**Solidity 32**

<https://www.youtube.com/watch?v=gyMwXuJrbJQ>

**Contract**: Contracts are similar to Class in any OOP Language.

2:18:42 - Functions

2:35:30 - Arrays & Structs

2:45:35 - Errors & Warnings

2:46:35 - Memory, Storage, Calldata (Intro)

2:50:15 - Mappings

2:53:40 - First Contract

WRITING TO A BLOCKCHAIN/ ANY CHANGE OF STATE REQUIRES GAS

1. The ‘view’, ‘pure’ functions only Read State of the blockchain, so no gas required, no change of state (Blue color button on Remix IDE)
2. Calling a view function is free unless you are calling it from a function that costs gas

Public: Visible externally and internally (creates a *getter function* for storage/state variables)

Private: Only visible in the current contract

External: Only visible externally (only for functions)

Internal: Only visible internally

Struct: Creating a new type.

Structs are **custom data types that can group several variables**. They represent a record of “something”. Suppose you want to keep a list of “To Do's” and need to know if they are completed. One would keep a simple list and create a Struct: item “To Do”

**Data Storing Types:**

EVM can access and store information in six places:

(Storage, Memory, Calldata) Important

1. Stack
2. Memory
3. Storage
4. Calldata
5. Code
6. Logs

Cannot call variable as Stack, Code or Logs

Calldata & Memory means the variable will only exist temporarily

Storage variables can exist outside the executing function

**Storage**: Permanent variables that can be modified

**Memory**: Temporary variables that can be modified

**Calldata**: Temporary variable that can’t be modified

*Array, Struct, Mapping special types in solidity\*\**

String an Array of Bytes

Data location only required for Array, Struct or Mapping types. So, not required for uint256

uint256 in memory by default

*Cannot give ‘Storage’ for string data in function because length of Array, Struct or Mapping are not fixed\*\**

Storage can only be used as state variables

**Mapping:**

Hash tables

Map is an Associative Array. Like a Dictionary.

mapping (string => uint256) public nameToFavNum;

nameToFavNum is the name of the Map

We have a dictionary where every single name is mapped to specific number

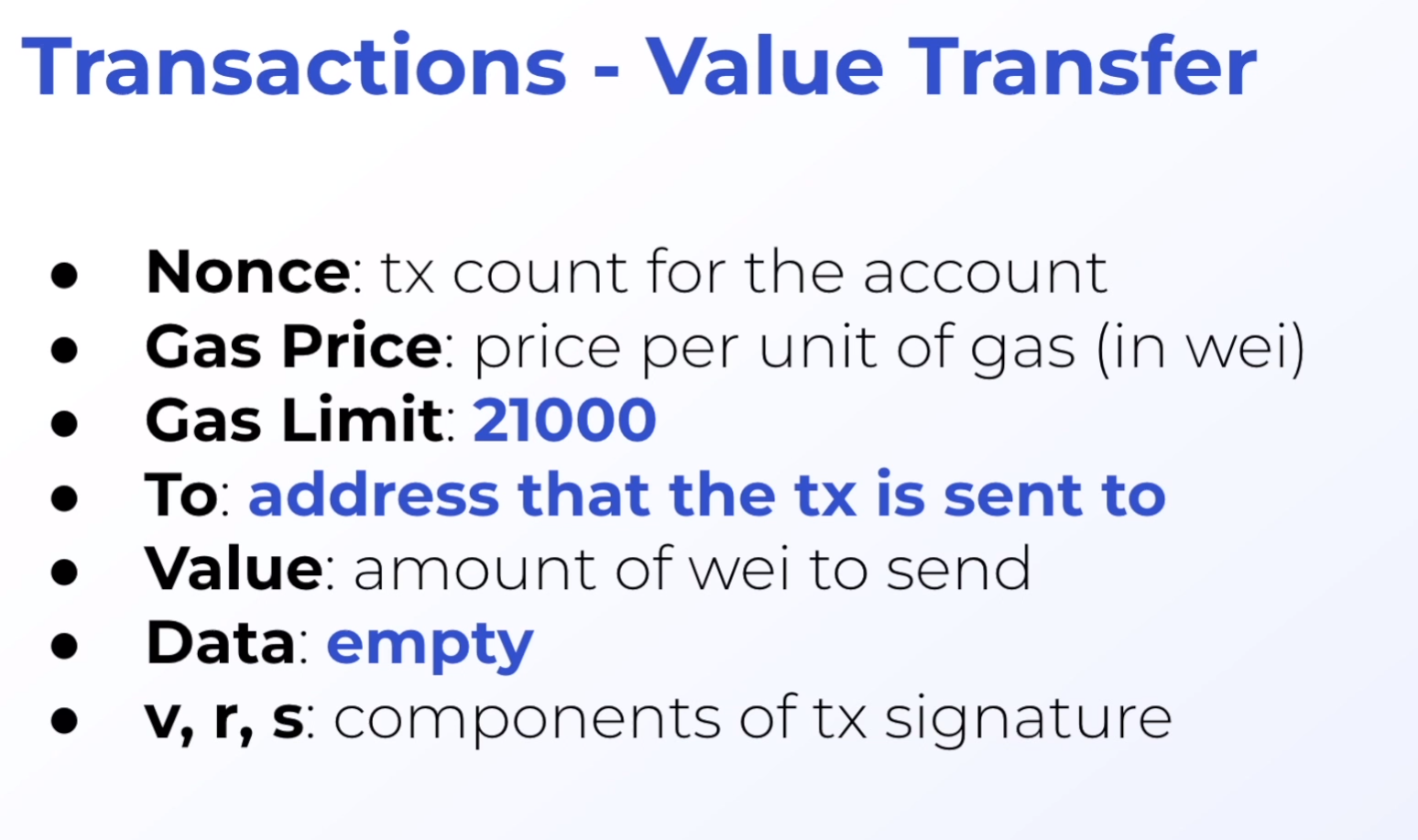
EVM compatible blockchains are Avalanche, Fantom, Polygon

