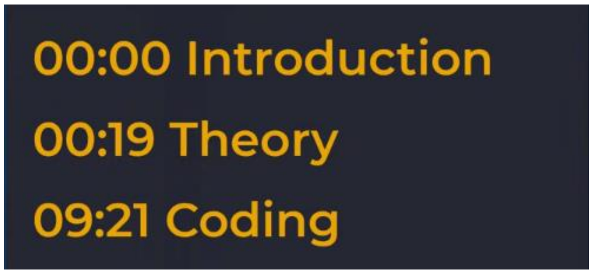
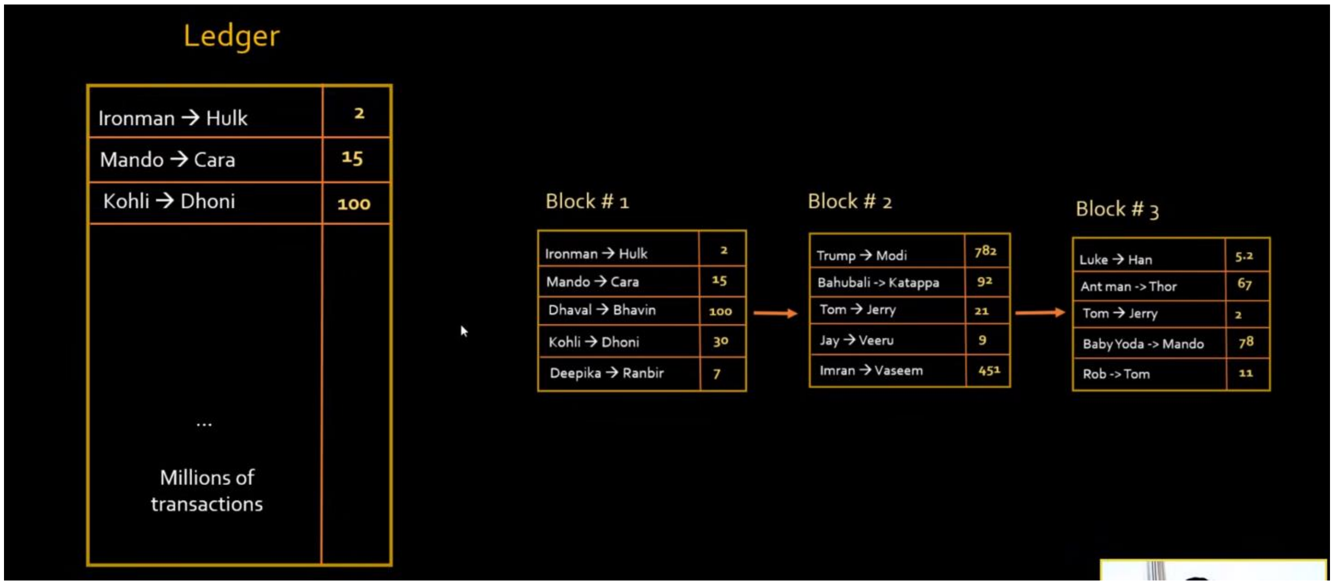


00:00 Introduction

00:19 Theory

09:21 Coding



Ledger

Ironman → Hulk	2
Mando → Cara	15
Kohli → Dhoni	100
… Millions of transactions	

Block # 1

Ironman → Hulk	2
Mando → Cara	15
Dhaval → Bhavin	100
Kohli → Dhoni	30
Deepika → Ranbir	7

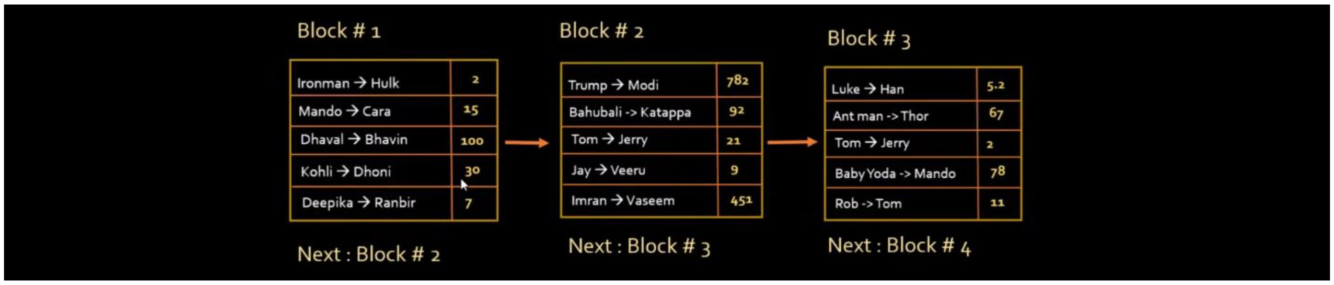
Block # 2

Trump → Modi	782
Bahubali -> Katappa	92
Tom → Jerry	21
Jay → Veeru	9
Imran → Vaseem	451

Block # 3

Luke → Han	5.2
Ant man -> Thor	67
Tom → Jerry	2
Baby Yoda -> Mando	78
Rob -> Tom	11

This is a ledger with transactions that are happening. We need to store the transactions in blocks using a linked list structure with blocks of a specific size like 1MB blocks.



Block # 1

Ironman → Hulk	2
Mando → Cara	15
Dhaval → Bhavin	100
Kohli → Dhoni	30
Deepika → Ranbir	7

Next : Block # 2

Block # 2

Trump → Modi	782
Bahubali -> Katappa	92
Tom → Jerry	21
Jay → Veeru	9
Imran → Vaseem	451

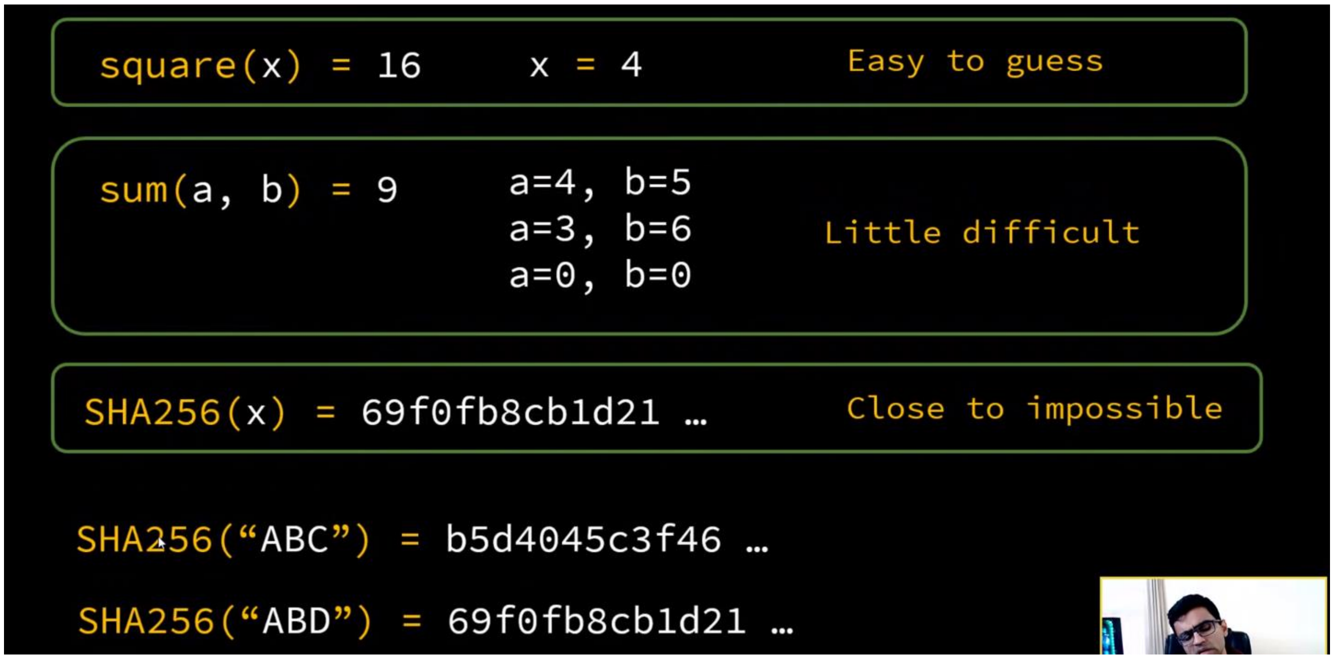
Next : Block # 3

Block # 3

Luke → Han	5.2
Ant man -> Thor	67
Tom → Jerry	2
Baby Yoda -> Mando	78
Rob -> Tom	11

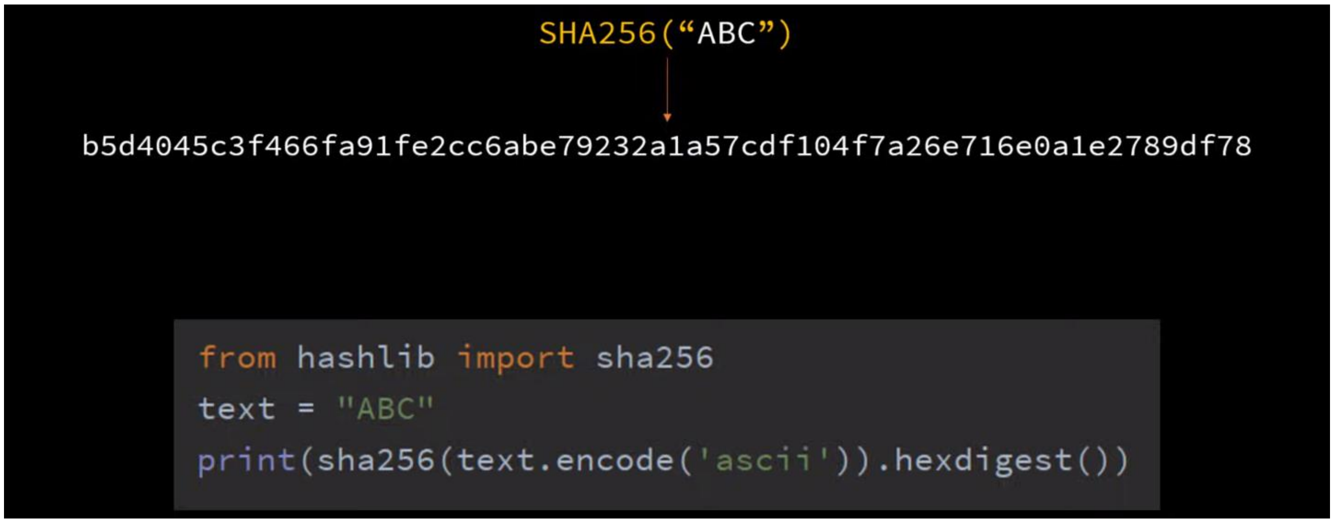
Next : Block # 4

Bitcoin blocks use cryptography to ensure security that the blocks have not been tampered with



```
square(x) = 16      x = 4          Easy to guess

sum(a, b) = 9       a=4, b=5
                    a=3, b=6       Little difficult
                    a=0, b=0

SHA256(x) = 69f0fb8cb1d21 …        Close to impossible

SHA256("ABC") = b5d4045c3f46 …

SHA256("ABD") = 69f0fb8cb1d21 …
```

Using SHA256 function, we give a function and get a 256bit hash value back. This SHA values are almost impossible to guess or hack.



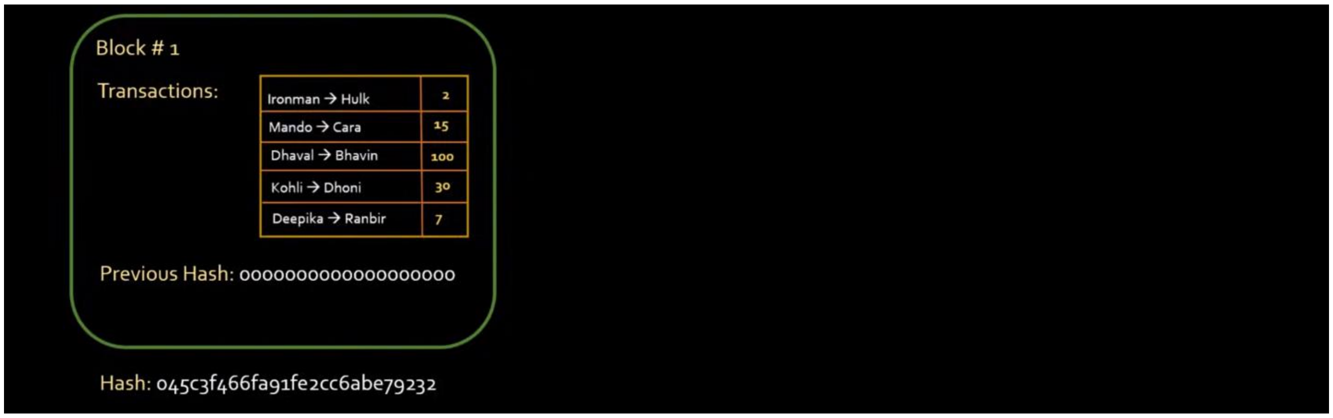
SHA256("ABC")

```
b5d4045c3f466fa91fe2cc6abe79232a1a57cdf104f7a26e716e0a1e2789df78
```

```python
from hashlib import sha256
text = "ABC"
print(sha256(text.encode('ascii')).hexdigest())
```

You can write the 3 lines of python code to get the SHA256 value for the block

SHA256 is a cryptographic hash function


Block # 1

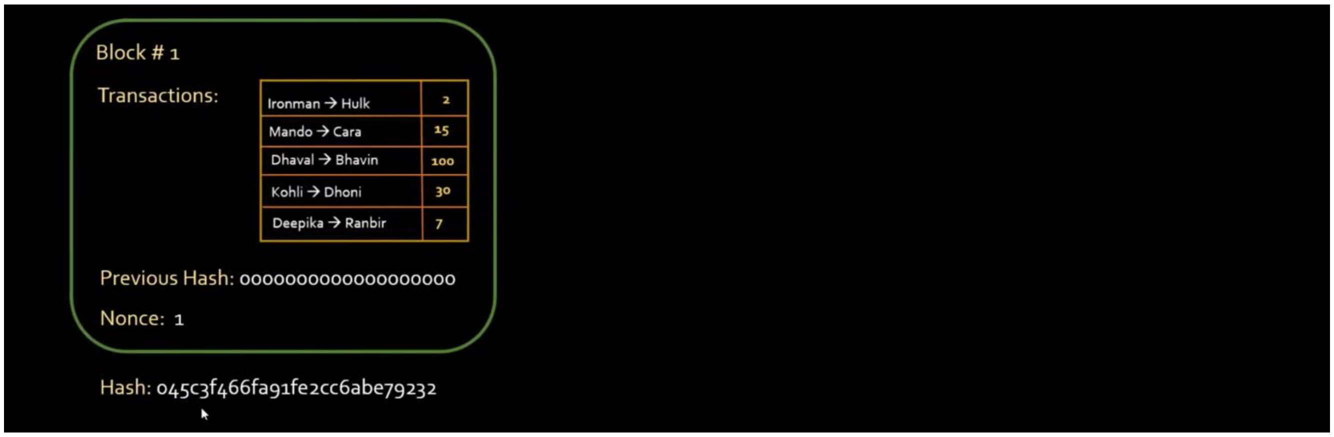
Transactions:

Ironman → Hulk	2
Mando → Cara	15
Dhaval → Bhavin	100
Kohli → Dhoni	30
Deepika → Ranbir	7

Previous Hash: 000000000000000000

Hash: 045c3f466fa91fe2cc6abe79232

The protocol requires that certain starting values of the SHA256 hash value should be zero, this is the level of difficulty because we have to introduce a guess value called a Nonce to fulfil the difficulty using a for-loop.


Block # 1

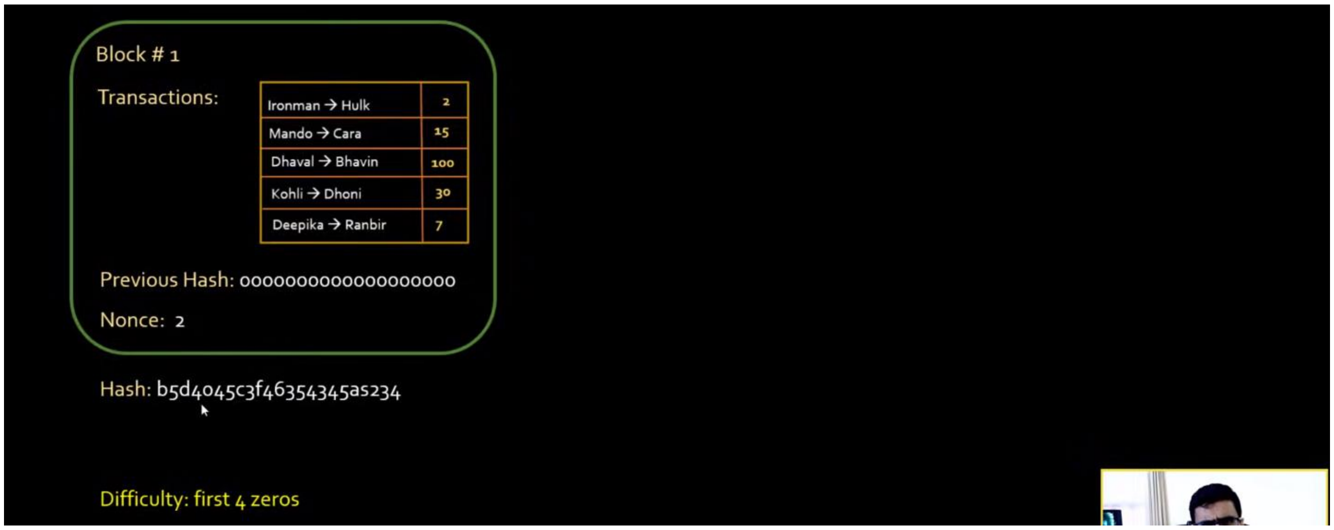
Transactions:

Ironman → Hulk	2
Mando → Cara	15
Dhaval → Bhavin	100
Kohli → Dhoni	30
Deepika → Ranbir	7

Previous Hash: 000000000000000000

Nonce: 1

Hash: 045c3f466fa91fe2cc6abe79232


Block # 1

Transactions:

