# Creating my own Block Chain

In [3]:
```
import hashlib

# to generate hash value we are using sha256 algorithm which can generate upto 256 bit hashvalue
# SHA-256 is a patented cryptographic hash function that outputs a value that is 256 bits long.

def hash_generator(data):
    result = hashlib.sha256(data.encode())
    return result.hexdigest()

## Block with 3 main parameters
## 1.Data
## 2.Hash Value of Block
## 3.Hash value of Previous Block

class Block:
    def __init__(self,data,hash,prev_hash):
        self.data = data
        self.hash=hash
        self.prev_hash=prev_hash

class Blockchain:
    def __init__(self):
        ## hashLast means hash of Last Block
        hashLast =hash_generator('gen_last')
        ## hast Start means hash of that block
        hashStart = hash_generator('gen_hash')

        ## genesis is first block of block chain or we can say base of all block
        genesis=Block('gen_data',hashStart,hashLast)
        self.chain =[genesis]

    #method to add new block in BlockChain
    def add_block(self,data):
        prev_hash =self.chain[-1].hash
        hash =hash_generator(data+prev_hash)
        block = Block(data,hash,prev_hash)
        self.chain.append(block)
```

In [4]:
```
block_chain = Blockchain()
block_chain.add_block('1')
block_chain.add_block('2')
block_chain.add_block('3')
```

In [5]:
```
for block in block_chain.chain:
    print(block.__dict__)
```

```
{'data': 'gen_data', 'hash': '0a87388e67f16d830a9a3323dad0fdfa4c4044a6a6389cab1a0a37b651a5717b', 'prev_hash': 'bd6fecc16d509c74d23b04f00f936705e3eaa907b04b78872044607665018477'}
{'data': '1', 'hash': 'e3e6c97161f3deaf01599fda60ba85593b07f70328bf228473d1d408f7400241', 'prev_hash': '0a87388e67f16d830a9a3323dad0fdfa4c4044a6a6389cab1a0a37b651a5717b'}
{'data': '2', 'hash': '47e8645e3c14bd4034a498aa88ea630bc0793375207bf90ca469792a5d9484e1', 'prev_hash': 'e3e6c97161f3deaf01599fda60ba85593b07f70328bf228473d1d408f7400241'}
{'data': '3', 'hash': '82084603decb1a14a8819dacaa86197659f1e150c4a50186e68043004b5a3c06', 'prev_hash': '47e8645e3c14bd4034a498aa88ea630bc0793375207bf90ca469792a5d9484e1'}
```

In [6]:
```
bc = Blockchain()
bc.add_block("Saurabh")
bc.add_block("Ankit")
bc.add_block("Raj")
bc.add_block("Aakash")
bc.add_block("Aarush")
bc.add_block("Vansham")
bc.add_block("Abhijeet")
```

In [7]:
```
for block in bc.chain:
    print(block.__dict__)
```

```
{'data': 'gen_data', 'hash': '0a87388e67f16d830a9a3323dad0fdfa4c4044a6a6389cab1a0a37b651a5717b', 'prev_hash': 'bd6fecc16d509c74d23b04f00f936705e3eaa907b04b78872044607665018477'}
{'data': 'Saurabh', 'hash': '29c99adecbd82bf3573e430cf9703b753b62e21946668004033653441638 4a7f', 'prev_hash': '0a87388e67f16d830a9a3323dad0fdfa4c4044a6a6389cab1a0a37b651a5717b'}
{'data': 'Ankit', 'hash': '929d530fc7f25b37c209f85270e15ca88f11ef522d956d3b9a8b96bca9ad32c8', 'prev_hash': '29c99adecbd82bf3573e430cf9703b753b62e21946668004033653441638 4a7f'}
{'data': 'Raj', 'hash': '6f507f66d440ca68047869617563290446f7245b54c75d4a80fa6deed435562d', 'prev_hash': '929d530fc7f25b37c209f85270e15ca88f11ef522d956d3b9a8b96bca9ad32c8'}
{'data': 'Aakash', 'hash': '58e8fcc7a94068c5de7247a55017880e58e5668bcf83fce2edf9b5c8c09e88dc', 'prev_hash': '6f507f66d440ca68047869617563290446f7245b54c75d4a80fa6deed435562d'}
{'data': 'Aarush', 'hash': 'a94e3750b796df930a7788321a01d61352cf4c11cd8e07a719b81f767a62b42a', 'prev_hash': '58e8fcc7a94068c5de7247a55017880e58e5668bcf83fce2edf9b5c8c09e88dc'}
{'data': 'Vansham', 'hash': 'a3755465441fd21a5947727aa1dddf2520de67f627d3d5f82e96649c521329fd', 'prev_hash': 'a94e3750b796df930a7788321a01d61352cf4c11cd8e07a719b81f767a62b42a'}
{'data': 'Abhijeet', 'hash': '7e141003121272cbec1af921a4bdfb18f1b24b5570eed2c1a72df7c79587247c', 'prev_hash': 'a3755465441fd21a5947727aa1dddf2520de67f627d3d5f82e96649c521329fd'}
```

In [ ]: