now observe the terminal which we have minimized.



```
=0 fees=0 elapsed=10.948ms
INFO [12-01|15:05:36.428] 🔨  mined potential block                  number=127 hash=ca7aff..e57dfa
INFO [12-01|15:05:36.842] Looking for peers                         peercount=0 tried=43 static=0
INFO [12-01|15:05:38.450] Successfully sealed new block             number=128 sealhash=099b45..ef12b3 hash=e6378a..04a4c
d elapsed=2.045s
INFO [12-01|15:05:38.451] 🔗  block reached canonical chain          number=121 hash=7d7be1..2f493c
INFO [12-01|15:05:38.477] Commit new mining work                    number=129 sealhash=437b07..ad8014 uncles=0 txs=0 gas
=0 fees=0 elapsed=27.077ms
INFO [12-01|15:05:38.482] 🔨  mined potential block                  number=128 hash=e6378a..04a4cd
INFO [12-01|15:05:38.535] Successfully sealed new block             number=129 sealhash=437b07..ad8014 hash=3cc344..bb8cc
9 elapsed=84.497ms
INFO [12-01|15:05:38.536] 🔗  block reached canonical chain          number=122 hash=6c124e..d715e4
INFO [12-01|15:05:38.542] Commit new mining work                    number=130 sealhash=971a9b..11daae uncles=0 txs=0 gas
=0 fees=0 elapsed=7.217ms
INFO [12-01|15:05:38.548] 🔨  mined potential block                  number=129 hash=3cc344..bb8cc9
INFO [12-01|15:05:39.656] Successfully sealed new block             number=130 sealhash=971a9b..11daae hash=373c72..a8b6f
2 elapsed=1.120s
INFO [12-01|15:05:39.657] 🔗  block reached canonical chain          number=123 hash=a803a8..b2875e
INFO [12-01|15:05:39.666] Commit new mining work                    number=131 sealhash=2af432..924483 uncles=0 txs=0 gas
=0 fees=0 elapsed=10.234ms
INFO [12-01|15:05:39.674] 🔨  mined potential block                  number=130 hash=373c72..a8b6f2
INFO [12-01|15:05:41.344] Successfully sealed new block             number=131 sealhash=2af432..924483 hash=638ec7..1efaa
9 elapsed=1.687s
INFO [12-01|15:05:41.344] 🔗  block reached canonical chain          number=124 hash=1a5385..9a513b
INFO [12-01|15:05:41.355] Commit new mining work                    number=132 sealhash=0fc9db..ce76b1 uncles=0 txs=0 gas
=0 fees=0 elapsed=10.843ms
INFO [12-01|15:05:41.363] 🔨  mined potential block                  number=131 hash=638ec7..1efaa9
INFO [12-01|15:05:46.893] Looking for peers                         peercount=0 tried=47 static=0
INFO [12-01|15:05:57.043] Looking for peers                         peercount=0 tried=38 static=0
```

Wait till you get msg on terminal that successfully sealed new block. Then run following command) miner.stop()

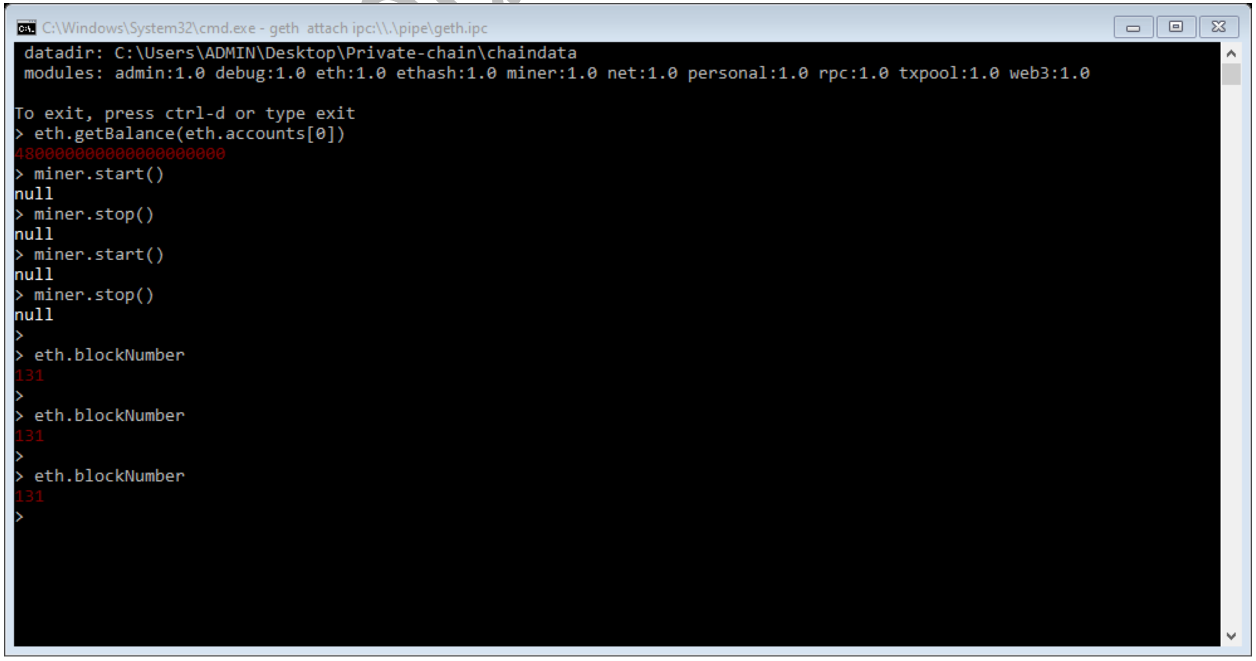


```
C:\Windows\System32\cmd.exe - geth  attach ipc:\\.\pipe\geth.ipc
Microsoft Windows [Version 10.0.19042.804]
(c) 2020 Microsoft Corporation. All rights reserved.

C:\Users\ADMIN\Desktop\Private-chain>geth attach ipc:\\.\pipe\geth.ipc
Welcome to the Geth JavaScript console!

instance: Geth/v1.10.13-stable-7a0c19f8/windows-amd64/go1.17.2
coinbase: 0xeedb6a41c60d82826eedb2f4299d6ac0325e225a
at block: 98 (Mon Nov 29 2021 15:03:50 GMT+0530 (IST))
 datadir: C:\Users\ADMIN\Desktop\Private-chain\chaindata
 modules: admin:1.0 debug:1.0 eth:1.0 ethash:1.0 miner:1.0 net:1.0 personal:1.0 rpc:1.0 txpool:1.0 web3:1.0

To exit, press ctrl-d or type exit
> eth.getBalance(eth.accounts[0])
480000000000000000000
> miner.start()
null
> miner.stop()
null
> miner.start()
null
> miner.stop()
null
>
```