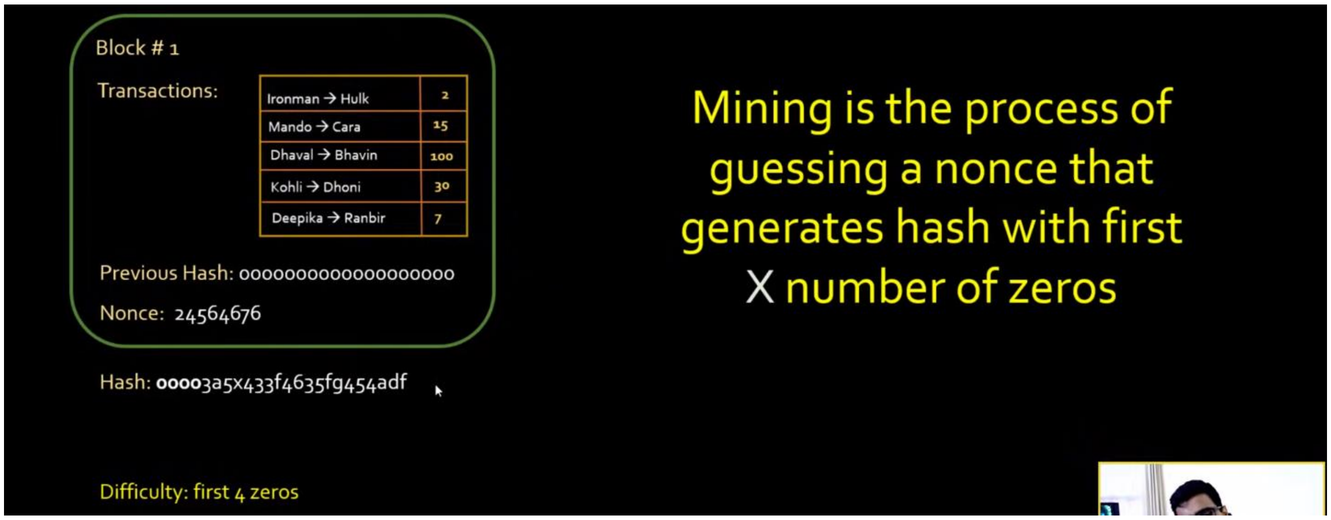
Ironman → Hulk	2
Mando → Cara	15
Dhaval → Bhavin	100
Kohli → Dhoni	30
Deepika → Ranbir	7

Previous Hash: 000000000000000000

Nonce: 2

Hash: b5d4045c3f46354345a5234

Difficulty: first 4 zeros


Block # 1

Transactions:

Ironman → Hulk	2
Mando → Cara	15
Dhaval → Bhavin	100
Kohli → Dhoni	30
Deepika → Ranbir	7

Previous Hash: 0000000000000000000

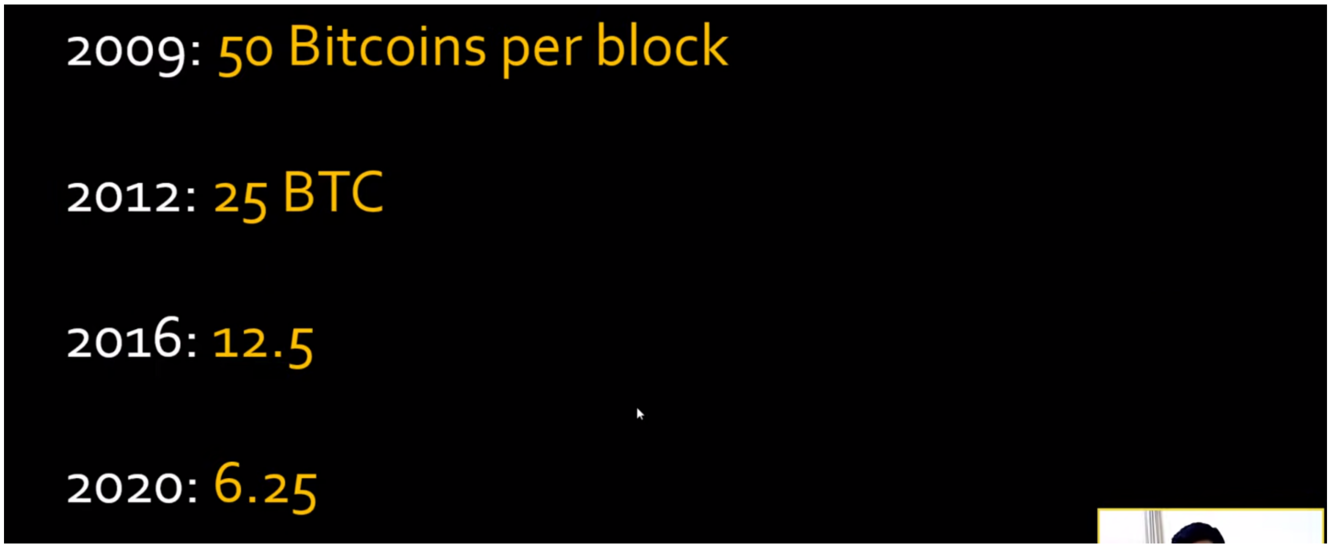
Nonce: 24564676

Hash: 00003a5x433f4635fg454adf

Difficulty: first 4 zeros

Mining is the process of guessing a nonce that generates hash with first X number of zeros

The process of guessing a correct nonce value to have the expected hash value with the required leading zeros is called Bitcoin mining.

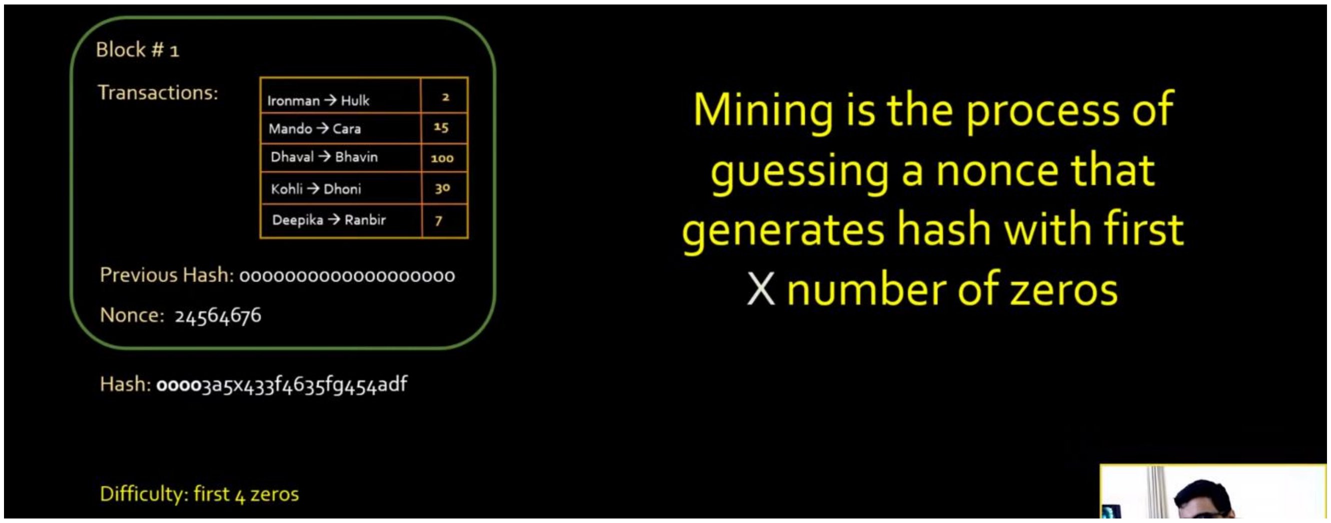

2009: 50 Bitcoins per block

2012: 25 BTC

2016: 12.5

2020: 6.25

Miners get a reward for doing bitcoin mining, the reward gets halved every 4 years. The reward is currently 6.25BTC today. Bitcoin is about guessing the correct nonce value and getting a reward. Currently, miners get 6.25 BTC per valid block mined. But this reward changes roughly every four years, or after every 210,000 blocks are mined and gets reduced by half each time. ... After the first halving, the reward was reduced to 25 BTC, then to 12.5 BTC and finally to 6.25 BTC.


Block # 1

Transactions:

Ironman → Hulk	2
Mando → Cara	15
Dhaval → Bhavin	100
Kohli → Dhoni	30
Deepika → Ranbir	7

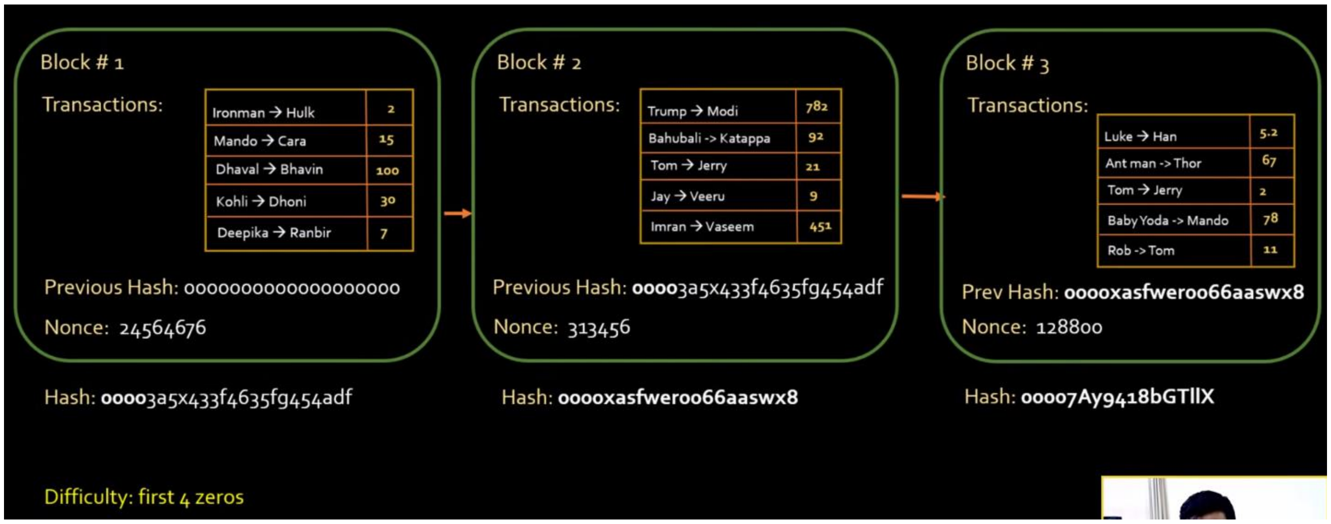
Previous Hash: 0000000000000000000

Nonce: 24564676

Hash: **0000**3a5x433f4635fg454adf

Mining is the process of guessing a nonce that generates hash with first X number of zeros

Difficulty: first 4 zeros


Block # 1

Transactions:

Ironman → Hulk	2
Mando → Cara	15
Dhaval → Bhavin	100
Kohli → Dhoni	30
Deepika → Ranbir	7

Previous Hash: 0000000000000000000

Nonce: 24564676

Hash: **0000**3a5x433f4635fg454adf

Block # 2

Transactions:

Trump → Modi	782
Bahubali -> Katappa	92
Tom → Jerry	21
Jay → Veeru	9
Imran → Vaseem	451

Previous Hash: **0000**3a5x433f4635fg454adf

Nonce: 313456

Hash: **0000xasfwero066aaswx8**

Block # 3

Transactions:

Luke → Han	5.2
Ant man -> Thor	67
Tom → Jerry	2
Baby Yoda -> Mando	78
Rob -> Tom	11

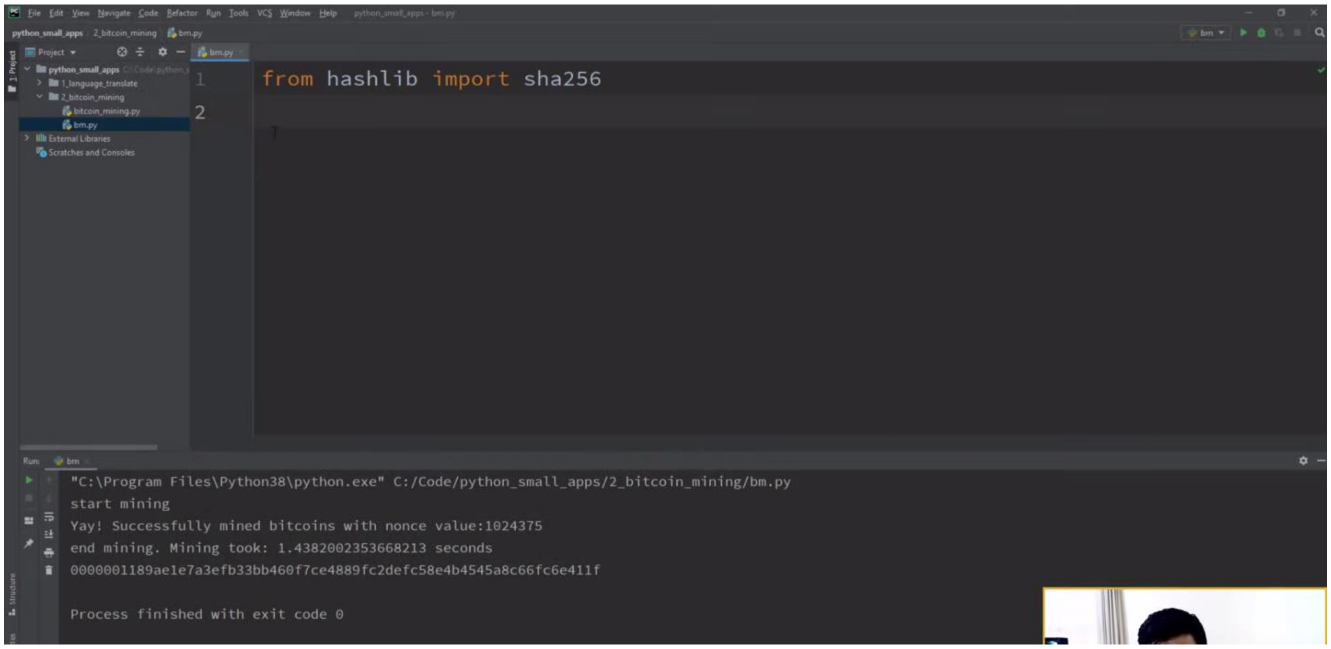
Prev Hash: **0000xasfwero066aaswx8**

Nonce: 128800

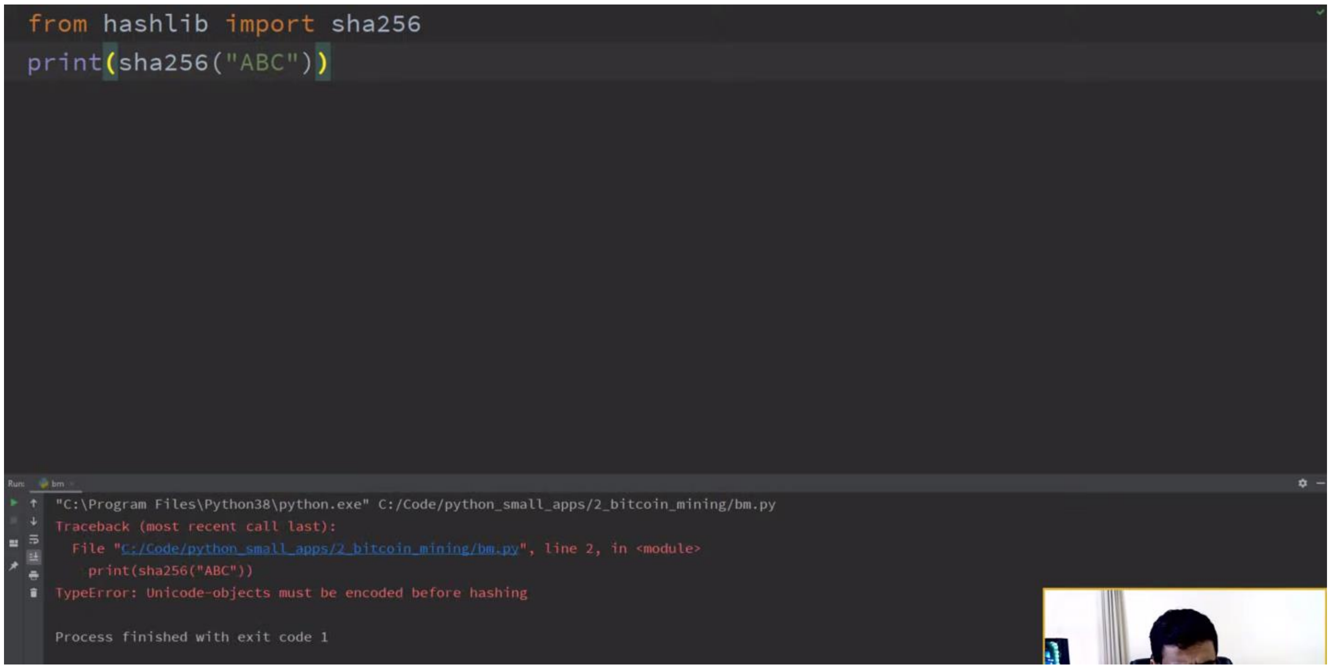
Hash: **00007Ay9418bGTllX**

Difficulty: first 4 zeros

In the end, the blockchain looks like the above where every block will have a reference to the previous blocks' hash value. The blocks become a chain as they are added.