**Virtual & Override:**

Write ‘virtual’ to the parent function to make it overridable

Write ‘override’ to the child function to override the parent function

**FundMe.sol**



**“msg.sender” IS WHOEVER CALLS THE FUNCTION**

([04:25:03](https://www.youtube.com/watch?v=gyMwXuJrbJQ&t=15903s)): Libraries

**Library**

Libraries are similar to contracts, but you can't declare any state variable and you can't send ether.

A library is embedded into the contract if all library functions are internal.

Otherwise the library must be deployed and then linked before the contract is deployed.

([04:29:54](https://www.youtube.com/watch?v=gyMwXuJrbJQ&t=16194s)): SafeMath, Overflow Checking, and the "unchecked" keyword

msg.sender = address

payable(msg.sender) = payable address

([04:50:26](https://www.youtube.com/watch?v=gyMwXuJrbJQ&t=17426s)): Basic Solidity Constructor

([05:05:41](https://www.youtube.com/watch?v=gyMwXuJrbJQ&t=18341s)): Advanced Solidity - Immutable & Constant

5:37

Node.js is a JavaScript runtime

**Hardhat Focus:**

yarn add solc

7.23

View and Pure function if called outside of a function call don’t cost any gas.

8:37

Lesson 9: Hardhat Smart Contract Lottery

Topic ~ Indexed Parameter

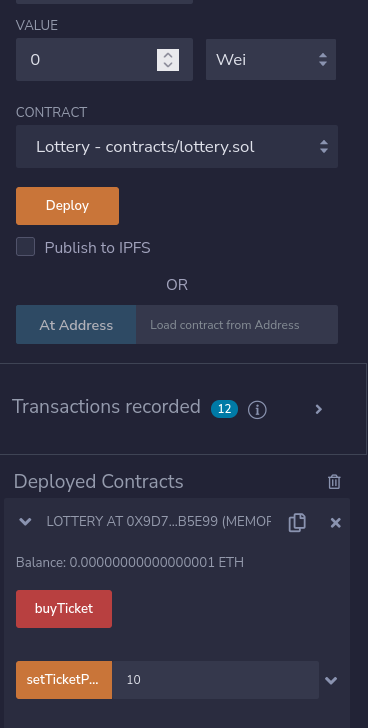
Virtual means it’s waiting to be overridden

**ENUMS:**

Enums can be used to create custom types with a finite set of ‘constant values’

14.47

15:20



***SetTicketPrice*** is for setting Function Parameters

The ***value*** slot is for passing values