## Introduction to Solidity: Coding Ethereum Smart Contracts



blockchair

## Session Four Scope

#### Scope

Learn about ERC20 and understand additional programming nuances

#### • What you will know

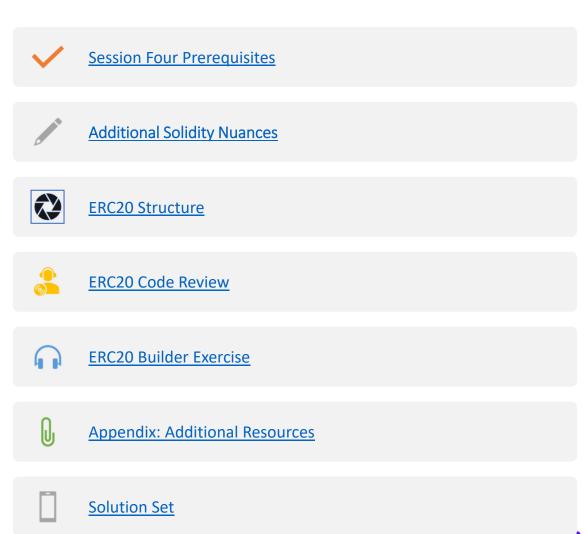
What ERC20 is, know what constructors are, know what mappings are, know what events are

#### • Next Steps: Session Five

Using session four knowledge we write a small ERC 20 contract and deploy it on metamask test network







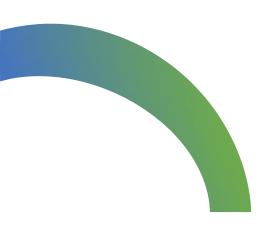
**L**nyc

## Session Four



## Session Four Prerequisites

- Google Chrome
- Remix <a href="https://remix.ethereum.org/">https://remix.ethereum.org/</a>



# Additional Solidity Nuances

## **Additional Solidity Nuances**

- Constructors
- Mappings
- Events
- Interface

## **ERC20 Structure**

What is an ERC20 Token?

ETHEREUM REQUEST FOR COMMENT 20



### **ERC20 Core Functions**

- transfer
- balanceOf
- transferFrom
- approve
- mappings



## **ERC20 Examples**

- AAVE
- USDT
- DAI

### ERC20 v. ERC777 v. ERC677?

• Core difference is that later ERC standards have additional functionalities but are still backwards compatible



## Why ERC20?

- Governance
- Secure a Network
- Synthetic Assets and Stable Coins
- Easy and uniform token creation



## **ERC20** Code Review

## Github Repository

• <a href="https://github.com/PatrickAlphaC/erc20-brownie/blob/main/contracts/TokenERC20.sol">https://github.com/PatrickAlphaC/erc20-brownie/blob/main/contracts/TokenERC20.sol</a>

\* Code Review based on repository of Patrick Collins at Chainlink



## Appendix: Additional Learning Resources

- Tutorialspoint -<u>https://www.tutorialspoint.com/solidity/index.htm</u>
- ERC20 Tutorial <u>https://betterprogramming.pub/python-blockchai</u>

   <u>token-deployment-tutorial-create-an-erc20-</u>
   77a5fd2e1a58

blockchain Lunyc