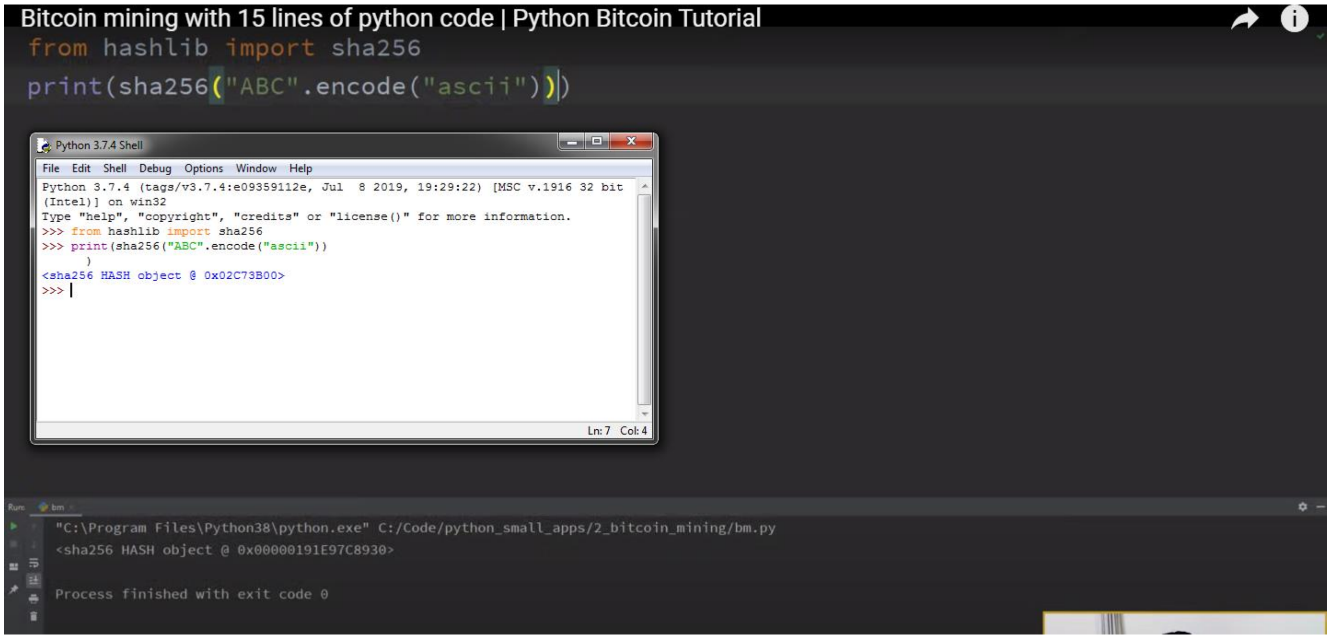

```python
from hashlib import sha256

```

```
"C:\Program Files\Python38\python.exe" C:/Code/python_small_apps/2_bitcoin_mining/bm.py
start mining
Yay! Successfully mined bitcoins with nonce value:1024375
end mining. Mining took: 1.4382002353668213 seconds
0000001189ae1e7a3efb33bb460f7ce4889fc2defc58e4b4545a8c66fc6e411f

Process finished with exit code 0
```



```
from hashlib import sha256
print(sha256("ABC"))
```

```
"C:\Program Files\Python38\python.exe" C:/Code/python_small_apps/2_bitcoin_mining/bm.py
Traceback (most recent call last):
  File "C:/Code/python_small_apps/2_bitcoin_mining/bm.py", line 2, in <module>
    print(sha256("ABC"))
TypeError: Unicode-objects must be encoded before hashing

Process finished with exit code 1
```



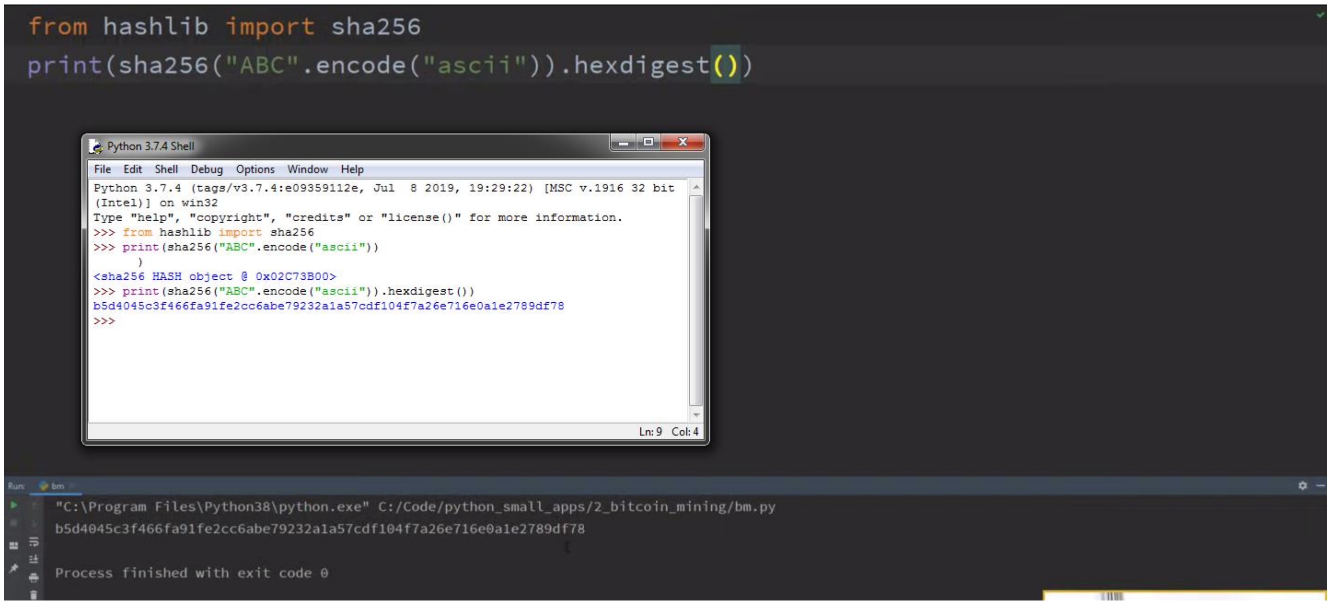
Bitcoin mining with 15 lines of python code | Python Bitcoin Tutorial

```
from hashlib import sha256
print(sha256("ABC".encode("ascii")))
```

```
Python 3.7.4 (tags/v3.7.4:e09359112e, Jul  8 2019, 19:29:22) [MSC v.1916 32 bit
(Intel)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>> from hashlib import sha256
>>> print(sha256("ABC".encode("ascii"))
      )
<sha256 HASH object @ 0x02C73B00>
>>> 
```

```
"C:\Program Files\Python38\python.exe" C:/Code/python_small_apps/2_bitcoin_mining/bm.py
<sha256 HASH object @ 0x00000191E97C8930>

Process finished with exit code 0
```

This will give you the sha256 object that you can get the actual hash out of

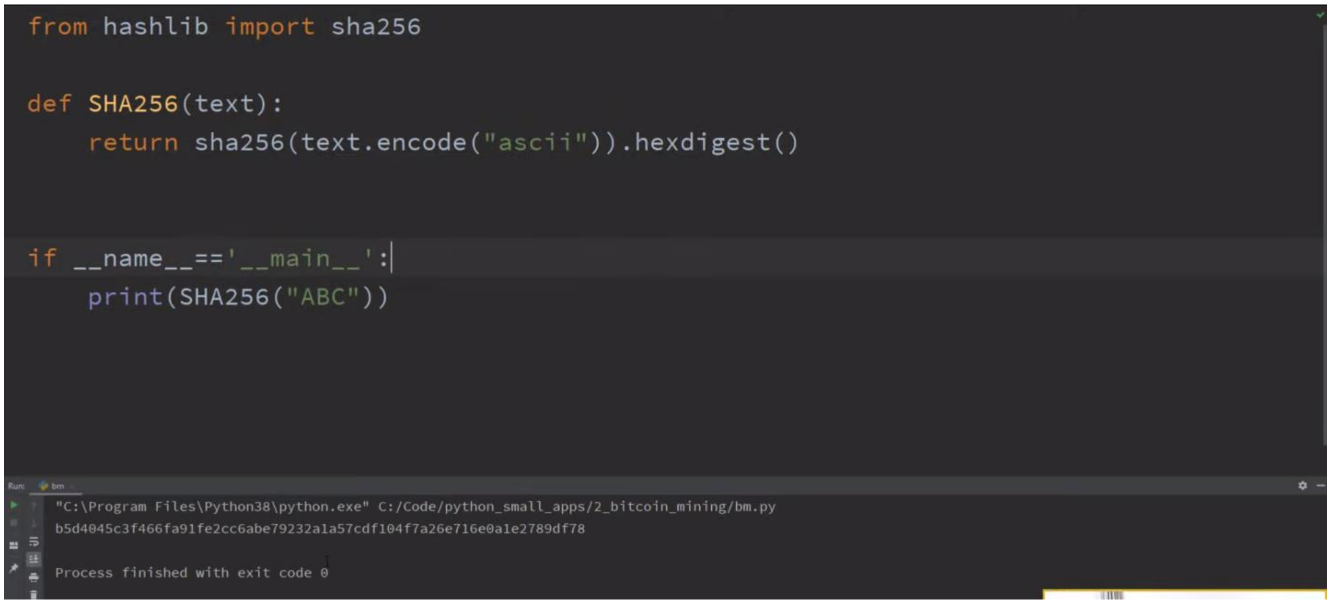


```
from hashlib import sha256
print(sha256("ABC".encode("ascii")).hexdigest())
```

```
Python 3.7.4 (tags/v3.7.4:e09359112e, Jul  8 2019, 19:29:22) [MSC v.1916 32 bit
(Intel)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>> from hashlib import sha256
>>> print(sha256("ABC".encode("ascii"))
      )
<sha256 HASH object @ 0x02C73B00>
>>> print(sha256("ABC".encode("ascii")).hexdigest())
b5d4045c3f466fa91fe2cc6abe79232a1a57cdf104f7a26e716e0a1e2789df78
>>>
```

```
"C:\Program Files\Python38\python.exe" C:/Code/python_small_apps/2_bitcoin_mining/bm.py
b5d4045c3f466fa91fe2cc6abe79232a1a57cdf104f7a26e716e0a1e2789df78

Process finished with exit code 0
```

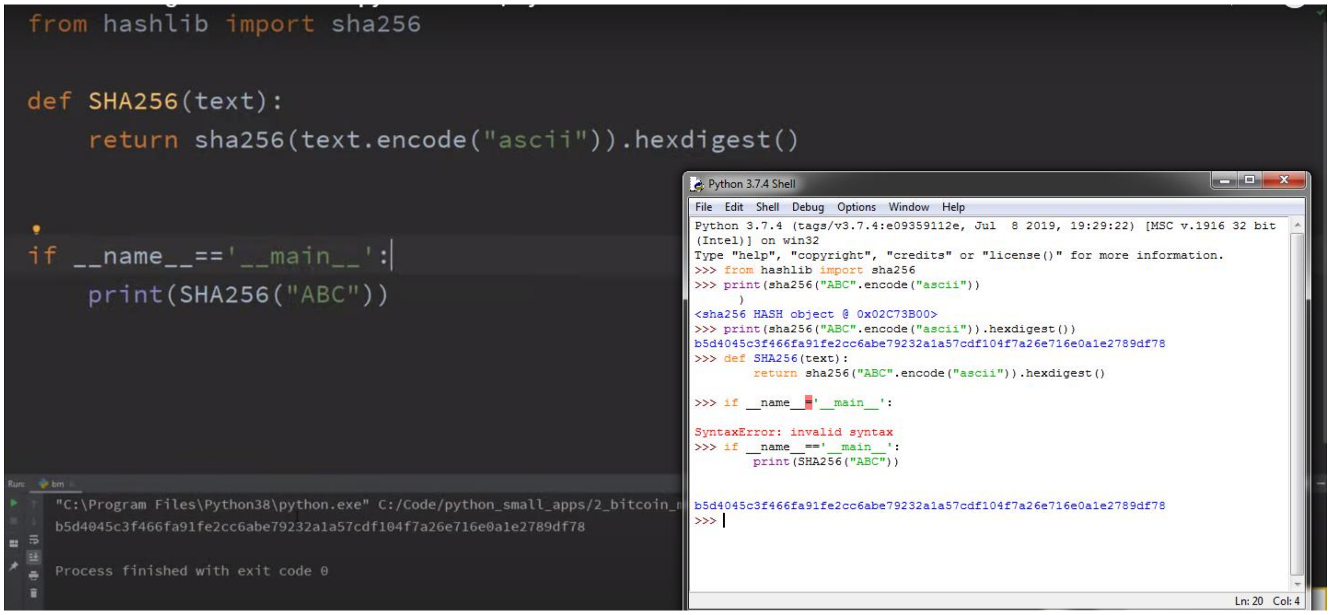
This **sha256("ABC".encode("ascii")).hexdigest())** command will give you the hash value as a 64-bit hexadecimal value which is actually 256-bit value.



```
from hashlib import sha256

def SHA256(text):
    return sha256(text.encode("ascii")).hexdigest()


if __name__=='__main__':
    print(SHA256("ABC"))
```

```
"C:\Program Files\Python38\python.exe" C:/Code/python_small_apps/2_bitcoin_mining/bm.py
b5d4045c3f466fa91fe2cc6abe79232a1a57cdf104f7a26e716e0a1e2789df78

Process finished with exit code 0
```

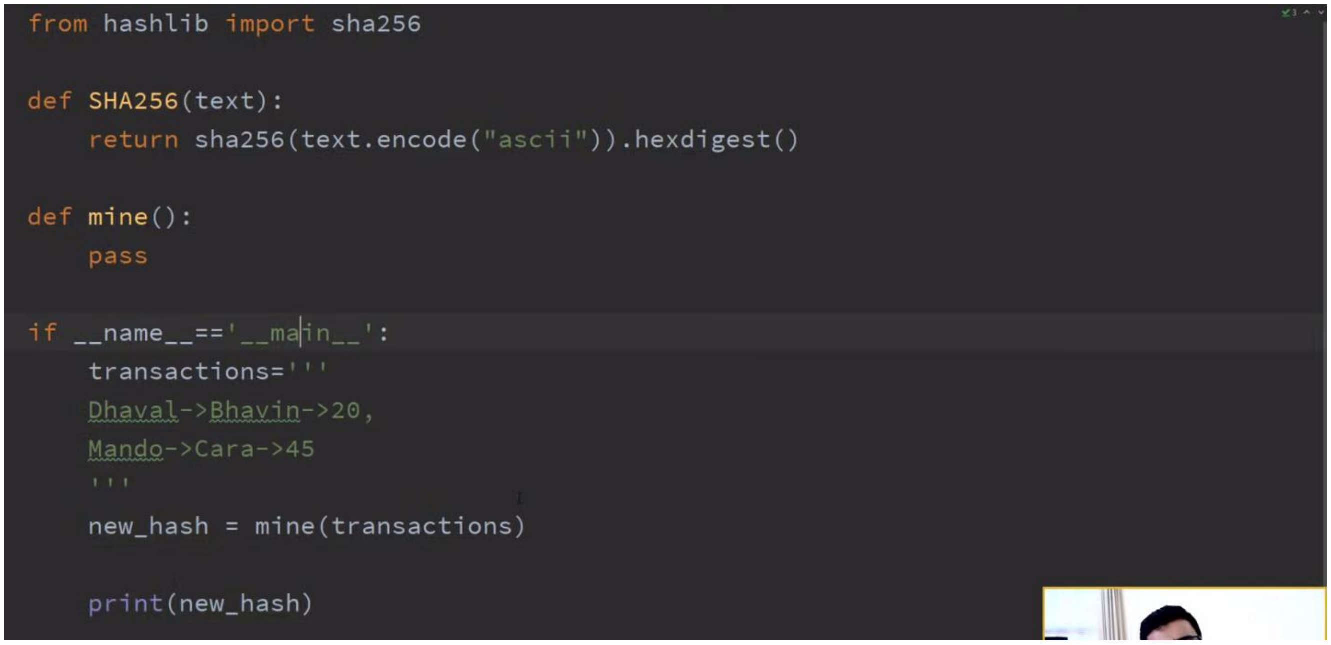
We can create a function to pass in our text value as above.



```python
from hashlib import sha256

def SHA256(text):
    return sha256(text.encode("ascii")).hexdigest()


if __name__=='__main__':
    print(SHA256("ABC"))
```

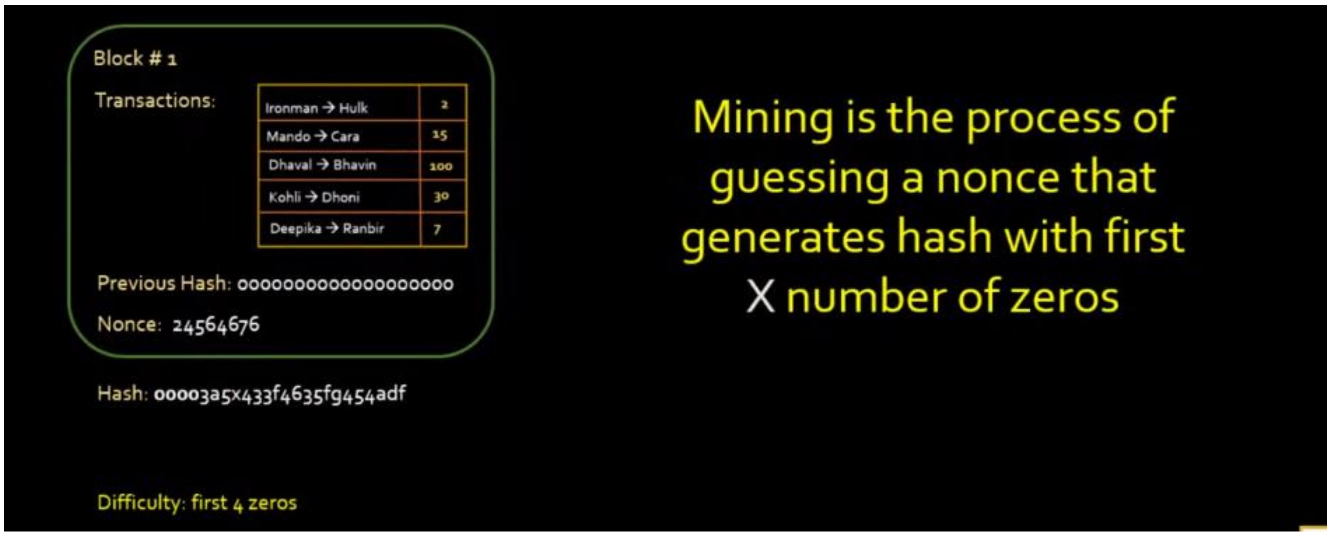
```
"C:\Program Files\Python38\python.exe" C:/Code/python_small_apps/2_bitcoin_m
b5d4045c3f466fa91fe2cc6abe79232a1a57cdf104f7a26e716e0a1e2789df78

Process finished with exit code 0
```

```
Python 3.7.4 (tags/v3.7.4:e09359112e, Jul  8 2019, 19:29:22) [MSC v.1916 32 bit
(Intel)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>> from hashlib import sha256
>>> print(sha256("ABC".encode("ascii"))
      )
<sha256 HASH object @ 0x02C73B00>
>>> print(sha256("ABC".encode("ascii")).hexdigest())
b5d4045c3f466fa91fe2cc6abe79232a1a57cdf104f7a26e716e0a1e2789df78
>>> def SHA256(text):
        return sha256("ABC".encode("ascii")).hexdigest()

>>> if __name__='__main__':

SyntaxError: invalid syntax
>>> if __name__=='__main__':
        print(SHA256("ABC"))


b5d4045c3f466fa91fe2cc6abe79232a1a57cdf104f7a26e716e0a1e2789df78
>>> 
```



```python
from hashlib import sha256

def SHA256(text):
    return sha256(text.encode("ascii")).hexdigest()

def mine():
    pass

if __name__=='__main__':
    transactions='''
    Dhaval->Bhavin->20,
    Mando->Cara->45
    '''
    new_hash = mine(transactions)

    print(new_hash)
```

