**How do Blockchains Work:**

**Demo link:** <https://andersbrownworth.com/blockchain/tokens>

**Understanding Hash Functions:**

At its simplest, a hash is a unique, fixed-length string that serves to identify any piece of data. When you input any kind of data into a hash function, it produces a hash. In this demo, the hash algorithm we’ll focus on is SHA-256.

If I add `Patrick Collins` to our `SHA-256` algorithm, it will:

1. Convert the letters to numbers

2. Convert the numbers to a fixed-length “string” or “hash”

`Patrick Collins` gets converted to `7e5b5a1a6b80e2908b534dd5728a998173d502469c37121dd63fca283068077c`

Ethereum, uses its own version of a hashing algorithm (Keccak256) that isn’t exactly SHA-256 but belongs to the SHA family. This doesn’t change things significantly here as we’re primarily concentrating on the concept of hashing.

In the application, whatever data you enter into the data section, undergoes processing by the SHA-256 hash algorithm resulting in a unique hash.

For example, when I input my name as “Patrick Collins,” the resulting hash uniquely represents “Patrick Collins.” The fascinating aspect is, no matter how much data is input, the length of the generated hash string remains constant.

**Understanding Blocks:**

Now that we’ve grasped the concept of hashing and fixed-length string, let’s inspect the structure of a blockchain. A collection of “blocks.

A block takes the same data input, but instead of a singular data field, a block is divided into ‘block’, ‘nonce’, and ‘data.’ All three are then run through the hash algorithm, producing the hash for that block. As a result, even a minor change in the data leads to an entirely different hash, hence, invalidating the block.

In essence, mining involves the computational trial and error process of finding an acceptable value to produce a hash which typically follows a certain pattern, such as starting with four zeros. The value found, which satisfies this criterion, is known as the ‘nonce’.

The problem or criteria a miner has to solve will vary from blockchain to blockchain, but the concept is the same.

**The Inherent Beauty of Blockchain: Immutability:**

In a blockchain, which is essentially a sequence of blocks, each block is comprised of the previous elements – a block number, a nonce and data – as well as `the hash of the previous block`

What this means in practice is that any changes to data, in any block of the chain, will invalidate every proceeding block, until they are recalculated, or re-mined.

Genesis Block: This is the first block in a blockchain.

**Decentralized Distribution:**

Now, if a single entity were to control the blockchain, they could conceivably change any data they want, and then re-mine, or re-validate subsequent blocks. This is bad.

Enter Decentralized Distribution.

The crux of blockchain’s power lies in its decentralization or distributed nature. Under this system, multiple entities or “peers” run the blockchain technology, each holding equal weight and power. In the event of disparity between the blockchains run by different peers (due to tampering or otherwise), the majority hash wins, as the majority of the network agrees on it.

Nodes that don’t agree with the majority effectively fork the network, continuing on their own with their own history.

**Interplay of Blockchain & Transactions:**

Until now we’ve been considering the data passed in a block to be a random string of text, but the reality is – this data can be anything. In the token and coinbase sections of this demo you can see how each block is comprised of a number of transactions that all get hashed together. Any edits to any of these transactions is going to invalidate the chain!

**Wrap Up:**

To summarize, every transaction, block, and indeed the whole blockchain itself comes down to understanding the concept of a hash. This unique fixed-length string that is intrinsically linked with the original data. We’ve also underscored the importance of decentralization and highlighted how the concept of immutability plays into the system’s security.