eth.blockNumber

(Run this command on terminal)

It will show you blocknumber.



```
C:\Windows\System32\cmd.exe - geth  attach ipc:\\.\pipe\geth.ipc
 datadir: C:\Users\ADMIN\Desktop\Private-chain\chaindata
 modules: admin:1.0 debug:1.0 eth:1.0 ethash:1.0 miner:1.0 net:1.0 personal:1.0 rpc:1.0 txpool:1.0 web3:1.0

To exit, press ctrl-d or type exit
> eth.getBalance(eth.accounts[0])
480000000000000000000
> miner.start()
null
> miner.stop()
null
> miner.start()
null
> miner.stop()
null
>
> eth.blockNumber
131
>
> eth.blockNumber
131
>
> eth.blockNumber
131
>
```

Now we will perform transaction

To perform transaction, we are having one account. Now we will create another account.

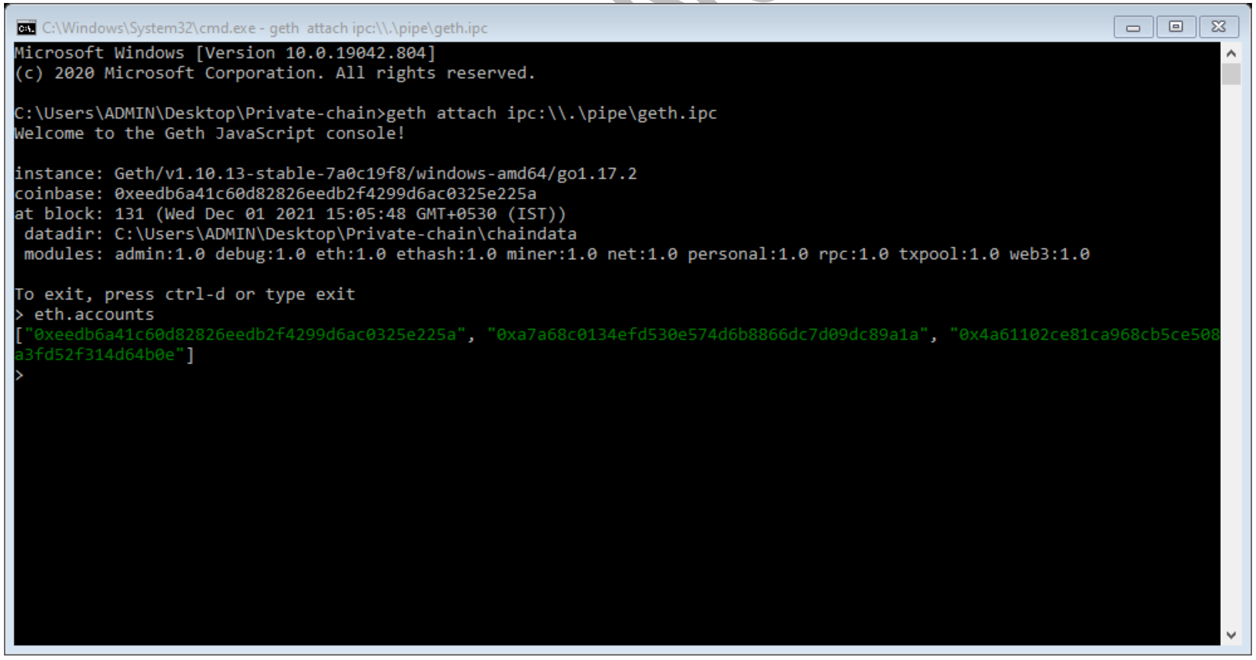
personal.newAccount()

(Run this command on terminal it will create new account)

eth.accounts

(Run this command on terminal)