For bitcoin mining, we need transactions to be passed to our **mine()** function as above



Block # 1

Transactions:

Ironman → Hulk	2
Mando → Cara	15
Dhaval → Bhavin	100
Kohli → Dhoni	30
Deepika → Ranbir	7

Previous Hash: 0000000000000000000

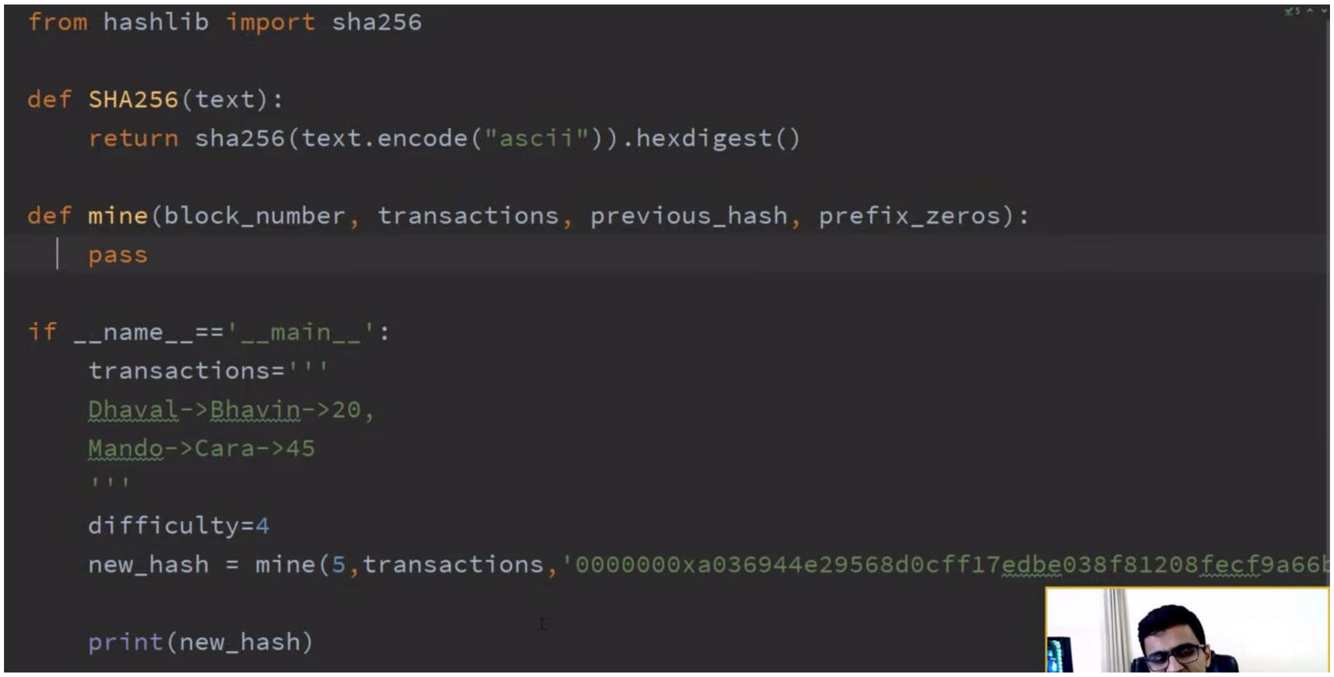
Nonce: 24564676

Hash: 00003a5x433f4635f9454adf

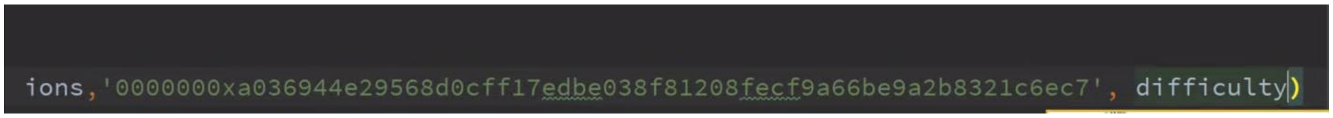
Difficulty: first 4 zeros

Mining is the process of guessing a nonce that generates hash with first X number of zeros

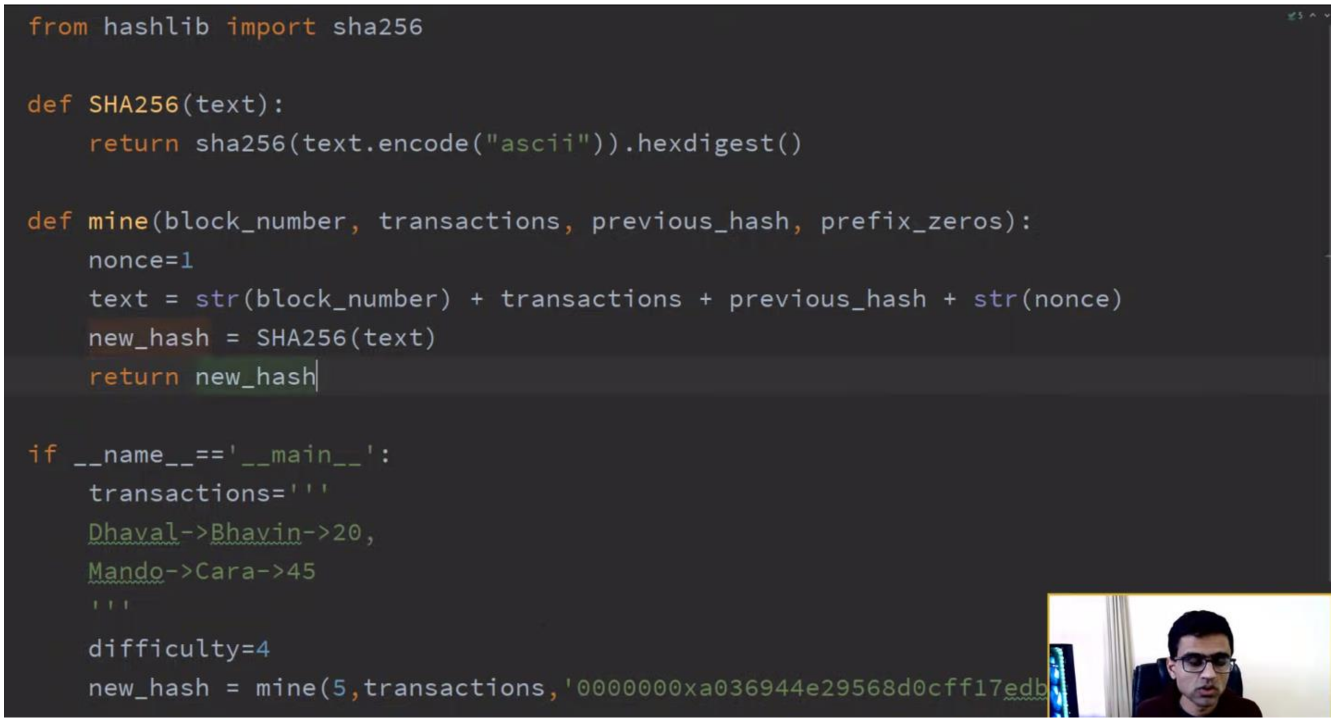
The hash value to be calculated needs all the values above plus our guessed nonce value, we are trying to get a nonce that gives a hash value that has 20 starting zeros on the hash value



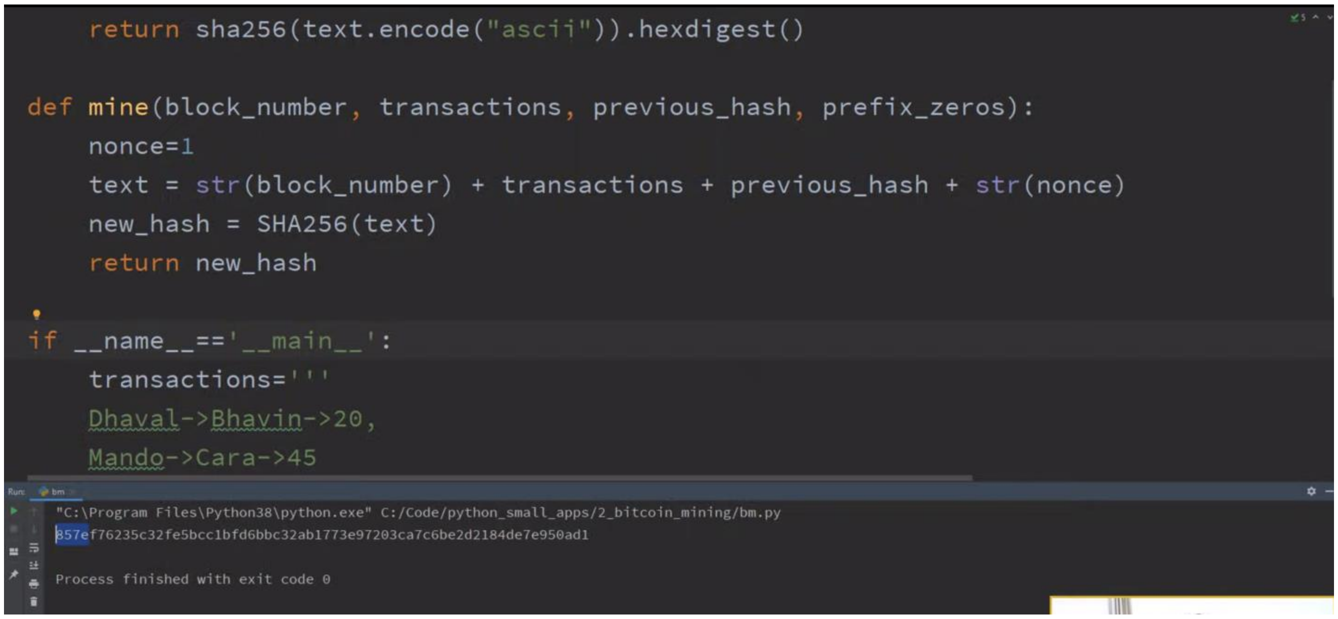
```python
from hashlib import sha256

def SHA256(text):
    return sha256(text.encode("ascii")).hexdigest()

def mine(block_number, transactions, previous_hash, prefix_zeros):
    pass

if __name__=='__main__':
    transactions='''
    Dhaval->Bhavin->20,
    Mando->Cara->45
    '''
    difficulty=4
    new_hash = mine(5,transactions,'0000000xa036944e29568d0cff17edbe038f81208fecf9a66b

    print(new_hash)
```



```python
ions,'0000000xa036944e29568d0cff17edbe038f81208fecf9a66be9a2b8321c6ec7', difficulty)
```



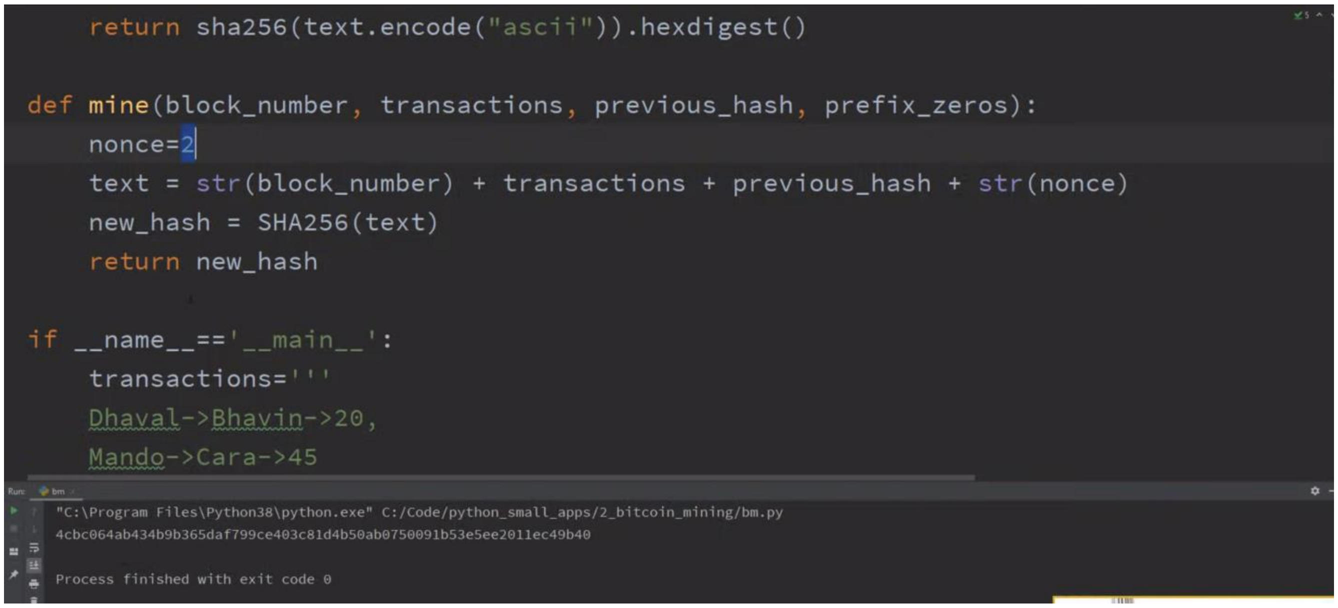
```python
from hashlib import sha256

def SHA256(text):
    return sha256(text.encode("ascii")).hexdigest()

def mine(block_number, transactions, previous_hash, prefix_zeros):
    nonce=1
    text = str(block_number) + transactions + previous_hash + str(nonce)
    new_hash = SHA256(text)
    return new_hash

if __name__=='__main__':
    transactions='''
    Dhaval->Bhavin->20,
    Mando->Cara->45
    '''
    difficulty=4
    new_hash = mine(5,transactions,'0000000xa036944e29568d0cff17edb
```



```python
    return sha256(text.encode("ascii")).hexdigest()

def mine(block_number, transactions, previous_hash, prefix_zeros):
    nonce=1
    text = str(block_number) + transactions + previous_hash + str(nonce)
    new_hash = SHA256(text)
    return new_hash

if __name__=='__main__':
    transactions='''
    Dhaval->Bhavin->20,
    Mando->Cara->45
```

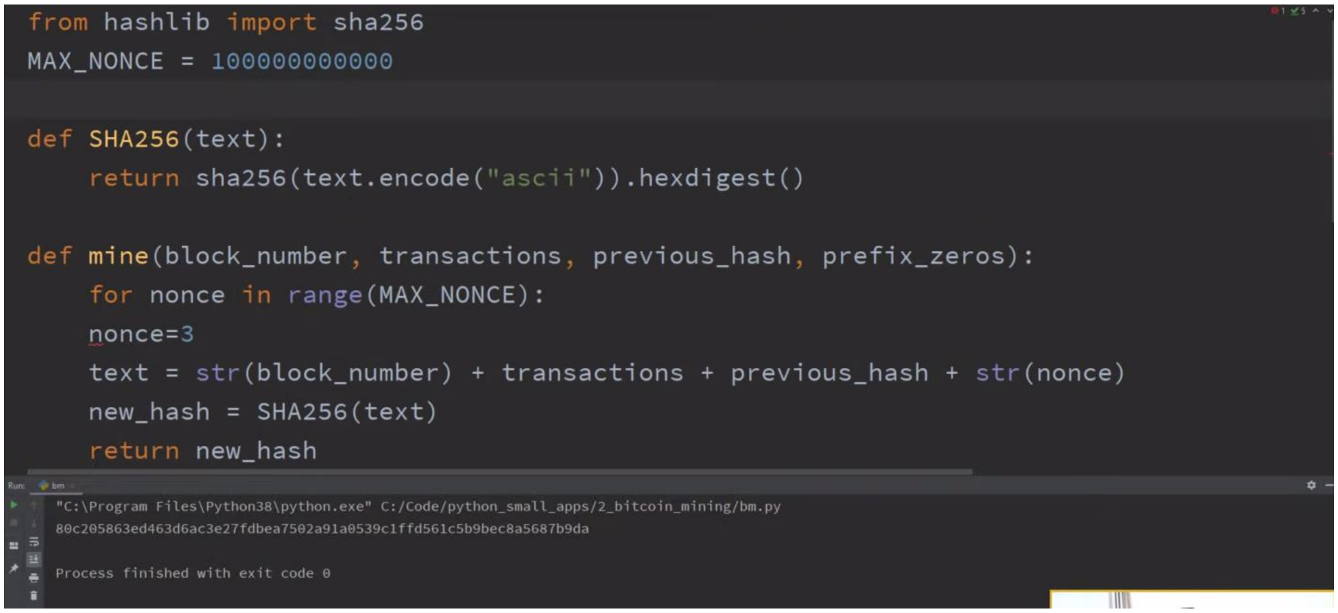
```
"C:\Program Files\Python38\python.exe" C:/Code/python_small_apps/2_bitcoin_mining/bm.py
857ef76235c32fe5bcc1bfd6bbc32ab1773e97203ca7c6be2d2184de7e950ad1

Process finished with exit code 0
```

Our first 4 digits of the hash are not zero for a guessed nonce value of 1. We need to try a different nonce guess



