Blockchain Bootcamp Day 3

* Misconceptions about blockchain development
* Solidity versions -> the best one is not always the latest version.
* Most projects they will not have the latest solidity version.
* Most project need to be audited.
* A couple of months for smart contracts ready.
* Current smart contracts in production are not latest version.
* Labs and Hands on materials are usually up to date.
* Ethereum-blockchain-developer.com has the materials for the course.
* Watch the theory, labs, and use the actual lab guide
* Actually do the labs yourself
* How to get the most out of it!!!  
  \* Do labs and show off on facebook group;
* Occasionally: go back to previous labs.
* Test your knowledge by answering questions.
* In cases when you are stuck, google, course q&a search, also the teachers will help.
* How to create smart contracts?
* Use the solidity programming language.
* Use online ide -> remix.ethereum.org.
* Each solidity file starts with a pragma with states the solidity version.
* Pragma solidity ^0.5.11 -> solidity version 5 or higher
* Create a new contract using the contract keyword
* When deploying a smart contract to account, it will create a etherscan server with the details of the transaction.(After the transaction is already mined)
* In deployed contract section, new instance to interact with new smart contract. Each property becomes a getter function.

How smart contracts work?

* A piece of code running on the blockchain. It’s a state chain.
* Needs transactions to change the state of a smart contract.
* Can do logic operations.
* State changes or state transition happened through mining + transactions.
* It’s implements the turing complete theory. Can complete anything with smart contract. It can solve any computational problem with smart contracts.
* You send evm bytecode. You are sending compiled code to the blockchain.(The data field)
* Solidity if one of the changes that is send as compiled code.
* It’s derived from ecmescript, but doesn’t have a lot compared to javascript.
* If you deploy a smart contract, every participating node in the network. Executes transaction and make sure surviving node validates smart contract. That is why there is shared ledger between participating nodes.
* Viper is also used, created by creator of Ethereum. Research-oriented derived from python.
* Structure of a solidity contract 🡪 Has a class like structure. -> Inheritable 🡪 functions, if/else, for/while, datatypes(uint, Boolean, array, struct, mapping , address, NO FLOATS!), special structures like modifiers. Imports)
* Everything is statically types, can’t convert datatypes dynamically unlike to javascript.
* Style guide for solidity files. Capwords style.
* The version pragma is a pre-compiler version, lock in specific solidity compiler version. This deals with security.
* There are variables.

The difference between developer blockchains and real blockchains.

* Speed of a transaction completes also depends how congested the network is.
* In remix the javascript vm. It mocks a blockchain in the browser
* Can switch between multiple mock accounts
* Connect to web3 provider, to a external blockchain node running externally such as ganache.
* RPC server that it’s running on 127.0.0.1:7545
* Javascript vm preferred with remix smart contract development, while ganache for local smart contract development.

Ethereum Nodes Explained.

* 4 different nodes(not blocks) connected with each other, talking to each other. They are talking each using the ethereum protocol. It doesn’t have to talk in same software, a tezos node can talk to a Ethereum node in the Ethereum protocol
* Can use Ethereum in a private network. Separated from main network.
* The main network does not contain the same data to the private network.
* Can suitable for banks or government for data privacy
* Can have some data from main network, then add private transaction in private network. The private network starts with a genesis block. The genesis file is the configuration of the genesis block. Only file you have configure to have private network..
* Main network where Ethereum cost money, and data is changing.
* Ropsten, rinkeby, gorli, other network and similar to main network. THEY ARE TESTING NETWORKS for protocols changes, and consensus protocol. THEY ALL STILL REQUIRE mining.
* Most of the time a lot of devs use developer networks.
* Javascript vm you can return values from transactions, while real network you CAN”T do this.
* Testing on ganache before going to production, test on real networks.
* Main network data is persistent, real-blockchain. Costs real money.
* Test network -> persisten but can be deleted. -> Real blockchain -> beta release.
* Developer network -> unit testing-> fast -> non-persistent.(private networks).