It will show number of accounts created

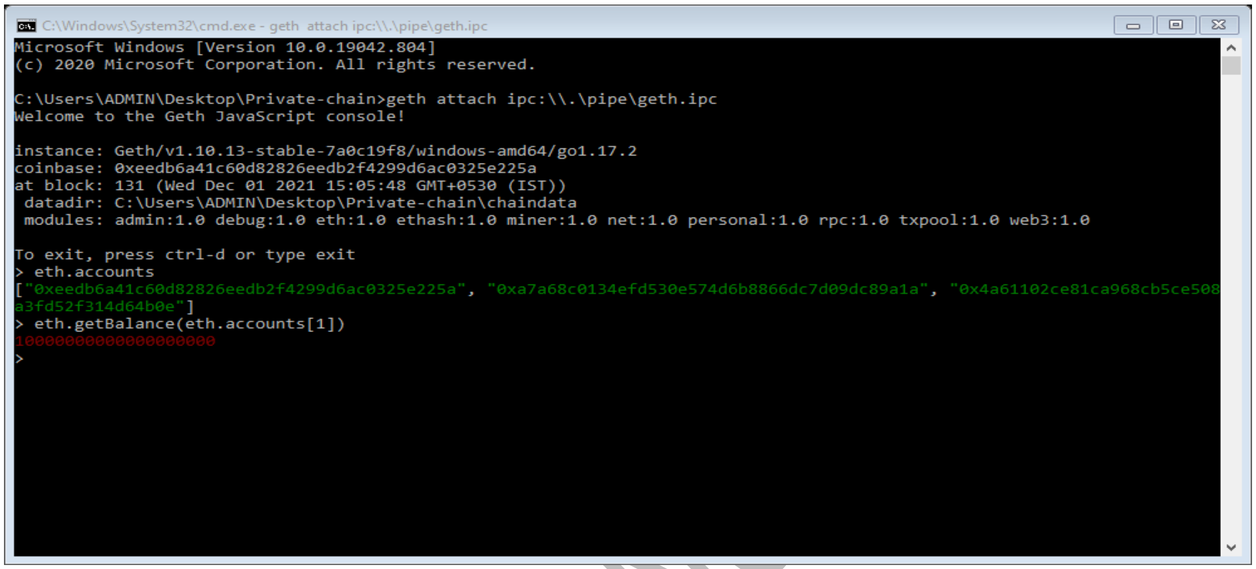

C:\Windows\System32\cmd.exe - geth attach ipc:\\.\pipe\geth.ipc

```
Microsoft Windows [Version 10.0.19042.804]
(c) 2020 Microsoft Corporation. All rights reserved.

C:\Users\ADMIN\Desktop\Private-chain>geth attach ipc:\\.\pipe\geth.ipc
Welcome to the Geth JavaScript console!

instance: Geth/v1.10.13-stable-7a0c19f8/windows-amd64/go1.17.2
coinbase: 0xeedb6a41c60d82826eedb2f4299d6ac0325e225a
at block: 131 (Wed Dec 01 2021 15:05:48 GMT+0530 (IST))
 datadir: C:\Users\ADMIN\Desktop\Private-chain\chaindata
 modules: admin:1.0 debug:1.0 eth:1.0 ethash:1.0 miner:1.0 net:1.0 personal:1.0 rpc:1.0 txpool:1.0 web3:1.0

To exit, press ctrl-d or type exit
> eth.accounts
["0xeedb6a41c60d82826eedb2f4299d6ac0325e225a", "0xa7a68c0134efd530e574d6b8866dc7d09dc89a1a", "0x4a61102ce81ca968cb5ce508
a3fd52f314d64b0e"]
>
```

eth.getBalance(eth.accounts[1])

(Run this command on terminal)

It will show balance of account.

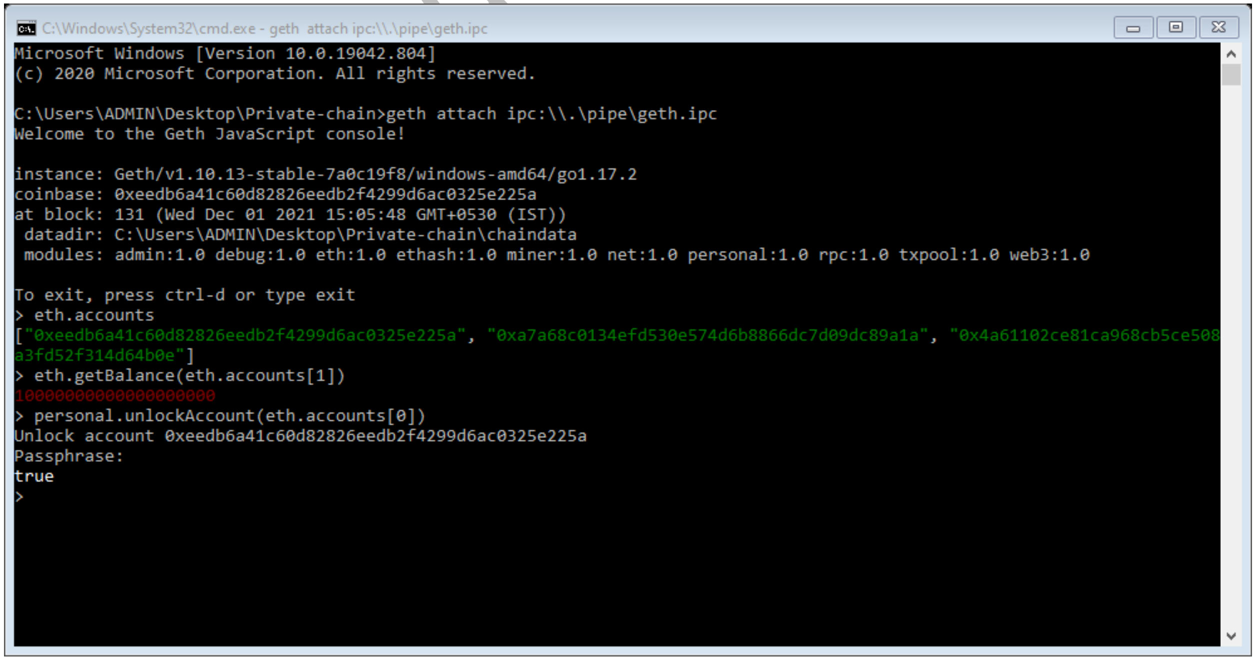


```
C:\Windows\System32\cmd.exe - geth  attach ipc:\\.\pipe\geth.ipc
Microsoft Windows [Version 10.0.19042.804]
(c) 2020 Microsoft Corporation. All rights reserved.

C:\Users\ADMIN\Desktop\Private-chain>geth attach ipc:\\.\pipe\geth.ipc
Welcome to the Geth JavaScript console!

instance: Geth/v1.10.13-stable-7a0c19f8/windows-amd64/go1.17.2
coinbase: 0xeedb6a41c60d82826eedb2f4299d6ac0325e225a
at block: 131 (Wed Dec 01 2021 15:05:48 GMT+0530 (IST))
 datadir: C:\Users\ADMIN\Desktop\Private-chain\chaindata
 modules: admin:1.0 debug:1.0 eth:1.0 ethash:1.0 miner:1.0 net:1.0 personal:1.0 rpc:1.0 txpool:1.0 web3:1.0

To exit, press ctrl-d or type exit
> eth.accounts
["0xeedb6a41c60d82826eedb2f4299d6ac0325e225a", "0xa7a68c0134efd530e574d6b8866dc7d09dc89a1a", "0x4a61102ce81ca968cb5ce508
a3fd52f314d64b0e"]
> eth.getBalance(eth.accounts[1])
10000000000000000000
>
```