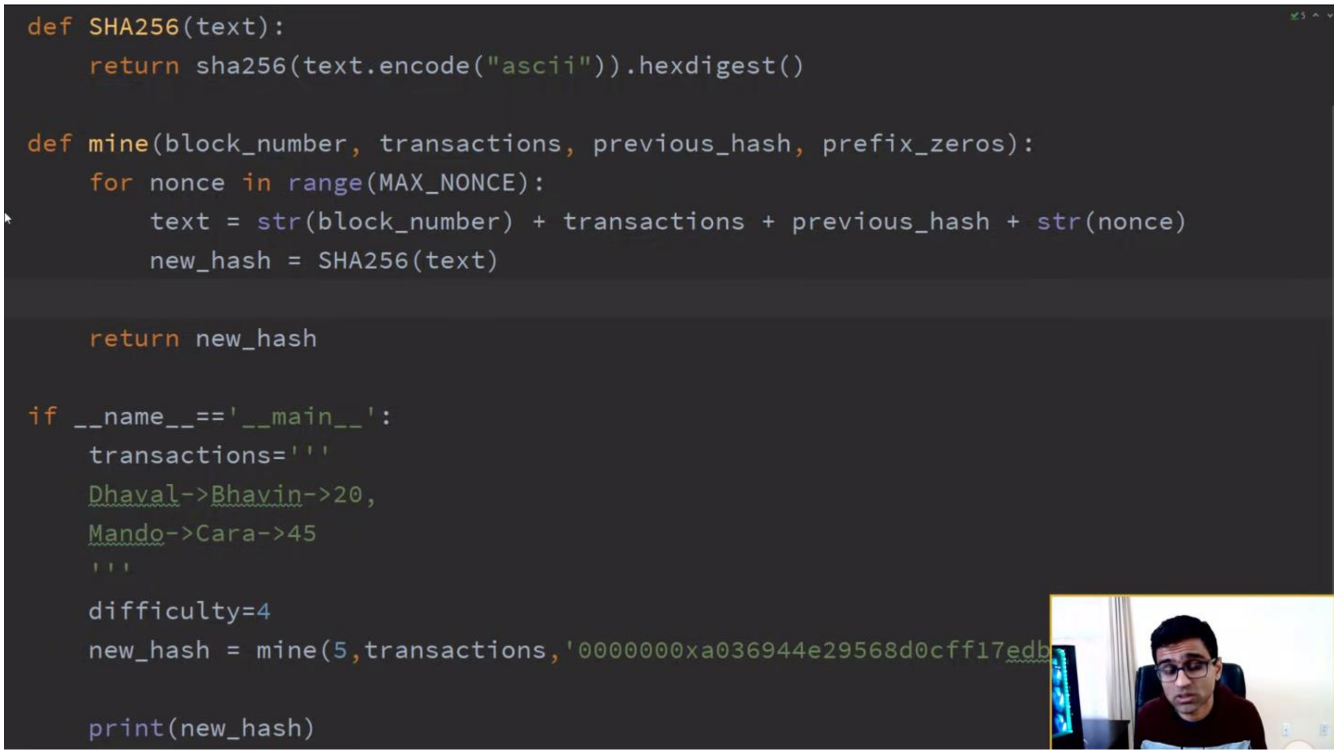
```python
    return sha256(text.encode("ascii")).hexdigest()

def mine(block_number, transactions, previous_hash, prefix_zeros):
    nonce=2
    text = str(block_number) + transactions + previous_hash + str(nonce)
    new_hash = SHA256(text)
    return new_hash

if __name__=='__main__':
    transactions='''
    Dhaval->Bhavin->20,
    Mando->Cara->45
```

```
"C:\Program Files\Python38\python.exe" C:/Code/python_small_apps/2_bitcoin_mining/bm.py
4cbc064ab434b9b365daf799ce403c81d4b50ab0750091b53e5ee2011ec49b40

Process finished with exit code 0
```

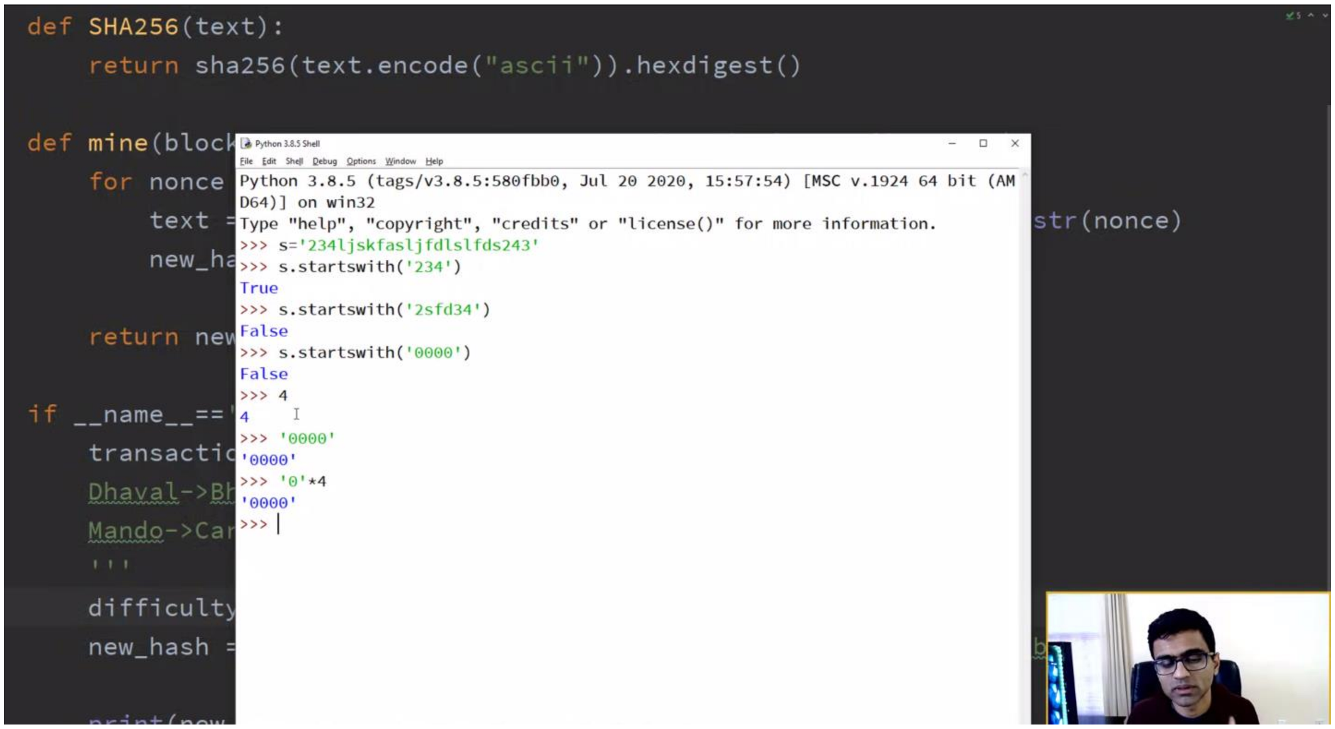


```python
from hashlib import sha256
MAX_NONCE = 100000000000

def SHA256(text):
    return sha256(text.encode("ascii")).hexdigest()

def mine(block_number, transactions, previous_hash, prefix_zeros):
    for nonce in range(MAX_NONCE):
    nonce=3
    text = str(block_number) + transactions + previous_hash + str(nonce)
    new_hash = SHA256(text)
    return new_hash
```

```
"C:\Program Files\Python38\python.exe" C:/Code/python_small_apps/2_bitcoin_mining/bm.py
80c205863ed463d6ac3e27fdbea7502a91a0539c1ffd561c5b9bec8a5687b9da

Process finished with exit code 0
```

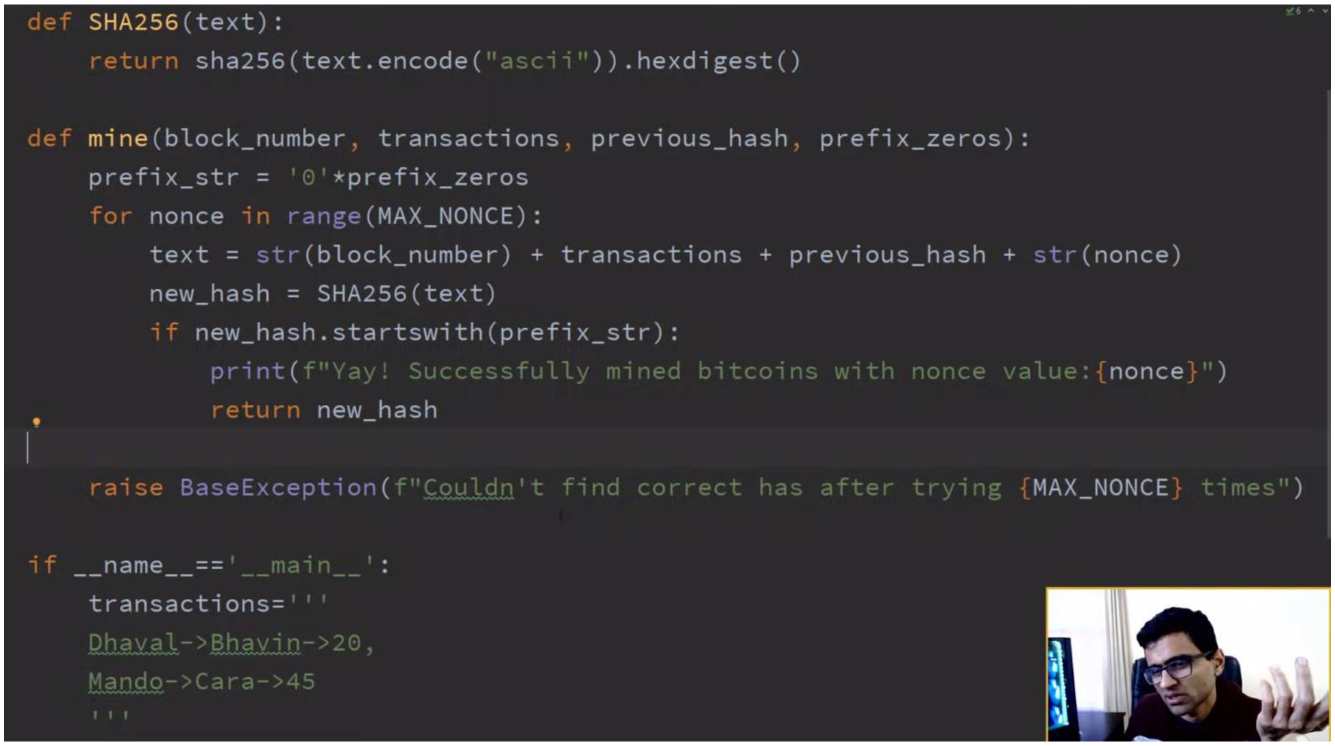


```python
def SHA256(text):
    return sha256(text.encode("ascii")).hexdigest()

def mine(block_number, transactions, previous_hash, prefix_zeros):
    for nonce in range(MAX_NONCE):
        text = str(block_number) + transactions + previous_hash + str(nonce)
        new_hash = SHA256(text)

    return new_hash

if __name__=='__main__':
    transactions='''
    Dhaval->Bhavin->20,
    Mando->Cara->45
    '''
    difficulty=4
    new_hash = mine(5,transactions,'0000000xa036944e29568d0cff17edb

    print(new_hash)
```

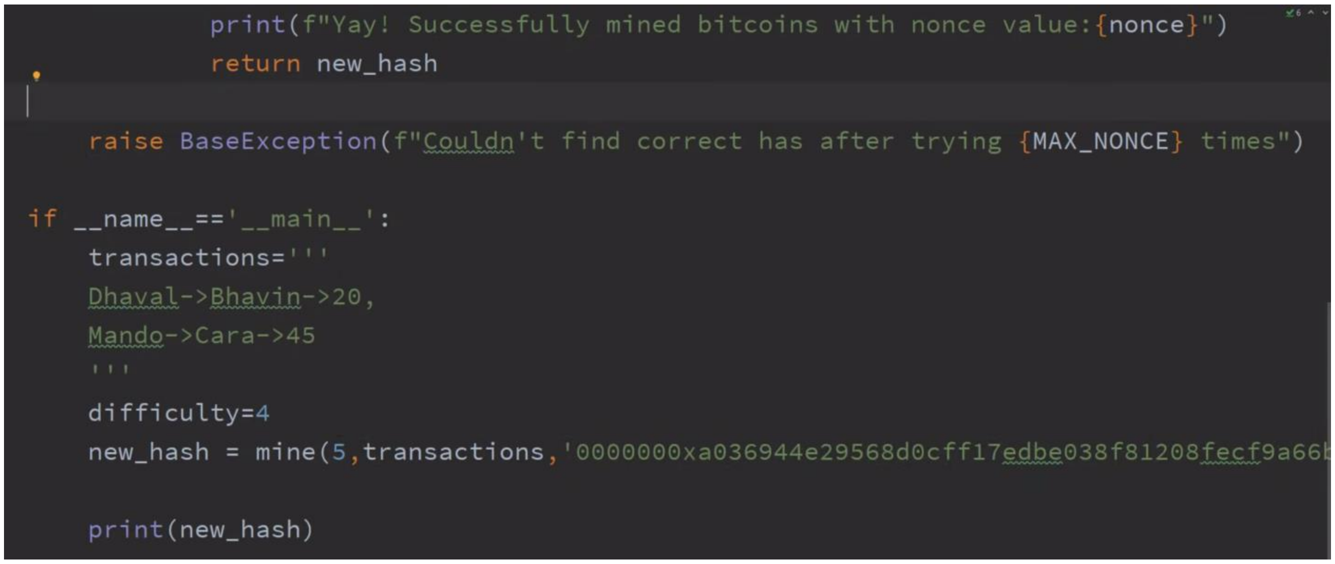


```
def SHA256(text):
    return sha256(text.encode("ascii")).hexdigest()

def mine(block
    for nonce
        text =                                                                          str(nonce)
        new_ha

    return new

if __name__=='
    transactic
    Dhaval->Bh
    Mando->Car
    '''
    difficulty
    new_hash =
```

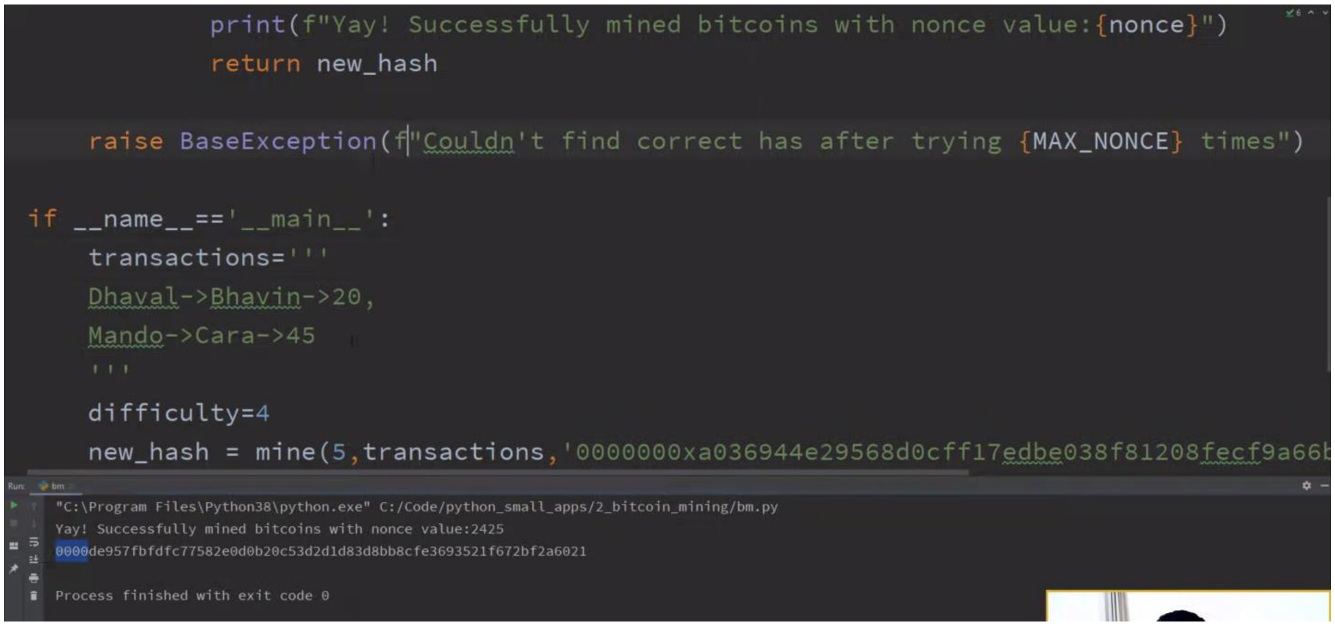
```
Python 3.8.5 (tags/v3.8.5:580fbb0, Jul 20 2020, 15:57:54) [MSC v.1924 64 bit (AM
D64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>> s='234ljskfasljfdlslfds243'
>>> s.startswith('234')
True
>>> s.startswith('2sfd34')
False
>>> s.startswith('0000')
False
>>> 4
4
>>> '0000'
'0000'
>>> '0'*4
'0000'
>>> 
```



```
def SHA256(text):
    return sha256(text.encode("ascii")).hexdigest()

def mine(block_number, transactions, previous_hash, prefix_zeros):
    prefix_str = '0'*prefix_zeros
    for nonce in range(MAX_NONCE):
        text = str(block_number) + transactions + previous_hash + str(nonce)
        new_hash = SHA256(text)
        if new_hash.startswith(prefix_str):
            print(f"Yay! Successfully mined bitcoins with nonce value:{nonce}")
            return new_hash

    raise BaseException(f"Couldn't find correct has after trying {MAX_NONCE} times")

if __name__=='__main__':
    transactions='''
    Dhaval->Bhavin->20,
    Mando->Cara->45
    '''
```



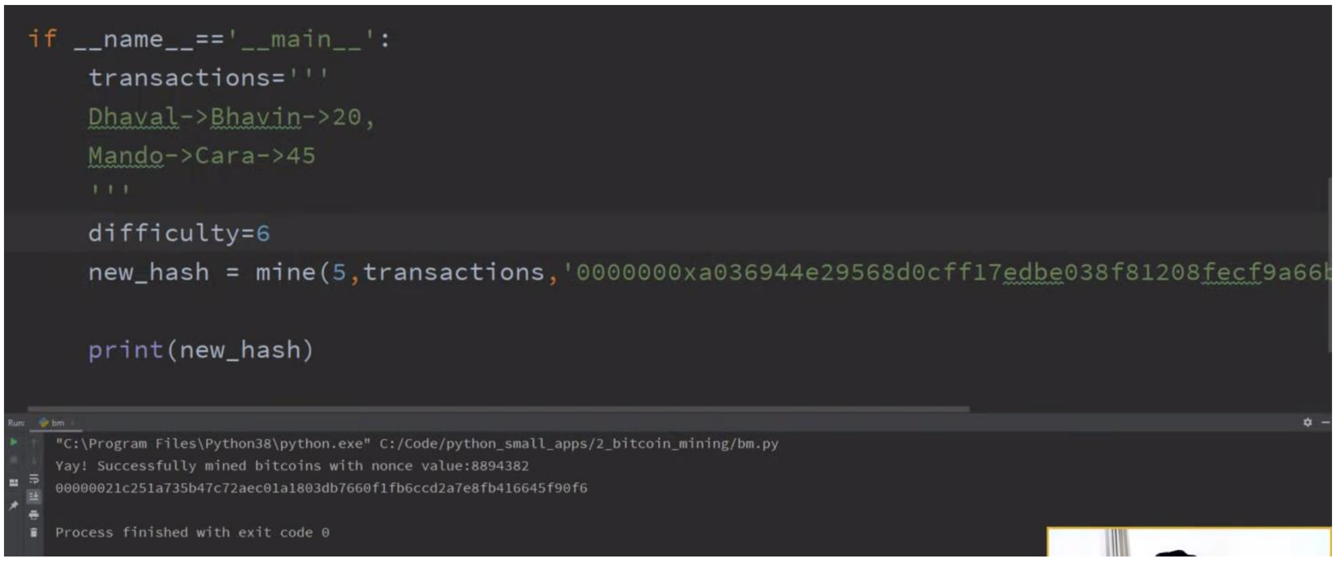
```
            print(f"Yay! Successfully mined bitcoins with nonce value:{nonce}")
            return new_hash

    raise BaseException(f"Couldn't find correct has after trying {MAX_NONCE} times")

if __name__=='__main__':
    transactions='''
    Dhaval->Bhavin->20,
    Mando->Cara->45
    '''
    difficulty=4
    new_hash = mine(5,transactions,'0000000xa036944e29568d0cff17edbe038f81208fecf9a66b

    print(new_hash)
```



```
            print(f"Yay! Successfully mined bitcoins with nonce value:{nonce}")
            return new_hash

    raise BaseException(f"Couldn't find correct has after trying {MAX_NONCE} times")

if __name__=='__main__':
    transactions='''
    Dhaval->Bhavin->20,
    Mando->Cara->45
    '''
    difficulty=4
    new_hash = mine(5,transactions,'0000000xa036944e29568d0cff17edbe038f81208fecf9a66b
```

```
"C:\Program Files\Python38\python.exe" C:/Code/python_small_apps/2_bitcoin_mining/bm.py
Yay! Successfully mined bitcoins with nonce value:2425
0000de957fbfdfc77582e0d0b20c53d2d1d83d8bb8cfe3693521f672bf2a6021

Process finished with exit code 0
```

