**Lesson Plan – Adding blocks to the blockchain**

**Learning objectives -**

1. **Revise concepts of blockchain class creation from previous class**
2. **How to add more blocks to the blockchain**

**Materials required -**

**repl.it login credentials**

**Adding More Blocks**

Adding a block to a blockchain typically involves a few steps, which may vary slightly depending on the specific blockchain implementation. Here is a general overview:

**Verify transactions**: Before adding a block to the blockchain, the transactions that will be included in the block need to be verified. This involves checking that each transaction is valid, which typically includes checking that the sender has sufficient funds, that the transaction is properly signed, and that it hasn't already been included in a previous block.

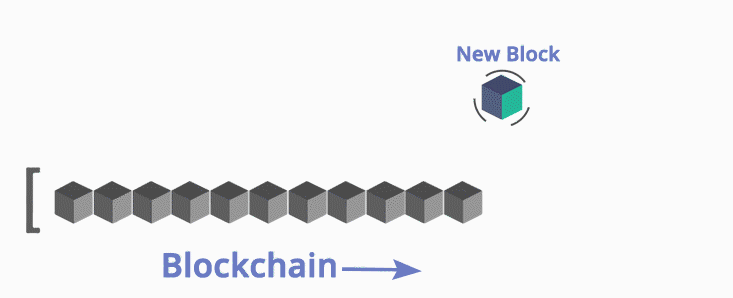
**Create a block**: Once the transactions have been verified, a new block is created. This block typically contains a header, which includes metadata such as the block's version number, a timestamp, and a reference to the previous block in the chain, as well as a list of the verified transactions that will be included in the block.

**Solve the proof-of-work**: Many blockchain implementations use a proof-of-work algorithm to ensure that new blocks are added to the blockchain at a predictable rate and to prevent any one node from monopolizing the process. This involves solving a complex mathematical puzzle, which requires significant computational power. Once a node solves the puzzle, it can add the new block to the chain.

**Propagate the block**: Once a node has added a new block to the chain, it will propagate the block to other nodes on the network so that they can update their own copies of the blockchain.

**Verify the new block**: Other nodes on the network will receive the new block and will verify that it is valid. This typically involves checking that the proof-of-work has been solved correctly, that the transactions in the block are valid, and that the block has been added to the correct place in the blockchain.

Once a block has been added to the blockchain, it is considered a permanent part of the chain, and subsequent blocks will build upon it. This creates an immutable ledger of transactions that is difficult to alter without consensus from a majority of the network.



**Keyterm -**

1. **Metadata** is defined as the data providing information about one or more aspects of the data.
2. **Monopolize** is defined as the ability to obtain exclusive possession or control of an organization or a group.
3. **Immutable ledger in blockchain** refers to any records that have the ability to remain unchanged.
4. **Consensus mechanism** is any method used to achieve agreement, trust, and security across a decentralized computer network.

Introducing the lesson/project including the concepts (Time - 45 min)

Link to Repl it Project :

https://replit.com/@ArindamJetLearn/Lesson6AddingMoreBlocks#main.py

**Homework –**

# Write a program to find the sum of all items in a dictionary

<https://www.geeksforgeeks.org/python-program-to-find-the-sum-of-all-items-in-a-dictionary/>

Write a program to

* [Get Current Timestamp](https://pynative.com/python-timestamp/#h-get-current-timestamp) using
  + [Datetime to Timestamp](https://pynative.com/python-timestamp/#h-datetime-to-timestamp)
  + [Get Timestamp Using time Module](https://pynative.com/python-timestamp/#h-get-timestamp-using-time-module)
  + [Get Timestamp Using calendar Module](https://pynative.com/python-timestamp/#h-get-timestamp-using-calendar-module)

Also convert timestamp to string and datetime (format)

https://pynative.com/python-timestamp/