personal.unlockAccount(eth.accounts[0])

(Run this command on terminal)

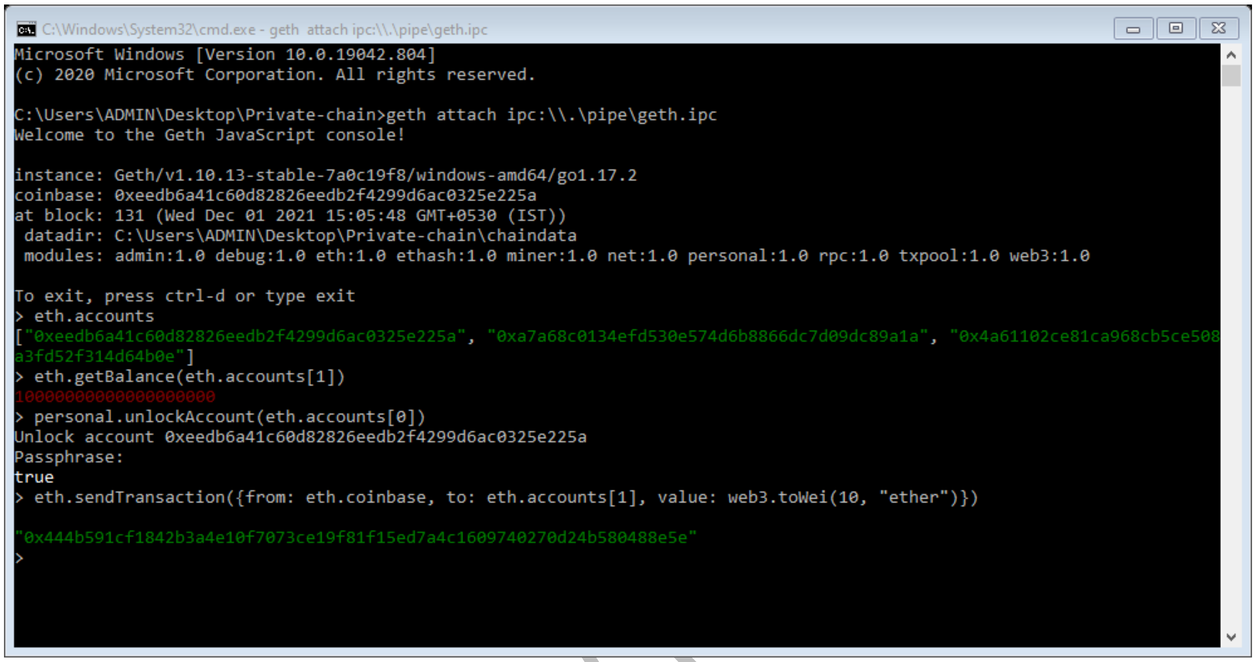
It will unlock the block which was sealed previously, it requires password)



```
C:\Windows\System32\cmd.exe - geth  attach ipc:\\.\pipe\geth.ipc
Microsoft Windows [Version 10.0.19042.804]
(c) 2020 Microsoft Corporation. All rights reserved.

C:\Users\ADMIN\Desktop\Private-chain>geth attach ipc:\\.\pipe\geth.ipc
Welcome to the Geth JavaScript console!

instance: Geth/v1.10.13-stable-7a0c19f8/windows-amd64/go1.17.2
coinbase: 0xeedb6a41c60d82826eedb2f4299d6ac0325e225a
at block: 131 (Wed Dec 01 2021 15:05:48 GMT+0530 (IST))
 datadir: C:\Users\ADMIN\Desktop\Private-chain\chaindata
 modules: admin:1.0 debug:1.0 eth:1.0 ethash:1.0 miner:1.0 net:1.0 personal:1.0 rpc:1.0 txpool:1.0 web3:1.0

To exit, press ctrl-d or type exit
> eth.accounts
["0xeedb6a41c60d82826eedb2f4299d6ac0325e225a", "0xa7a68c0134efd530e574d6b8866dc7d09dc89a1a", "0x4a61102ce81ca968cb5ce508
a3fd52f314d64b0e"]
> eth.getBalance(eth.accounts[1])
10000000000000000000
> personal.unlockAccount(eth.accounts[0])
Unlock account 0xeedb6a41c60d82826eedb2f4299d6ac0325e225a
Passphrase:
true
>
```