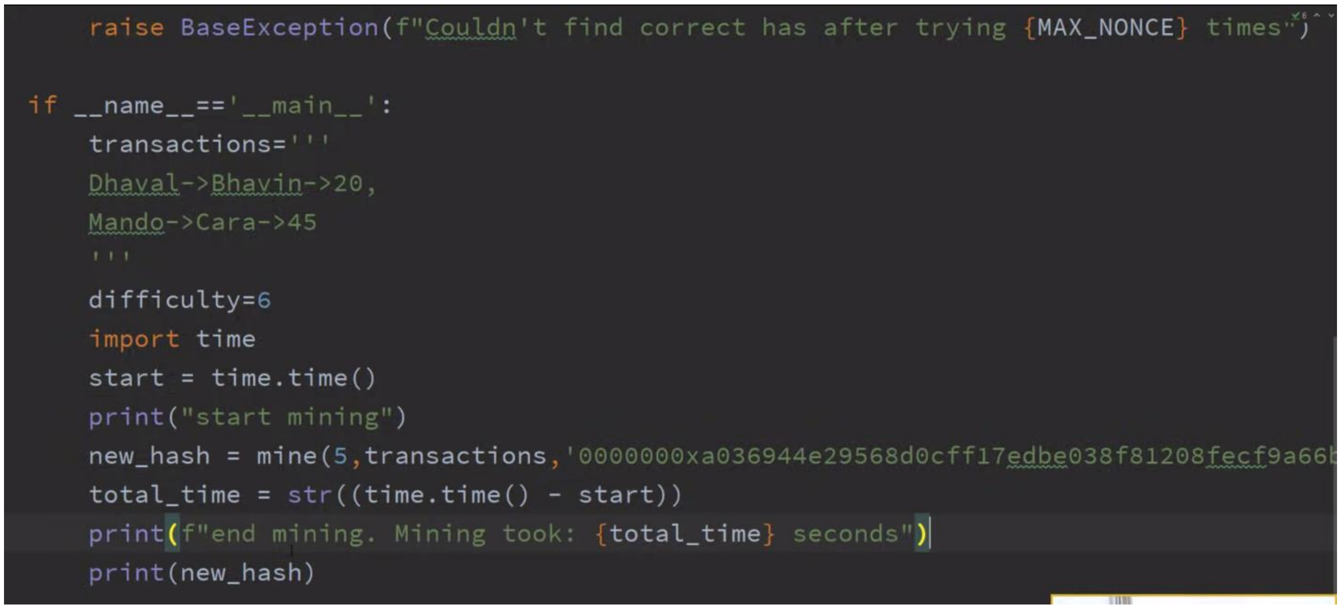
It ran the for loop 2425 times before getting the nonce value for a hash starting with 4 zeros



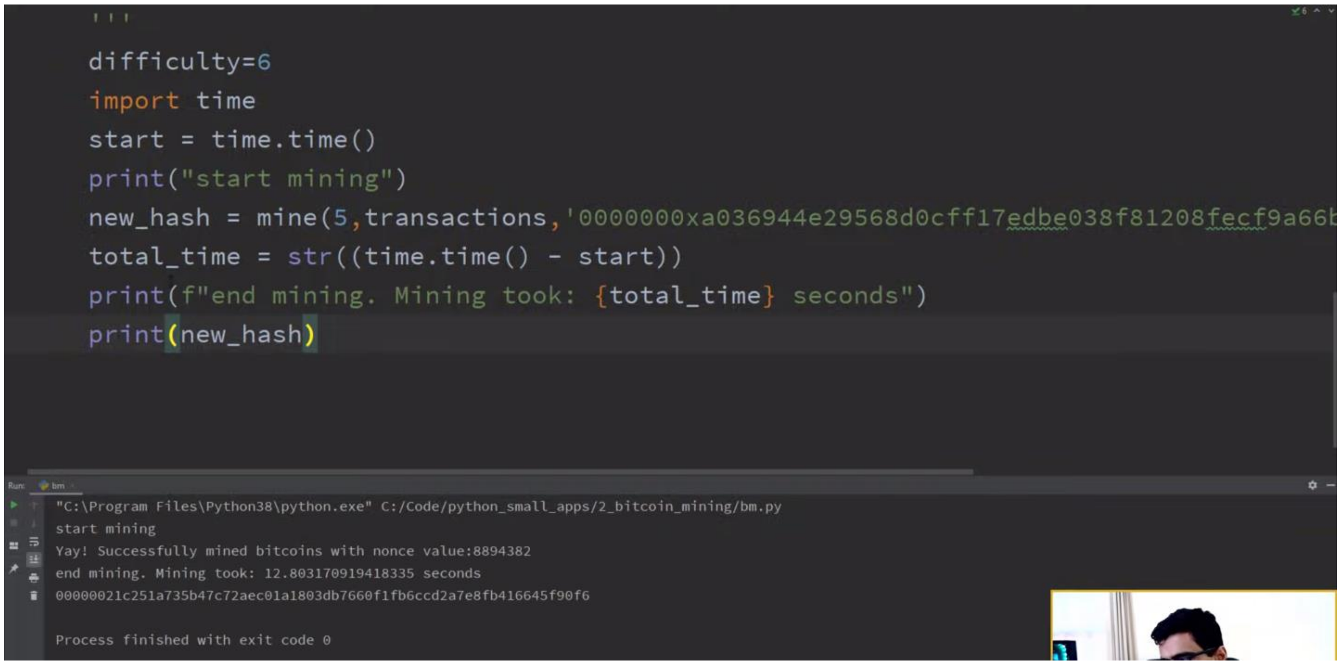
```
if __name__=='__main__':
    transactions='''
    Dhaval->Bhavin->20,
    Mando->Cara->45
    '''
    difficulty=6
    new_hash = mine(5,transactions,'0000000xa036944e29568d0cff17edbe038f81208fecf9a66b

    print(new_hash)
```

```
"C:\Program Files\Python38\python.exe" C:/Code/python_small_apps/2_bitcoin_mining/bm.py
Yay! Successfully mined bitcoins with nonce value:8894382
00000021c251a735b47c72aec01a1803db7660f1fb6ccd2a7e8fb416645f90f6

Process finished with exit code 0
```

It ran the for loop 8894392 times before getting the nonce value for a hash starting with 6 zeros

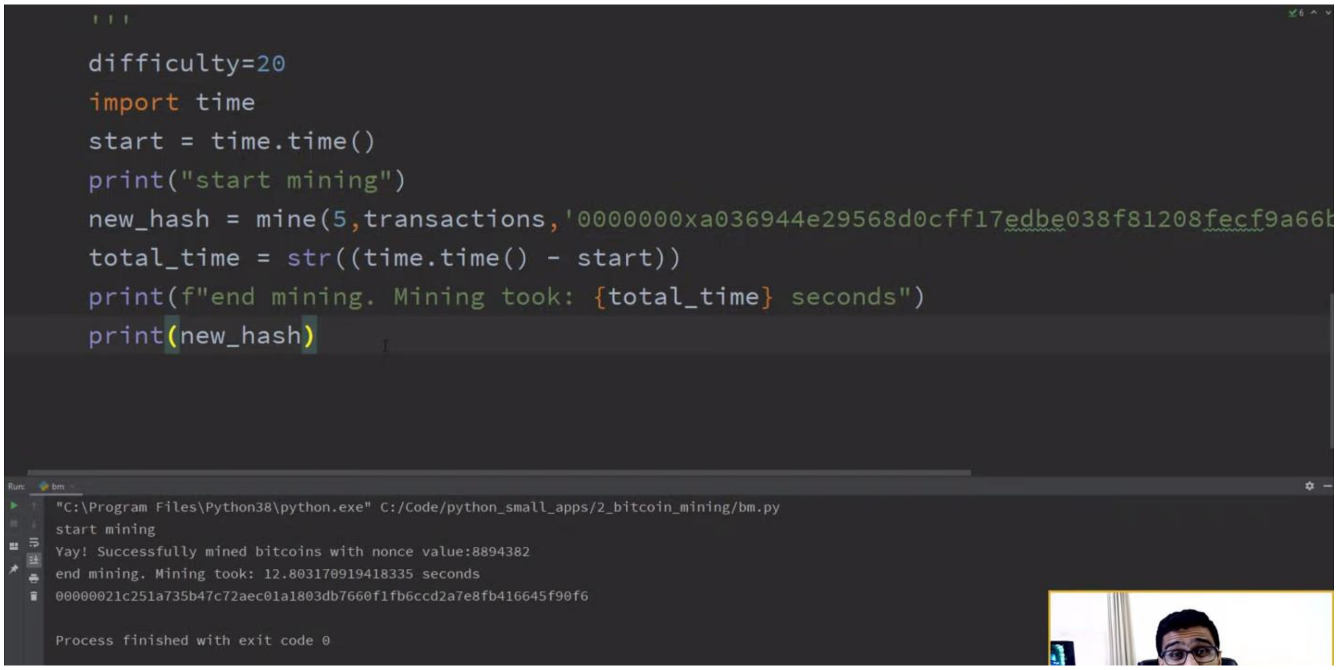


```
    raise BaseException(f"Couldn't find correct has after trying {MAX_NONCE} times")

if __name__=='__main__':
    transactions='''
    Dhaval->Bhavin->20,
    Mando->Cara->45
    '''
    difficulty=6
    import time
    start = time.time()
    print("start mining")
    new_hash = mine(5,transactions,'0000000xa036944e29568d0cff17edbe038f81208fecf9a66b
    total_time = str((time.time() - start))
    print(f"end mining. Mining took: {total_time} seconds")
    print(new_hash)
```



```
    '''
    difficulty=6
    import time
    start = time.time()
    print("start mining")
    new_hash = mine(5,transactions,'0000000xa036944e29568d0cff17edbe038f81208fecf9a66b
    total_time = str((time.time() - start))
    print(f"end mining. Mining took: {total_time} seconds")
    print(new_hash)
```

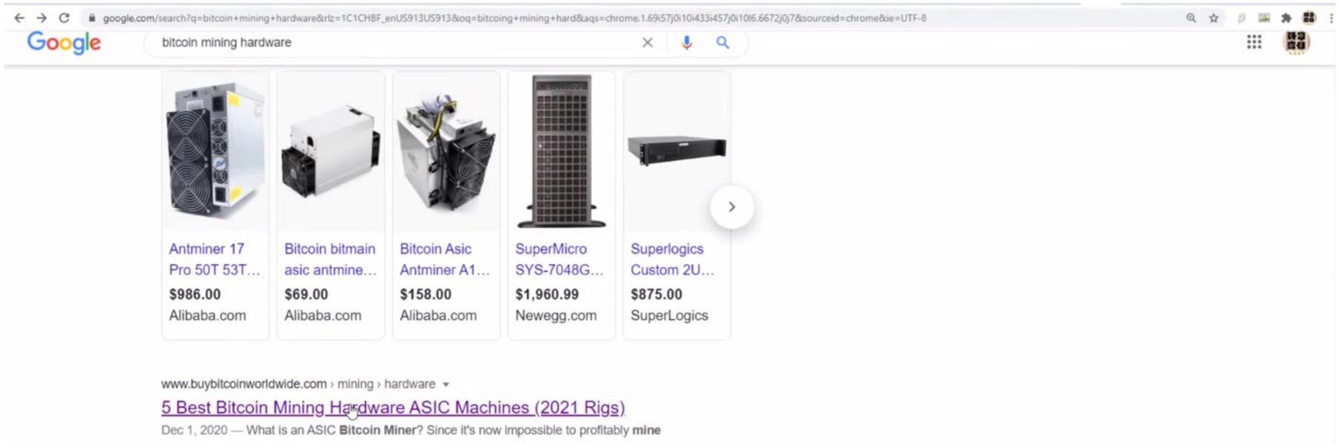
```
"C:\Program Files\Python38\python.exe" C:/Code/python_small_apps/2_bitcoin_mining/bm.py
start mining
Yay! Successfully mined bitcoins with nonce value:8894382
end mining. Mining took: 12.803170919418335 seconds
00000021c251a735b47c72aec01a1803db7660f1fb6ccd2a7e8fb416645f90f6

Process finished with exit code 0
```



```
'''
difficulty=20
import time
start = time.time()
print("start mining")
new_hash = mine(5,transactions,'0000000xa036944e29568d0cff17edbe038f81208fecf9a66b
total_time = str((time.time() - start))
print(f"end mining. Mining took: {total_time} seconds")
print(new_hash)
```

```
"C:\Program Files\Python38\python.exe" C:/Code/python_small_apps/2_bitcoin_mining/bm.py
start mining
Yay! Successfully mined bitcoins with nonce value:8894382
end mining. Mining took: 12.803170919418335 seconds
00000021c251a735b47c72aec01a1803db7660f1fb6ccd2a7e8fb416645f90f6

Process finished with exit code 0
```

The current difficulty level is 20 leading zeros and this requires specialized hardware to make the guess faster

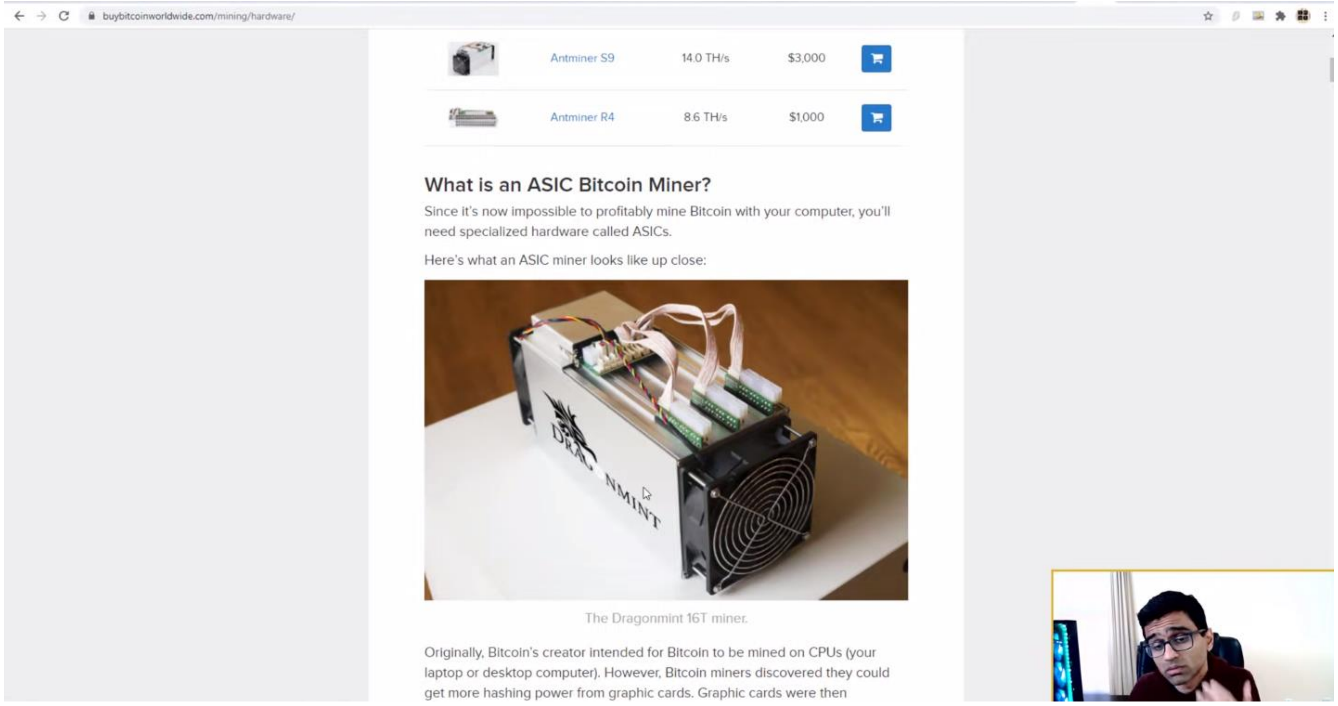


Antminer 17 Pro 50T 53T…	Bitcoin bitmain asic antmine…	Bitcoin Asic Antminer A1…	SuperMicro SYS-7048G…	Superlogics Custom 2U…
$986.00	$69.00	$158.00	$1,960.99	$875.00
Alibaba.com	Alibaba.com	Alibaba.com	Newegg.com	SuperLogics

www.buybitcoinworldwide.com › mining › hardware

5 Best Bitcoin Mining Hardware ASIC Machines (2021 Rigs)

Dec 1, 2020 — What is an ASIC **Bitcoin Miner**? Since it's now impossible to profitably **mine**



Antminer S9	14.0 TH/s	$3,000
Antminer R4	8.6 TH/s	$1,000

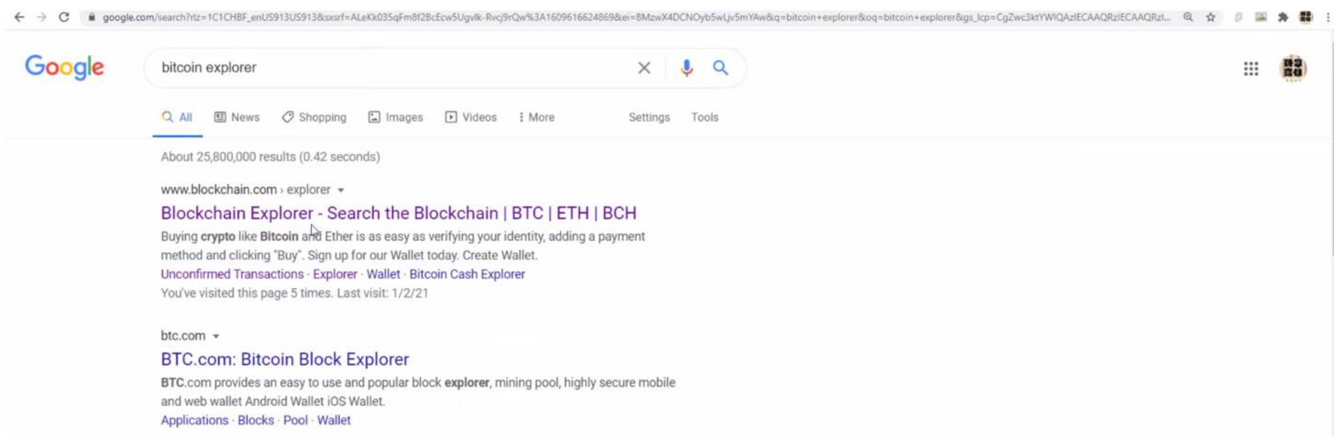
What is an ASIC Bitcoin Miner?

