What's in a Transaction?

A transaction contains inputs and outputs. An output is the receiver of the coins, and the input points to a previously unspent output. When you create a transaction, typically you gather enough inputs from previous transactions that you've recieved, and then create the new outputs to which the coins will be sent. Usually there is more value in the inputs that you want to send, so you create an additional output (a change output) where the remainder is sent back to yourself. Finally, the difference between the total input value and the total output value is considered the transaction fee because it is the leftover that the miner can claim when they mine the block.

Transaction Structure

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

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

```
010000000000000000000000000000000000000000000000000000000000000000000000ba3edfd7a7b12b27ac72c3e67768f617fc81bc3888a51323a9fb8aa4b1e5e4adae5494dffff7f20020000000101000000010000000000000000000000000000000000000000000000000000000000000000000000000000000000ffffffff4d04ffff001d0104455468652054696d65732030332f4a616e2f323030392043686616e63656c6c6f72206f6e206272696e6b206f66207365636f6e64206261696c6f757420666f722062616e6b73ffffffff0100f2052a0100000043410 4678afdb0fe5548271967f1a67130b7105cd6a828e03909a67962e0ea1f61deb649f6bc3f4cef38c4f35504e51ec112de5c384df7ba0 b8d578a4c702b6bf11d5fac00000000
```

We know from Lesson 1 that the gensis block contains one transaction and that part of the block is:

```
0100000001000000000000000000000000000000000000000000000000000000000000000000000000ffffffff4d04ffff001d0104455468652054696d65732030332f4a616e2f323030392043686616e63656c6c6f72206f6e206272696e6b206f66207365636f6e64206261696c6f757420666f722062616e6b73ffffffff0100f2052a0100000043410 4678afdb0fe5548271967f1a67130b7105cd6a828e03909a67962e0ea1f61deb649f6bc3f4cef38c4f35504e51ec112de5c384df7ba0b8d578a4c702b6bf11d5fac00000000
```

This is a coinbase transaction. A coinbase transaction is unique because each block contains one (and only one) coinbase transaction where new coins are minted. Depending on the height of the block, a reward is given based on the distribution algorithm. For the first 210,000 blocks the miner can claim 50 new BTC. Since these are newly minted coins, there are no inputs that are spent.

```
{version} {flag (optional)} {input counter} {inputs} {output counter} {outputs} {witnesses (optional)} {lock time}
```

- `version` (4 Bytes) - Transaction format version
- `flag` (2 Byte Array) - Optional flag, if present, must be 0001, which indicates there is witness data in this transaction
- `input counter` (Variable Length) - Number of inputs in the transaction represented by a Variable Length Integer.
- `inputs` (based on Input Counter) - List of all transaction inputs which will be spent and which reference unspent transaction outputs from previous transactions.
- `output counter` (Variable Length) - Number of outputs in the transaction represented by a Variable Length Integer.
- `outputs` (based on Output Counter) - List of all transaction outputs where the coins will be sent and which will become unspent transaction outputs to be spent in future transactions.
- `witnesses` (variable) - Optional list of all witnesses only if Flag is present, which are used in transaction validation according to specifications for Segregated Witness.

- `locktime` (4 Bytes) - If non-zero and sequence numbers are < `ffffffff` : it represents either the block height or timestamp when transaction is final.

Values:

- `version - 01000000` (1)
- `flag` - N/A
- `input counter - 01` (1)
- `inputs -` `0000000000000000000000000000000000000000000000000000000000000000ffffffff4d04ffff001d0104455468652054696d65732030332f4a616e2f3230303920436861696e63656c6c6f72206f6e206272696e6b206f66207365636f6e64206261696c6f757420666f722062616e6b73ffffffff`
- `output counter - 01` (1)
- `outputs -` `00f2052a01000000434104678afdb0fe5548271967f1a67130b7105cd6a828e03909a67962e0ea1f61deb649f6bc3f4cef38c4f35504e51ec112de5c384df7ba0b8d578a4c702b6bf11d5fac00000000`
- `locktime - 00000000` (N/A)

Inputs

```
0000000000000000000000000000000000000000000000000000000000000000ffffffff4d04ffff001d0104455468652054696d65732030332f4a616e2f3230303920436861696e63656c6c6f72206f6e206272696e6b206f66207365636f6e64206261696c6f757420666f722062616e6b73ffffffff
```

```
{previous tx hash} {previous utxo index} {scriptSig length} {scriptSig} {sequence number}
```

- `previous tx hash` (32 Bytes) - Hash of the previous transaction. This is used to lookup the transaction in order to validate that this input can be spent.
- `previous utxo index` (4 Bytes) - Previous Unspent Transaction Output (UTXO) index. Since transactions can have multiple outputs, in order to specify which output, you can provide a zero-based index.
- `scriptSig length` (Variable Length) - Length in bytes of the scriptSig.
- `scriptSig` (based on `scriptSig length`)- The first part of the script that is executed before the scriptPubKey from the UTXO. When they are both executed in order, the result must be true in order to be considered a valid spend.
- `sequence number` (4 Bytes) - Used as a relative lock time if transaction version is >= 2. See BIP68.

Values:

- `previous tx hash - 0000000000000000000000000000000000000000000000000000000000000000` (N/A for coinbase transaction)
- `previous utxo index - ffffffff` (N/A for coinbase transaction)
- `scriptSig length - 4d` (77 Bytes)
- `scriptSig -` `04ffff001d0104455468652054696d65732030332f4a616e2f3230303920436861696e63656c6c6f72206f6e206272696e6b206f66207365636f6e64206261696c6f757420666f722062616e6b73`
- `sequence number - ffffffff`

Note: The scriptSig in the coinbase transaction of the genesis block contains a message from Satoshi Nakamoto, the pseudonymous creator of Bitcoin. If you use ascii encoding:

```
$ printf
"04ffff001d0104455468652054696d65732030332f4a616e2f3230303920436861696e63656c6c6f72206f6e206272696e6b206f662(
 | xxd -r -p && echo
▯▯EThe Times 03/Jan/2009 Chancellor on brink of second bailout for banks
```

There is speculation that it hints the reason that Bitcoin was created, as well as provides a verifiable timestamp that the block could not have been created before the news article was published.

Outputs

`00f2052a01000000434104678afdb0fe5548271967f1a67130b7105cd6a828e03909a67962e0ea1f61deb649f6bc3f4cef38c4f35504e51ec112de5c384df7ba0b8d578a4c702b6bf11d5fac00000000`

`{coin value} {scriptPubKey length} {scriptPubKey}`

- `coin value` (8 Bytes) - The value in satoshis of the output, i.e. how much Bitcoin is being sent. There are 1e8 satoshis per Bitcoin.
- `scriptPubKey length` (variable length) - The size of the scriptPubKey in bytes represented as a Variable Length Integer.
- `scriptPubKey` (based on scriptPubKey length) - The second part of the script that is executed after the scriptSig when this output is spent. The scriptPubKey is said to "encumber" the coins meaning they can only be spent if a valid solution to the scriptPubKey is provided.

Values:

- `coin value - 00f2052a01000000` (50 BTC or 5000000000 satoshis)
- `scriptPubKey length - 43` (67 Bytes)
- scriptPubKey - `4104678afdb0fe5548271967f1a67130b7105cd6a828e03909a67962e0ea1f61deb649f6bc3f4cef38c4f35504e51ec112de5c384df7ba0b8d578a4c702b6bf11d5fac`