Transaction:

eth.sendTransaction({from: eth.coinbase, to: eth.accounts[1], value: web3.toWei(10, "ether")})

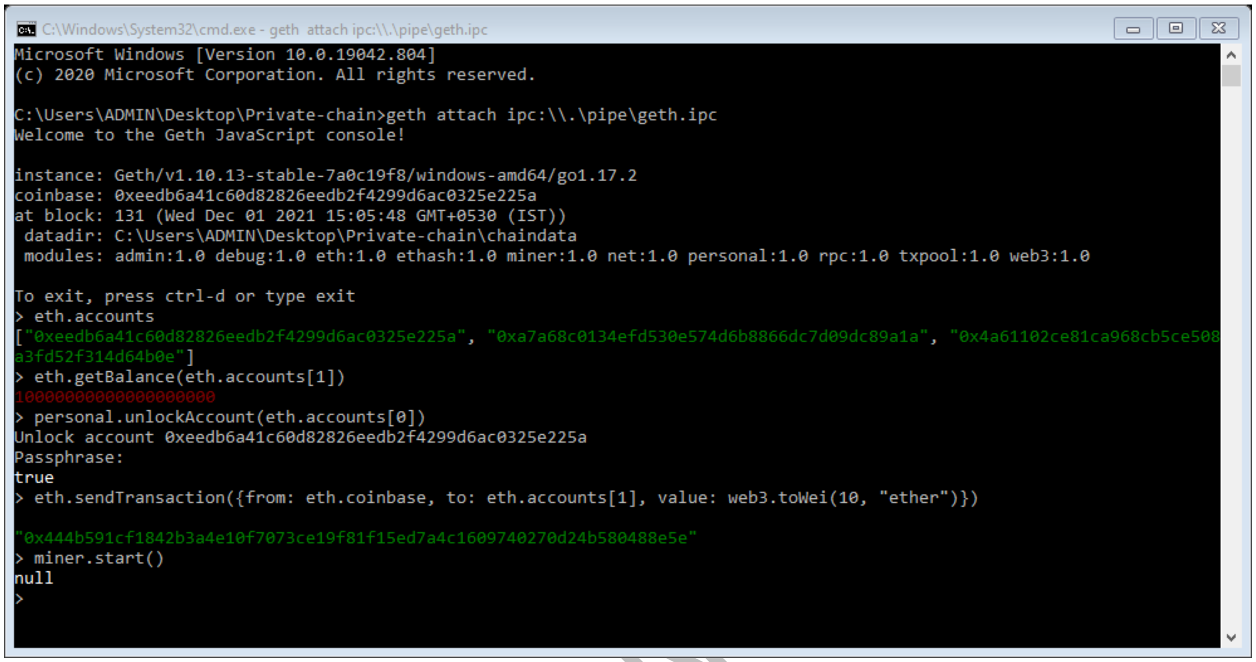

C:\Windows\System32\cmd.exe - geth attach ipc:\\.\pipe\geth.ipc

```
Microsoft Windows [Version 10.0.19042.804]
(c) 2020 Microsoft Corporation. All rights reserved.

C:\Users\ADMIN\Desktop\Private-chain>geth attach ipc:\\.\pipe\geth.ipc
Welcome to the Geth JavaScript console!

instance: Geth/v1.10.13-stable-7a0c19f8/windows-amd64/go1.17.2
coinbase: 0xeedb6a41c60d82826eedb2f4299d6ac0325e225a
at block: 131 (Wed Dec 01 2021 15:05:48 GMT+0530 (IST))
 datadir: C:\Users\ADMIN\Desktop\Private-chain\chaindata
 modules: admin:1.0 debug:1.0 eth:1.0 ethash:1.0 miner:1.0 net:1.0 personal:1.0 rpc:1.0 txpool:1.0 web3:1.0

To exit, press ctrl-d or type exit
> eth.accounts
["0xeedb6a41c60d82826eedb2f4299d6ac0325e225a", "0xa7a68c0134efd530e574d6b8866dc7d09dc89a1a", "0x4a61102ce81ca968cb5ce508
a3fd52f314d64b0e"]
> eth.getBalance(eth.accounts[1])
10000000000000000000
> personal.unlockAccount(eth.accounts[0])
Unlock account 0xeedb6a41c60d82826eedb2f4299d6ac0325e225a
Passphrase:
true
> eth.sendTransaction({from: eth.coinbase, to: eth.accounts[1], value: web3.toWei(10, "ether")})

"0x444b591cf1842b3a4e10f7073ce19f81f15ed7a4c1609740270d24b580488e5e"
>
```

(start mining and stop)

miner.start()