Since it's now impossible to profitably mine Bitcoin with your computer, you'll need specialized hardware called ASICs.

Here's what an ASIC miner looks like up close:

The Dragonmint 16T miner.

Originally, Bitcoin's creator intended for Bitcoin to be mined on CPUs (your laptop or desktop computer). However, Bitcoin miners discovered they could get more hashing power from graphic cards. Graphic cards were then



google.com/search?rlz=1C1CHBF_enUS913US913&sxsrf=ALeKk035qFm8f2BcEcw5Ugvlk-Rvcj9rQw%3A1609616624869&ei=8MzwX4DCNOyb5wLjv5mYAw&q=bitcoin+explorer&oq=bitcoin+explorer&gs_lcp=CgZwc3kiYWIQAzIECAAQRzIECAAQRzI...
Google
bitcoin explorer
All
News
Shopping
Images
Videos
More
Settings
Tools

About 25,800,000 results (0.42 seconds)

www.blockchain.com › explorer

Blockchain Explorer - Search the Blockchain | BTC | ETH | BCH

Buying **crypto** like **Bitcoin** and Ether is as easy as verifying your identity, adding a payment method and clicking "Buy". Sign up for our Wallet today. Create Wallet.

Unconfirmed Transactions · Explorer · Wallet · Bitcoin Cash Explorer

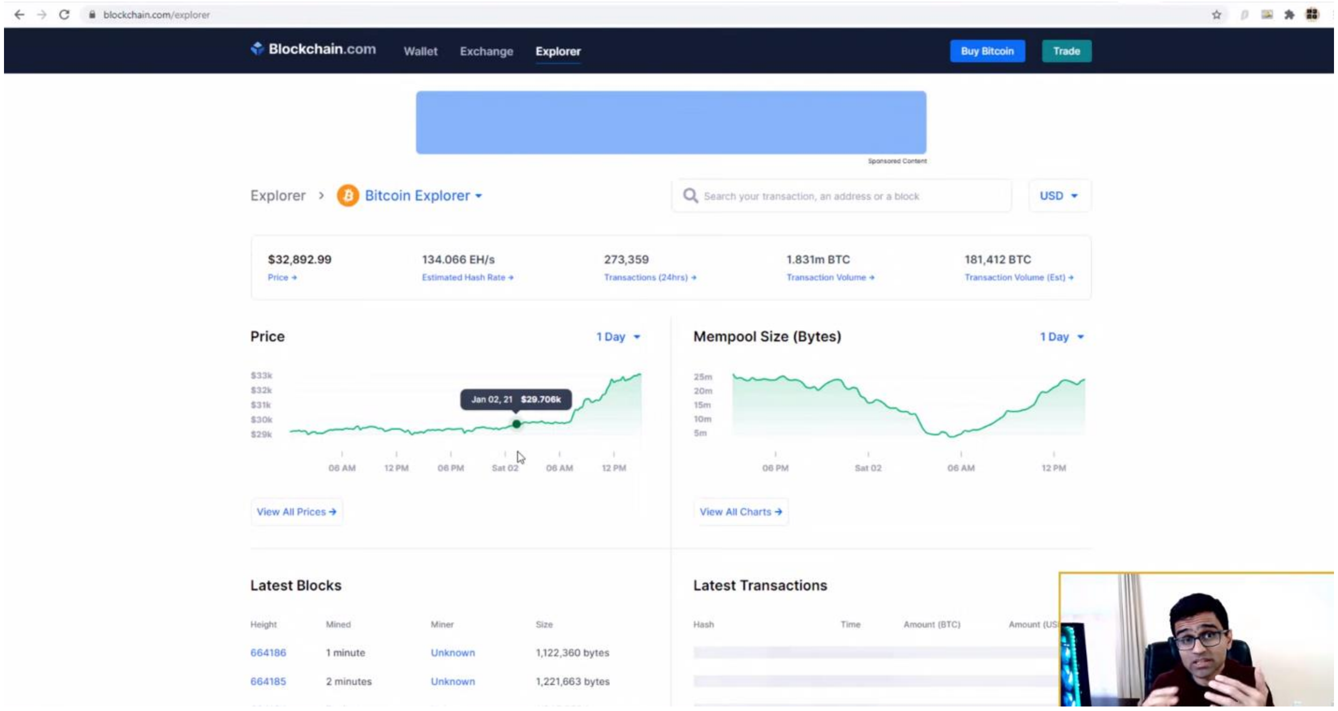
You've visited this page 5 times. Last visit: 1/2/21

btc.com

BTC.com: Bitcoin Block Explorer

BTC.com provides an easy to use and popular block **explorer**, mining pool, highly secure mobile and web wallet Android Wallet iOS Wallet.

Applications · Blocks · Pool · Wallet



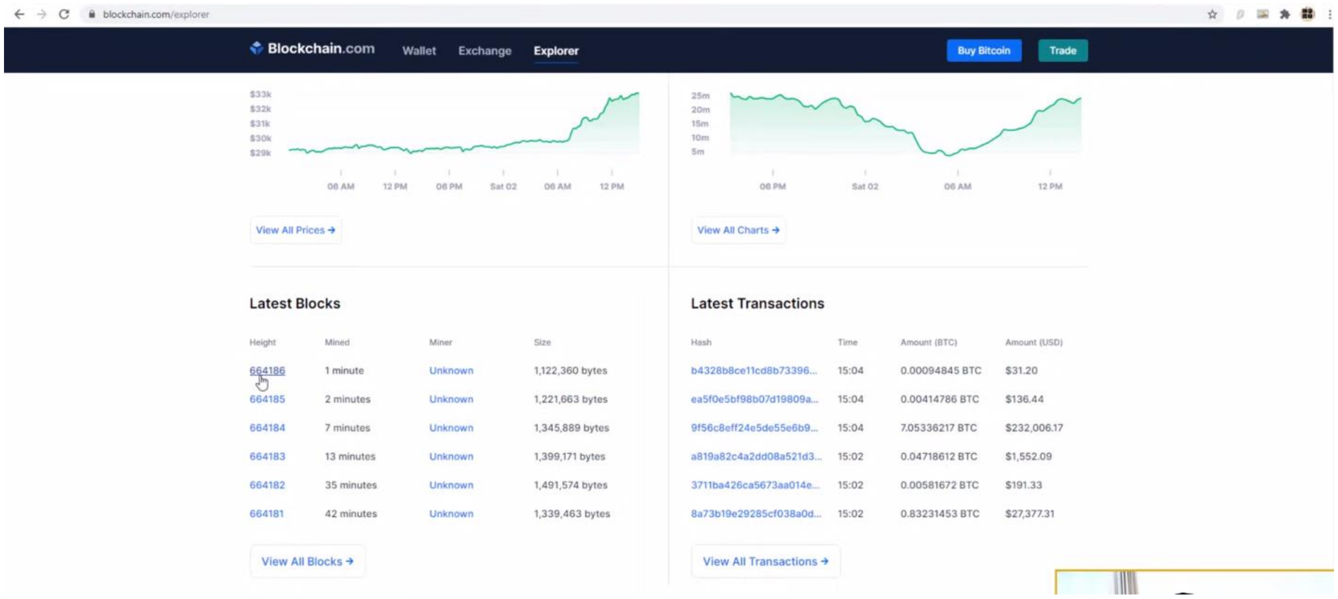
$32,892.99	134.066 EH/s	273,359	1.831m BTC	181,412 BTC
Price	Estimated Hash Rate	Transactions (24hrs)	Transaction Volume	Transaction Volume (Est)

Latest Blocks

Height	Mined	Miner	Size
664186	1 minute	Unknown	1,122,360 bytes
664185	2 minutes	Unknown	1,221,663 bytes

Latest Transactions

Hash	Time	Amount (BTC)	Amount (USD)



Latest Blocks

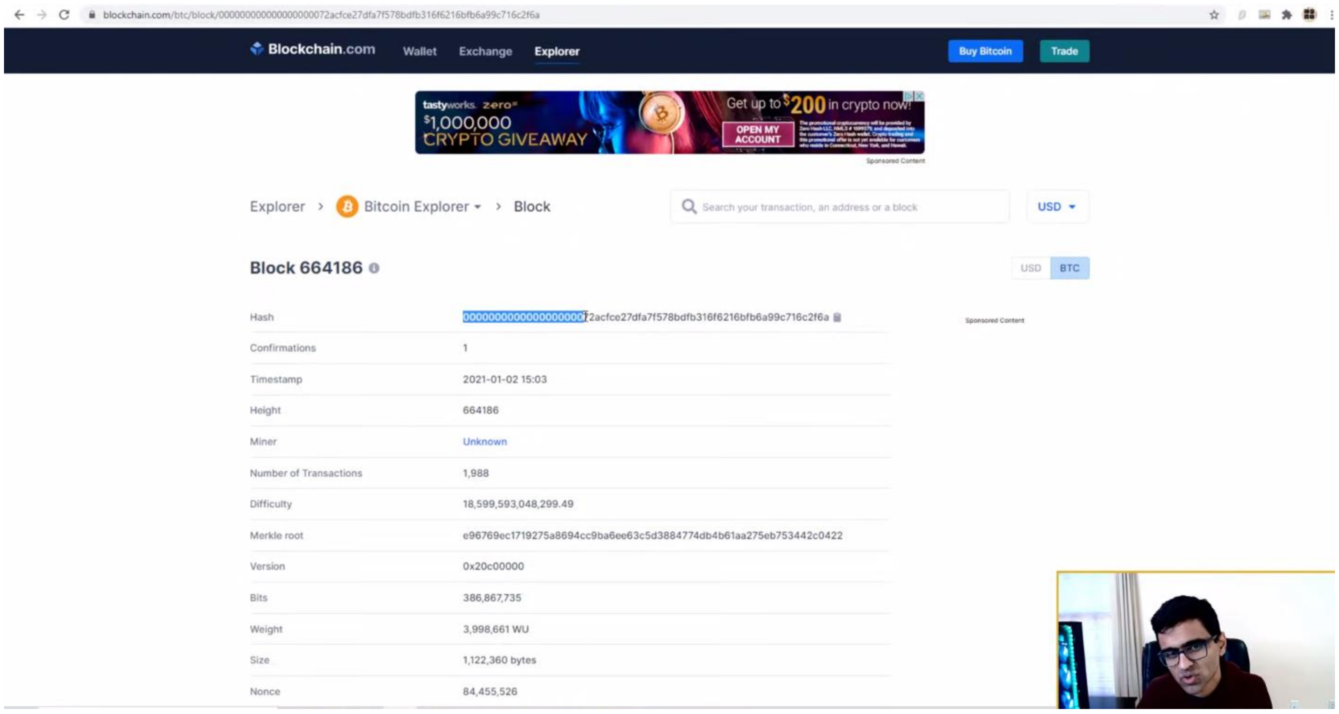
Height	Mined	Miner	Size
664186	1 minute	Unknown	1,122,360 bytes
664185	2 minutes	Unknown	1,221,663 bytes
664184	7 minutes	Unknown	1,345,889 bytes
664183	13 minutes	Unknown	1,399,171 bytes
664182	35 minutes	Unknown	1,491,574 bytes
664181	42 minutes	Unknown	1,339,463 bytes

View All Blocks →

Latest Transactions

Hash	Time	Amount (BTC)	Amount (USD)
b4328b8ce11cd8b73396...	15:04	0.00094845 BTC	$31.20
ea5f0e5bf98b07d19809a...	15:04	0.00414786 BTC	$136.44
9f56c8eff24e5de55e6b9...	15:04	7.05336217 BTC	$232,006.17
a819a82c4a2dd08a521d3...	15:02	0.04718612 BTC	$1,552.09
3711ba426ca5673aa014e...	15:02	0.00581672 BTC	$191.33
8a73b19e29285cf038a0d...	15:02	0.83231453 BTC	$27,377.31

View All Transactions →

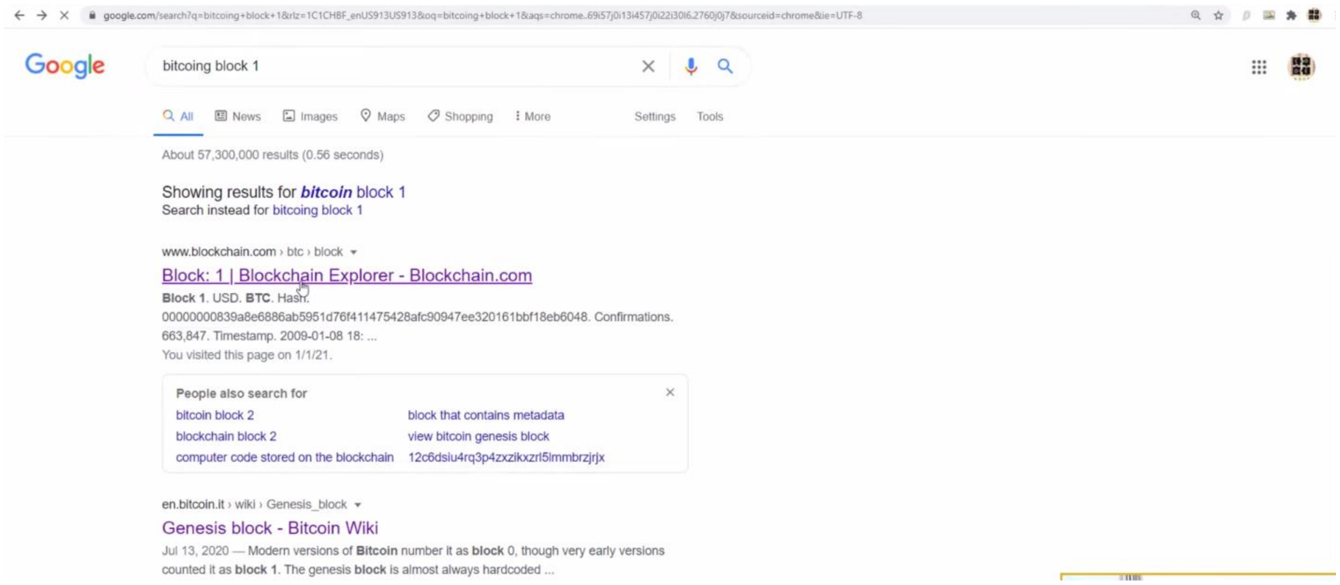

blockchain.com/btc/block/000000000000000000072acfce27dfa7f578bdfb316f6216bfb6a99c716c2f6a

Blockchain.com
Wallet
Exchange
Explorer
Buy Bitcoin
Trade
tastyworks. zero
$1,000,000 CRYPTO GIVEAWAY
Get up to $200 in crypto now!
OPEN MY ACCOUNT
Sponsored Content
Explorer › Bitcoin Explorer › Block
Search your transaction, an address or a block
USD
BTC

Block 664186

Hash	000000000000000000072acfce27dfa7f578bdfb316f6216bfb6a99c716c2f6a
Confirmations	1
Timestamp	2021-01-02 15:03
Height	664186
Miner	Unknown
Number of Transactions	1,988
Difficulty	18,599,593,048,299.49
Merkle root	e96769ec1719275a8694cc9ba6ee63c5d3884774db4b61aa275eb753442c0422
Version	0x20c00000
Bits	386,867,735
Weight	3,998,661 WU
Size	1,122,360 bytes
Nonce	84,455,526

The current bitcoin difficulty level is 20 leading zeros


google.com/search?q=bitcoing+block+1&rlz=1C1CHBF_enUS913US913&oq=bitcoing+block+1&aqs=chrome..69i57j0i13i457j0i22i30l6.2760j0j7&sourceid=chrome&ie=UTF-8

About 57,300,000 results (0.56 seconds)

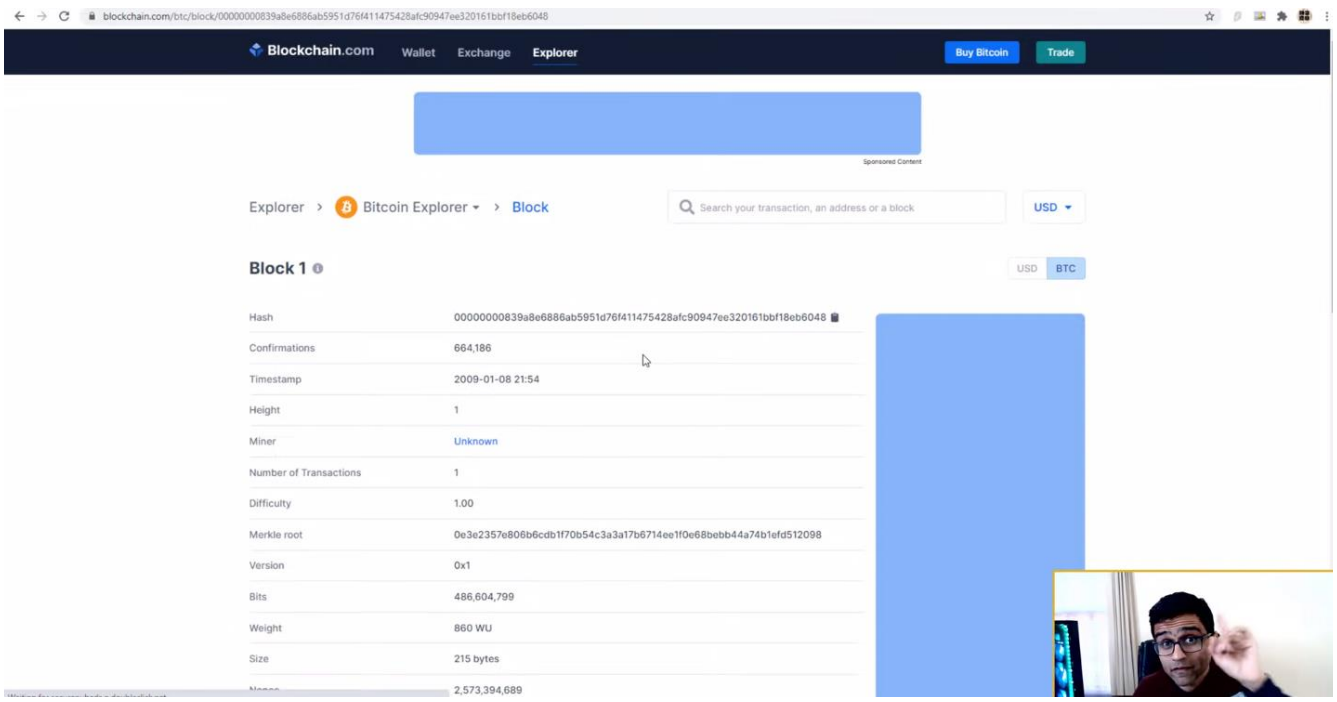
Showing results for ***bitcoin*** block 1
Search instead for bitcoing block 1

www.blockchain.com › btc › block
Block: 1 | Blockchain Explorer - Blockchain.com
Block 1. USD. **BTC**. Hash. 00000000839a8e6886ab5951d76f411475428afc90947ee320161bbf18eb6048. Confirmations. 663,847. Timestamp. 2009-01-08 18: ...
You visited this page on 1/1/21.

