What's in a Block?

A block is a data structure that contains transactions as well as metadata about the block. All transactions must be included in a valid block in order to be considered final. All full nodes on the network will store a copy of the block in order to validate new transactions and share the data with other nodes.

Block Structure

```
$ bitcoin-cli -regtest getblockhash 0
0f9188f13cb7b2c71f2a335e3a4fc328bf5beb436012afca590b1a11466e2206

$ bitcoin-cli -regtest getblock 0f9188f13cb7b2c71f2a335e3a4fc328bf5beb436012afca590b1a11466e2206 0
```

```
0100000000000000000000000000000000000000000000000000000000000000000000003ba3edfd7a7b12b27ac72c3e67768f617fc
81bc3888a51323a9fb8aa4b1e5e4adae5494dffff7f20020000000101000000010000000000000000000000000000000000000000000
000000000000000000000ffffffff4d04ffff001d0104455468652054696d65732030332f4a616e2f32303039204368616e63656c6c6
f72206f6e206272696e6b206f66207365636f6e64206261696c6f757420666f722062616e6b73ffffffff0100f2052a010000004341
04678afdb0fe5548271967f1a67130b7105cd6a828e03909a67962e0ea1f61deb649f6bc3f4cef38c4f35504e51ec112de5c384df7ba0
b8d578a4c702b6bf11d5fac00000000
```

This is the genesis block for the regtest network, and as you can see, it is not really human readable. We are going to learn how the data is encoded from left to right.

```
{block header} {transaction counter} {transactions}
```

Block Header

```
{version 4B} {previous block hash 32B} {merkle root hash 32B} {time 4B} {bits 4B} {nonce 4B}
```

- `version` (4 Bytes) - Block format version.
- `previous block hash` (32 Bytes) - The hash of the preceding block. This is important to include in the header because the hash of the block is calculated from the header, and thus depends on the value of the previous block, linking each new block to the last. This is the link in the chain of the blockchain.
- `merkle root hash` (32 Bytes) - The hash of the merkle tree root of all transactions in the block. If any transaction is changed, removed, or reordered, it will change the merkle root hash. This is what locks all of the transactions in the block.
- `time` (4 Bytes) - Timestamp in Unix Time {TODO: provide link} (seconds). Since the clocks of each node around the world is not guaranteed to be synchronized, this is just required to be within {TODO} of the rest of the network.
- `bits` (4 Bytes) - Target hash value in Compact Format. The block hash must be equal to or less than this value in order to be considered valid.
- `nonce` (4 Bytes) - Can be any 4 Byte value, and is continuously changed while mining until a valid block hash is found.

Values::

- **version** - `01000000` (1)
- **previous block hash** - `0000000000000000000000000000000000000000000000000000000000000000`
- **merkle root hash** - `3ba3edfd7a7b12b27ac72c3e67768f617fc81bc3888a51323a9fb8aa4b1e5e4a`
- **time** - `dae5494d` (1296688602 Wednesday, February 2, 2011 11:16:42 PM GMT)

- **bits** - `ffff7f20` (7fffff0000000000000000000000000000000000000000000000000000000000)
- **nonce** - `02000000` (2)

Note: Remember Bitcoin uses little endian format, which may seem backwards. You can use the provided tool `$ reverse_endian {hex}` to reverse the endianess of a hex string.

Transaction Counter

`01` (1) Transaction in the block

- Transaction Counter (Variable Length) - Number of transactions in the block represented as a [Variable Length Integer](#).

Transactions

See 2.2-Transactions.md