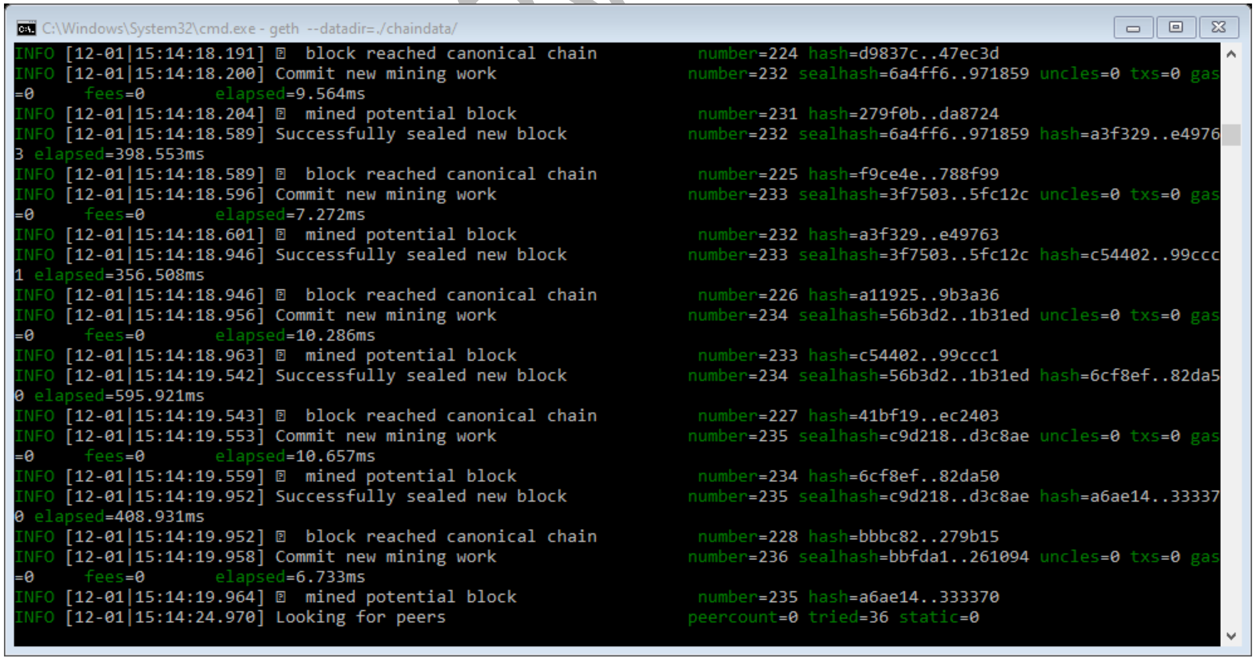
(Run this command on terminal)



C:\Windows\System32\cmd.exe - geth attach ipc:\\.\pipe\geth.ipc

```
Microsoft Windows [Version 10.0.19042.804]
(c) 2020 Microsoft Corporation. All rights reserved.

C:\Users\ADMIN\Desktop\Private-chain>geth attach ipc:\\.\pipe\geth.ipc
Welcome to the Geth JavaScript console!

instance: Geth/v1.10.13-stable-7a0c19f8/windows-amd64/go1.17.2
coinbase: 0xeedb6a41c60d82826eedb2f4299d6ac0325e225a
at block: 131 (Wed Dec 01 2021 15:05:48 GMT+0530 (IST))
 datadir: C:\Users\ADMIN\Desktop\Private-chain\chaindata
 modules: admin:1.0 debug:1.0 eth:1.0 ethash:1.0 miner:1.0 net:1.0 personal:1.0 rpc:1.0 txpool:1.0 web3:1.0

To exit, press ctrl-d or type exit
> eth.accounts
["0xeedb6a41c60d82826eedb2f4299d6ac0325e225a", "0xa7a68c0134efd530e574d6b8866dc7d09dc89a1a", "0x4a61102ce81ca968cb5ce508
a3fd52f314d64b0e"]
> eth.getBalance(eth.accounts[1])
10000000000000000000
> personal.unlockAccount(eth.accounts[0])
Unlock account 0xeedb6a41c60d82826eedb2f4299d6ac0325e225a
Passphrase:
true
> eth.sendTransaction({from: eth.coinbase, to: eth.accounts[1], value: web3.toWei(10, "ether")})

"0x444b591cf1842b3a4e10f7073ce19f81f15ed7a4c1609740270d24b580488e5e"
> miner.start()
null
>
```