People also search for

bitcoin block 2	block that contains metadata
blockchain block 2	view bitcoin genesis block
computer code stored on the blockchain	12c6dsiu4rq3p4zxzlkxzrl5lmmbrzjrjx

en.bitcoin.it › wiki › Genesis_block
Genesis block - Bitcoin Wiki
Jul 13, 2020 — Modern versions of **Bitcoin** number it as **block** 0, though very early versions counted it as **block 1**. The genesis **block** is almost always hardcoded ...

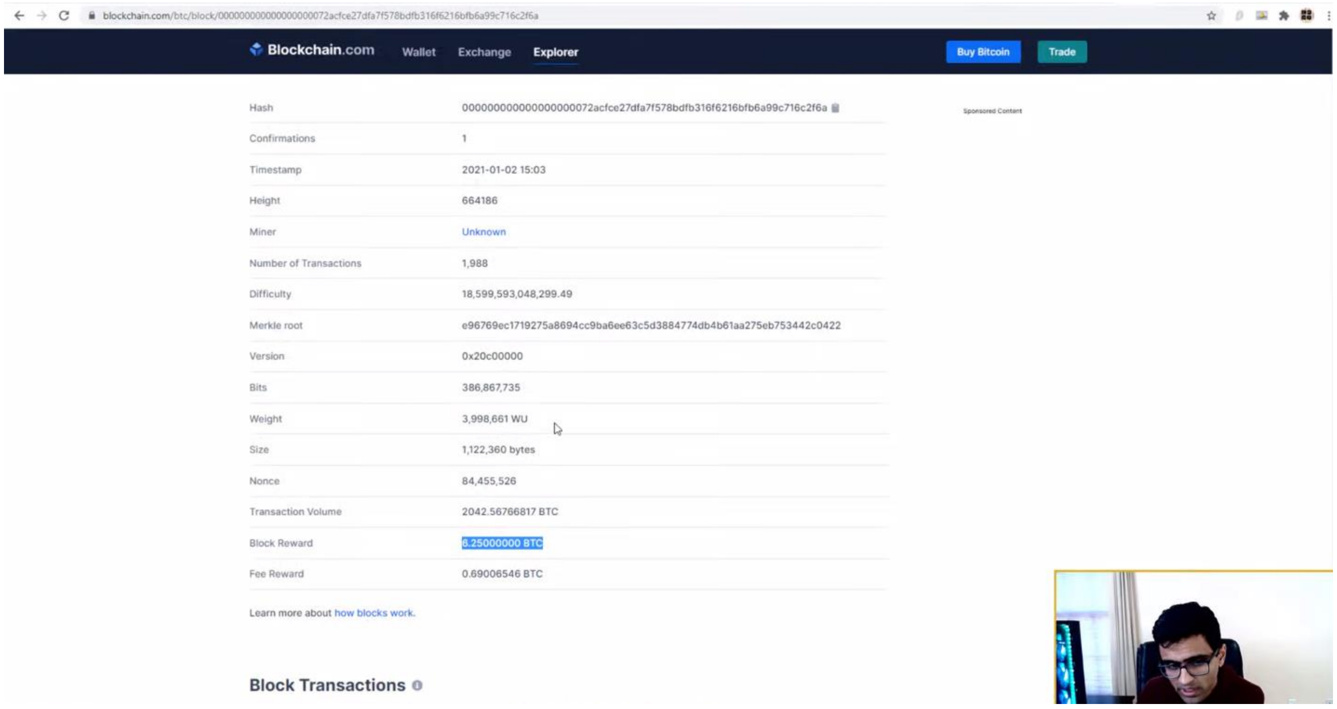

blockchain.com/btc/block/00000000839a8e6886ab5951d76f411475428afc90947ee320161bbf18eb6048

Blockchain.com
Wallet
Exchange
Explorer
Buy Bitcoin
Trade
Sponsored Content
Explorer › Bitcoin Explorer › Block
Search your transaction, an address or a block
USD
USD
BTC

Block 1

Hash	00000000839a8e6886ab5951d76f411475428afc90947ee320161bbf18eb6048
Confirmations	664,186
Timestamp	2009-01-08 21:54
Height	1
Miner	Unknown
Number of Transactions	1
Difficulty	1.00
Merkle root	0e3e2357e806b6cdb1f70b54c3a3a17b6714ee1f0e68bebb44a74b1efd512098
Version	0x1
Bits	486,604,799
Weight	860 WU
Size	215 bytes
Nonce	2,573,394,689

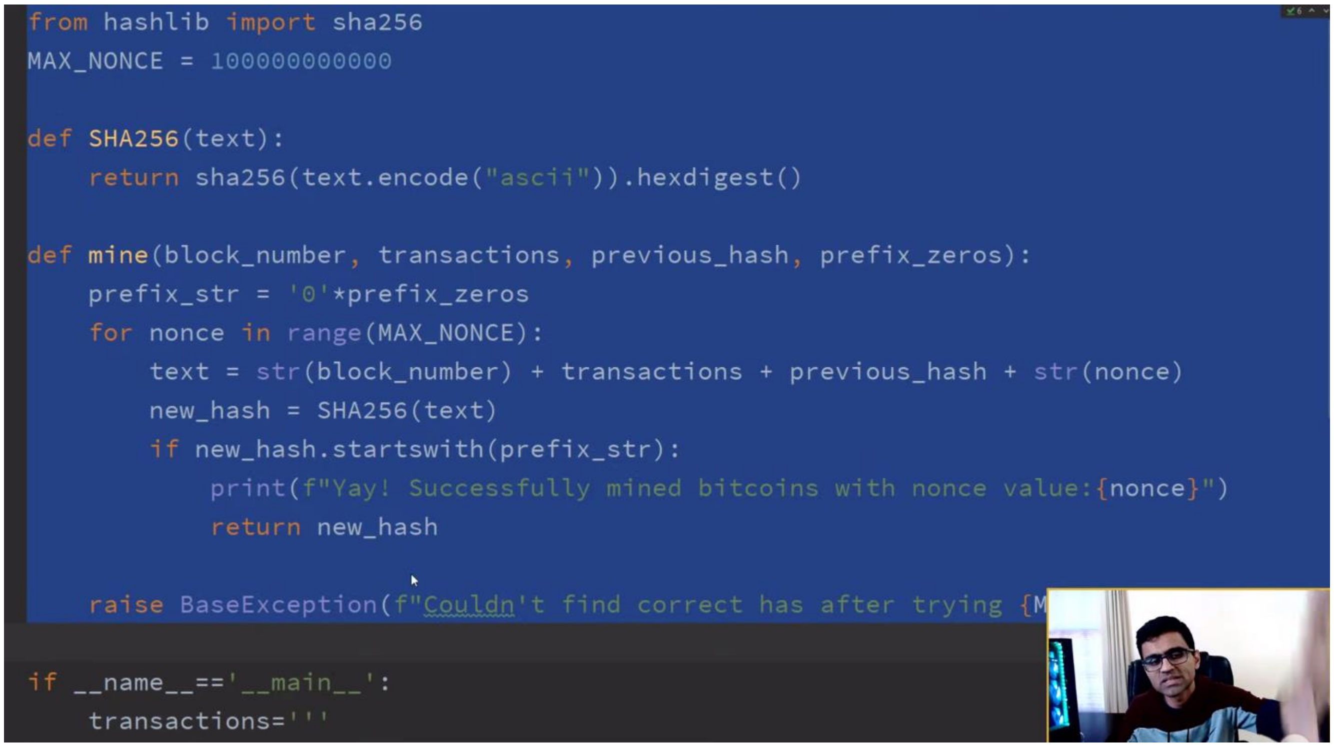
This is the first block even created by Satoshi with 8 leading zeros


blockchain.com/btc/block/0000000000000000000072acfce27dfa7f578bdfb316f6216bfb6a99c716c2f6a

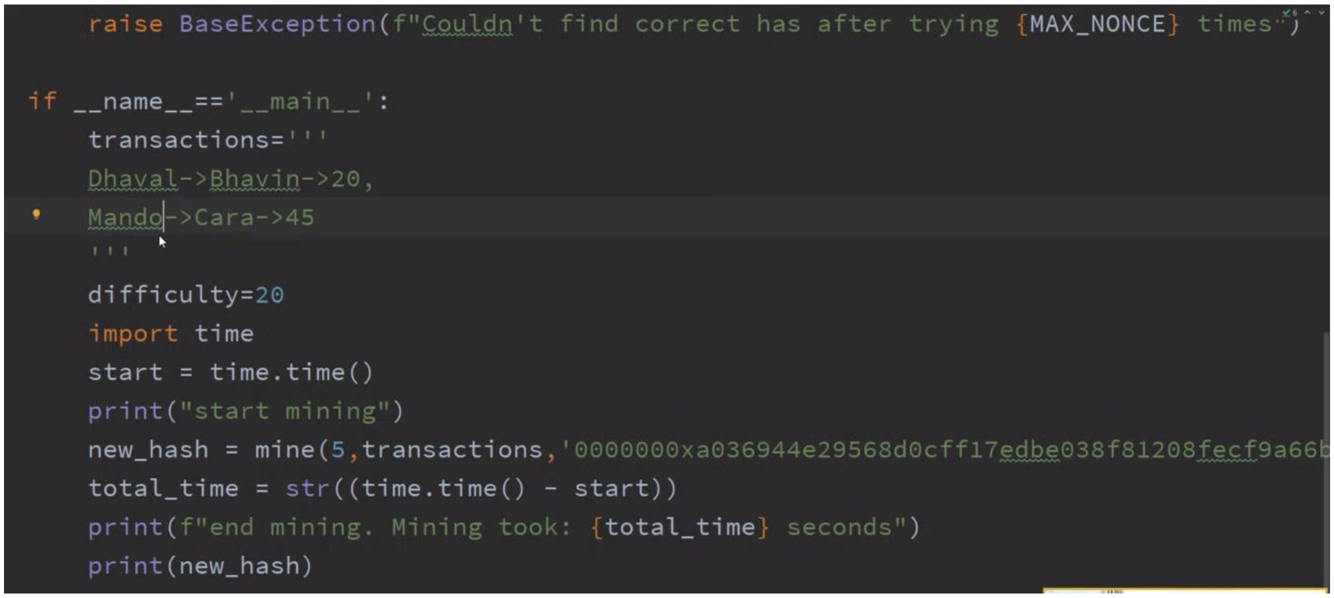
Hash	0000000000000000000072acfce27dfa7f578bdfb316f6216bfb6a99c716c2f6a
Confirmations	1
Timestamp	2021-01-02 15:03
Height	664186
Miner	Unknown
Number of Transactions	1,988
Difficulty	18,599,593,048,299.49
Merkle root	e96769ec1719275a8694cc9ba6ee63c5d3884774db4b61aa275eb753442c0422
Version	0x20c00000
Bits	386,867,735
Weight	3,998,661 WU
Size	1,122,360 bytes
Nonce	84,455,526
Transaction Volume	2042.56766817 BTC
Block Reward	6.25000000 BTC
Fee Reward	0.69006546 BTC

Learn more about how blocks work.

Block Transactions

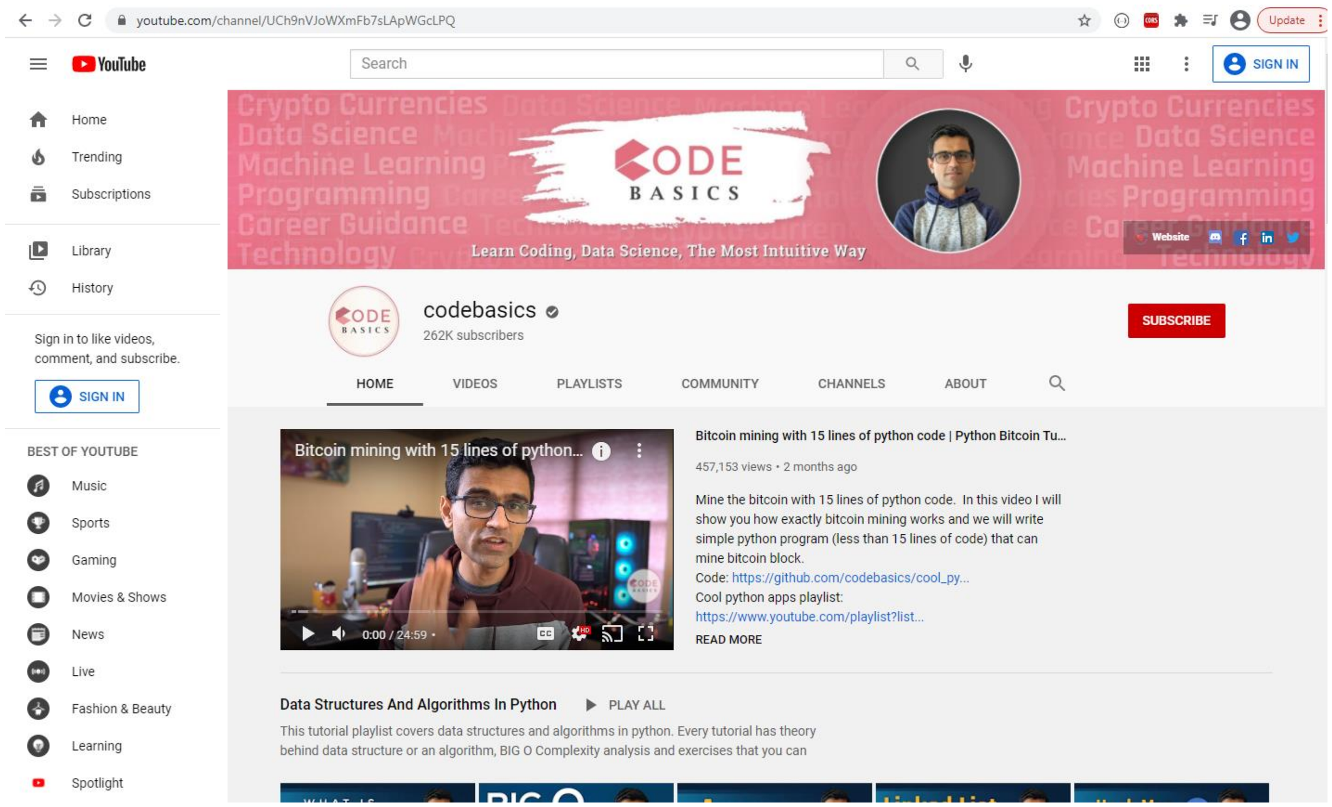


```
from hashlib import sha256
MAX_NONCE = 100000000000

def SHA256(text):
    return sha256(text.encode("ascii")).hexdigest()

def mine(block_number, transactions, previous_hash, prefix_zeros):
    prefix_str = '0'*prefix_zeros
    for nonce in range(MAX_NONCE):
        text = str(block_number) + transactions + previous_hash + str(nonce)
        new_hash = SHA256(text)
        if new_hash.startswith(prefix_str):
            print(f"Yay! Successfully mined bitcoins with nonce value:{nonce}")
            return new_hash

    raise BaseException(f"Couldn't find correct has after trying {M

if __name__=='__main__':
    transactions='''
```



```
    raise BaseException(f"Couldn't find correct has after trying {MAX_NONCE} times")

if __name__=='__main__':
    transactions='''
    Dhaval->Bhavin->20,
    Mando->Cara->45
    '''
    difficulty=20
    import time
    start = time.time()
    print("start mining")
    new_hash = mine(5,transactions,'0000000xa036944e29568d0cff17edbe038f81208fecf9a66b
    total_time = str((time.time() - start))
    print(f"end mining. Mining took: {total_time} seconds")
    print(new_hash)
```

The real difficulty is who might guess the correct nonce value faster.



youtube.com/channel/UCh9nVJoWXmFb7sLApWGcLPQ

Sign in to like videos, comment, and subscribe.

codebasics
262K subscribers

Bitcoin mining with 15 lines of python code | Python Bitcoin Tu...
457,153 views • 2 months ago

Mine the bitcoin with 15 lines of python code. In this video I will show you how exactly bitcoin mining works and we will write simple python program (less than 15 lines of code) that can mine bitcoin block.
Code: https://github.com/codebasics/cool_py...
Cool python apps playlist:
https://www.youtube.com/playlist?list...

Data Structures And Algorithms In Python

This tutorial playlist covers data structures and algorithms in python. Every tutorial has theory behind data structure or an algorithm, BIG O Complexity analysis and exercises that you can