Solidity Cheat Sheet

Solidity

- It is a contract-oriented high-level language for implementing smart contracts. It was influenced by C++, JavaScript, and Python and is designed to target the EVM.

Contracts

- Contracts in solidity are similar to classes in object-oriented language. A contract can be created using "new" keyword.
- contract L { function transfer(address recipient, uint256 amount) { return _p + _q; } }

ABI & WEI

- ABI: A data encoding scheme called "Application Binary Interface" (ABI) is used in working with smart contracts.
- WEI: The smallest denomination or base unit of Ether

Global Variables

- blockhash: hash of the given block which works for the 356 most recent blocks excluding the current block.
- block.coinbase (address): shows miner's address of current block.
- block.number: number of current block.
- block.timestamp: the gas that remains.
- msg.value: amount of WEI sent along with the message.

Pragma

- Used to specify certain conditions under which the source files can or cannot run.
- pragma solidity ^0.22; // function f(x) { return x; }

Interface

- In solidity, interface is defined as contracts, but the function bodies are omitted for the functions.
- Example:

<table>
<thead>
<tr>
<th>Function</th>
<th>Parameters</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>transfer</td>
<td>(address, uint256)</td>
<td>send given amount of Wei to the address given, on failure it returns false as an output</td>
</tr>
</tbody>
</table>

Data Types

- Byte: "uint8" represents one byte (8 bit).
- Uint: "uint256" represents 256 bit.

Transaction Example

- From: 0x712 To: 0x145
- Data: 0x614
- Transaction Example

<table>
<thead>
<tr>
<th>BlockNumber</th>
<th>TransactionHash</th>
</tr>
</thead>
<tbody>
<tr>
<td>0x712</td>
<td>0x145</td>
</tr>
</tbody>
</table>

Important Terms

- Voting: When the contract is quiet complex, it uses voting contract, it shows how delegated voting can be done so that vote counting is automatic and completely transparent at the same time.
- Delegate call: It is a reusable library code that can be applied to a contracts storage in solidity.
- Logs: A feature called logs is used in solidity to implement events.
- NVM: is a convenient and portable way to install Solidity compiler.
- Truffle: It is a test bed that will allow to easily test and compile the smart contract.
- Inheritance: In solidity inheritance is more syntactic, it has its own scoping and visibility rules.