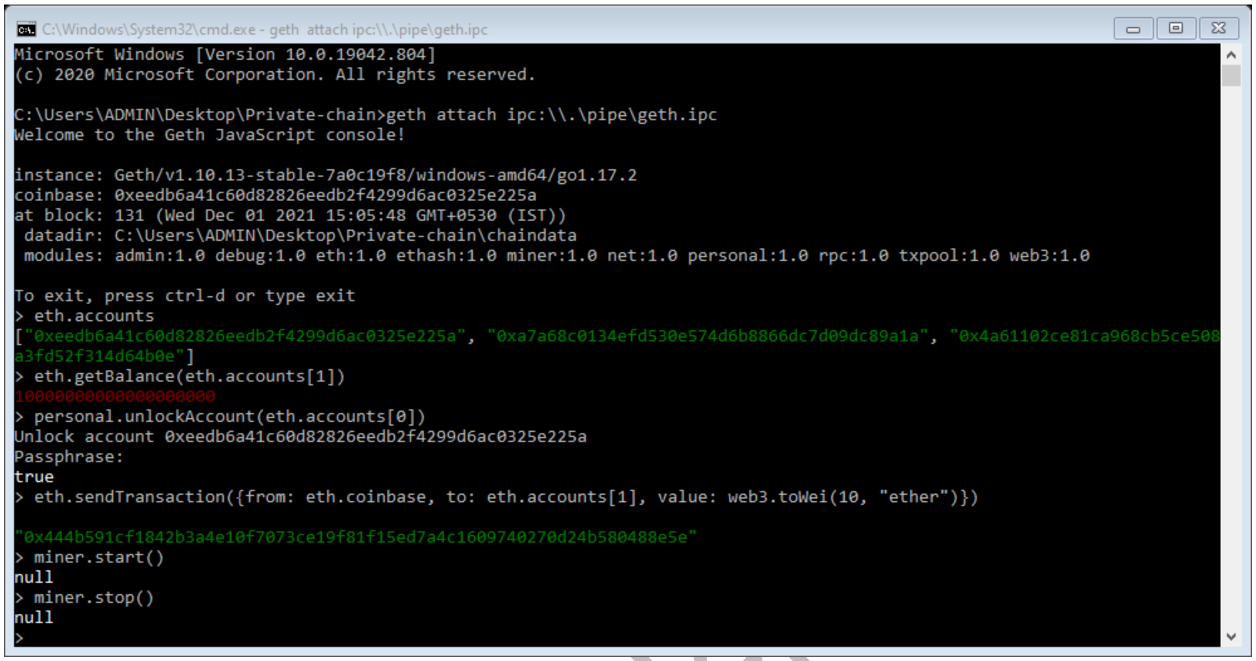
(Now observe the terminal which we have minimized.



C:\Windows\System32\cmd.exe - geth --datadir=./chaindata/

```
INFO [12-01|15:14:18.191] □  block reached canonical chain          number=224 hash=d9837c..47ec3d
INFO [12-01|15:14:18.200] Commit new mining work                   number=232 sealhash=6a4ff6..971859 uncles=0 txs=0 gas
=0     fees=0        elapsed=9.564ms
INFO [12-01|15:14:18.204] □  mined potential block                  number=231 hash=279f0b..da8724
INFO [12-01|15:14:18.589] Successfully sealed new block            number=232 sealhash=6a4ff6..971859 hash=a3f329..e4976
3 elapsed=398.553ms
INFO [12-01|15:14:18.589] □  block reached canonical chain          number=225 hash=f9ce4e..788f99
INFO [12-01|15:14:18.596] Commit new mining work                   number=233 sealhash=3f7503..5fc12c uncles=0 txs=0 gas
=0     fees=0        elapsed=7.272ms
INFO [12-01|15:14:18.601] □  mined potential block                  number=232 hash=a3f329..e49763
INFO [12-01|15:14:18.946] Successfully sealed new block            number=233 sealhash=3f7503..5fc12c hash=c54402..99ccc
1 elapsed=356.508ms
INFO [12-01|15:14:18.946] □  block reached canonical chain          number=226 hash=a11925..9b3a36
INFO [12-01|15:14:18.956] Commit new mining work                   number=234 sealhash=56b3d2..1b31ed uncles=0 txs=0 gas
=0     fees=0        elapsed=10.286ms
INFO [12-01|15:14:18.963] □  mined potential block                  number=233 hash=c54402..99ccc1
INFO [12-01|15:14:19.542] Successfully sealed new block            number=234 sealhash=56b3d2..1b31ed hash=6cf8ef..82da5
0 elapsed=595.921ms
INFO [12-01|15:14:19.543] □  block reached canonical chain          number=227 hash=41bf19..ec2403
INFO [12-01|15:14:19.553] Commit new mining work                   number=235 sealhash=c9d218..d3c8ae uncles=0 txs=0 gas
=0     fees=0        elapsed=10.657ms
INFO [12-01|15:14:19.559] □  mined potential block                  number=234 hash=6cf8ef..82da50
INFO [12-01|15:14:19.952] Successfully sealed new block            number=235 sealhash=c9d218..d3c8ae hash=a6ae14..33337
0 elapsed=408.931ms
INFO [12-01|15:14:19.952] □  block reached canonical chain          number=228 hash=bbbc82..279b15
INFO [12-01|15:14:19.958] Commit new mining work                   number=236 sealhash=bbfda1..261094 uncles=0 txs=0 gas
=0     fees=0        elapsed=6.733ms
INFO [12-01|15:14:19.964] □  mined potential block                  number=235 hash=a6ae14..333370
INFO [12-01|15:14:24.970] Looking for peers                        peercount=0 tried=36 static=0
```

Wait till you get msg on terminal that successfully sealed. Then run following command) miner.stop()

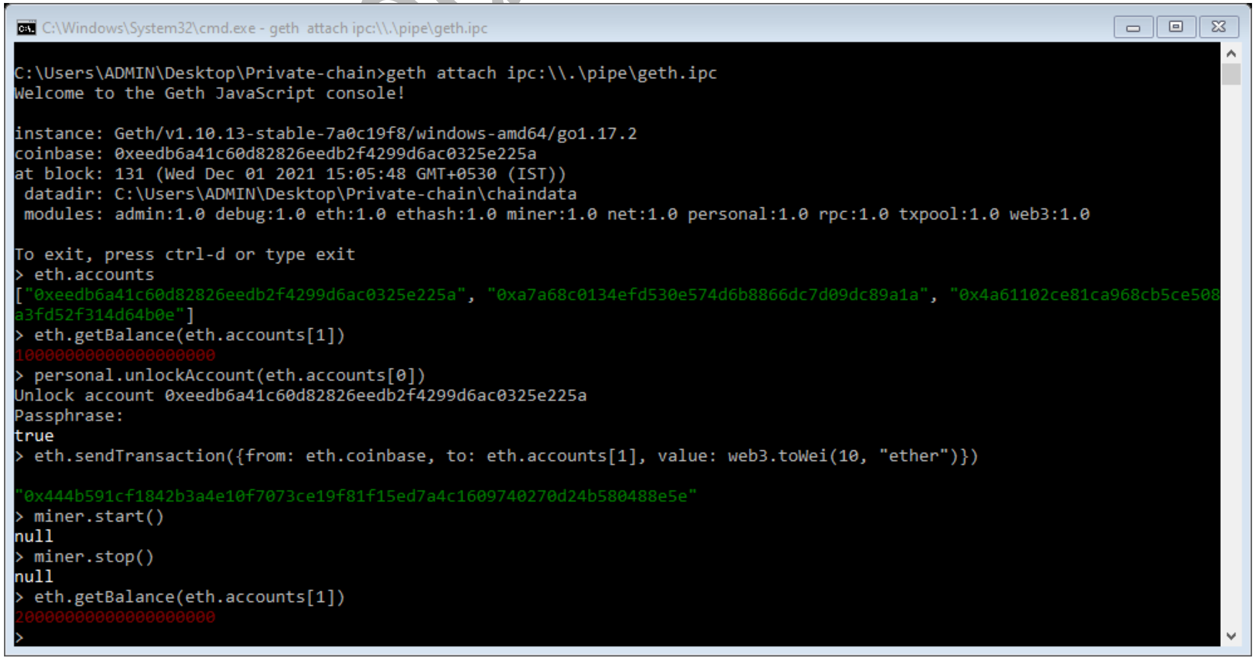

C:\Windows\System32\cmd.exe - geth attach ipc:\\.\pipe\geth.ipc

```
Microsoft Windows [Version 10.0.19042.804]
(c) 2020 Microsoft Corporation. All rights reserved.

C:\Users\ADMIN\Desktop\Private-chain>geth attach ipc:\\.\pipe\geth.ipc
Welcome to the Geth JavaScript console!

instance: Geth/v1.10.13-stable-7a0c19f8/windows-amd64/go1.17.2
coinbase: 0xeedb6a41c60d82826eedb2f4299d6ac0325e225a
at block: 131 (Wed Dec 01 2021 15:05:48 GMT+0530 (IST))
 datadir: C:\Users\ADMIN\Desktop\Private-chain\chaindata
 modules: admin:1.0 debug:1.0 eth:1.0 ethash:1.0 miner:1.0 net:1.0 personal:1.0 rpc:1.0 txpool:1.0 web3:1.0

To exit, press ctrl-d or type exit
> eth.accounts
["0xeedb6a41c60d82826eedb2f4299d6ac0325e225a", "0xa7a68c0134efd530e574d6b8866dc7d09dc89a1a", "0x4a61102ce81ca968cb5ce508
a3fd52f314d64b0e"]
> eth.getBalance(eth.accounts[1])
10000000000000000000
> personal.unlockAccount(eth.accounts[0])
Unlock account 0xeedb6a41c60d82826eedb2f4299d6ac0325e225a
Passphrase:
true
> eth.sendTransaction({from: eth.coinbase, to: eth.accounts[1], value: web3.toWei(10, "ether")})

"0x444b591cf1842b3a4e10f7073ce19f81f15ed7a4c1609740270d24b580488e5e"
> miner.start()
null
> miner.stop()
null
>
```

eth.getBalance(eth.accounts[1])

(Run this command on terminal)

It will show balance of account.


C:\Windows\System32\cmd.exe - geth attach ipc:\\.\pipe\geth.ipc

```
C:\Users\ADMIN\Desktop\Private-chain>geth attach ipc:\\.\pipe\geth.ipc
Welcome to the Geth JavaScript console!

instance: Geth/v1.10.13-stable-7a0c19f8/windows-amd64/go1.17.2
coinbase: 0xeedb6a41c60d82826eedb2f4299d6ac0325e225a
at block: 131 (Wed Dec 01 2021 15:05:48 GMT+0530 (IST))
 datadir: C:\Users\ADMIN\Desktop\Private-chain\chaindata
 modules: admin:1.0 debug:1.0 eth:1.0 ethash:1.0 miner:1.0 net:1.0 personal:1.0 rpc:1.0 txpool:1.0 web3:1.0

To exit, press ctrl-d or type exit
> eth.accounts
["0xeedb6a41c60d82826eedb2f4299d6ac0325e225a", "0xa7a68c0134efd530e574d6b8866dc7d09dc89a1a", "0x4a61102ce81ca968cb5ce508
a3fd52f314d64b0e"]
> eth.getBalance(eth.accounts[1])
10000000000000000000
> personal.unlockAccount(eth.accounts[0])
Unlock account 0xeedb6a41c60d82826eedb2f4299d6ac0325e225a
Passphrase:
true
> eth.sendTransaction({from: eth.coinbase, to: eth.accounts[1], value: web3.toWei(10, "ether")})

"0x444b591cf1842b3a4e10f7073ce19f81f15ed7a4c1609740270d24b580488e5e"
> miner.start()
null
> miner.stop()
null
> eth.getBalance(eth.accounts[1])
20000000000000000000
>
```