OpenZeppelin Contracts

For solidity development

Learning Objectives

1. Understanding what OpenZeppelin
2. Benefits of OpenZeppelin Contracts
3. Use Cases for OpenZeppelin Contracts
    a. Token Standards
    b. Access Control
    c. Math
    d. Reentrancy Protection
    e. Pausable
    f. Crowdsale, Distribution, Emission, Validation
4. How to use OpenZeppelin Contracts on the Remix IDE

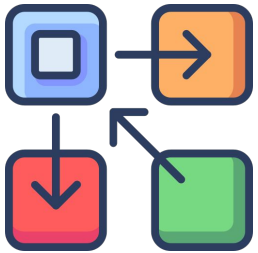
What is OpenZeppelin in Blockchain?

- An Open-source platform that provide tools for development and managing of Dapp.
- Offers a library of reusable, secure smart contract templates.
- Goal:
  - Reduce risk of errors
  - Time efficiency
  - Reliability

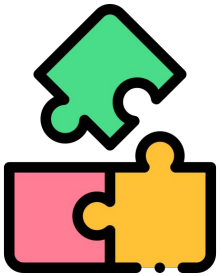
OpenZeppelin

Why OpenZeppelin Contract?

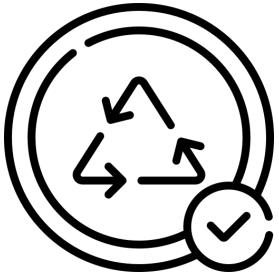
Security



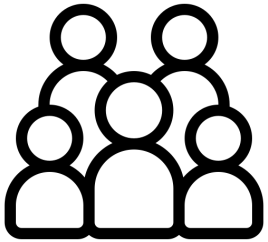
Interoperability



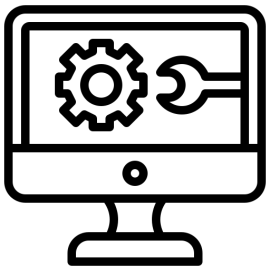
Flexibility



Reusability



Community



Support

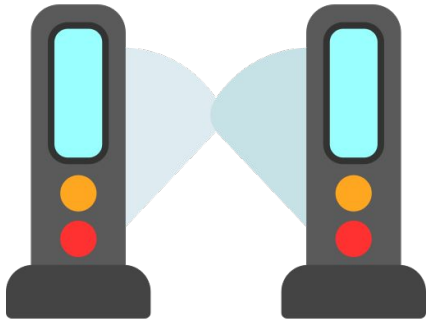
Use Cases

Token Standards Contract

	ERC20	**ERC721**	**ERC777**
Type of Token	Fungible	Non-Fungible	Fungible
Base Interface	IERC20	IERC721	IERC777
Extensions	ERC20Mintable ERC20Burnable ERC20Pausable ERC20Capped	ERC721Mintable ERC721MetadataMintable ERC721Burnable ERC721Pausable	“Operators” “Hooks” “Safe Transfer” “Notifications”

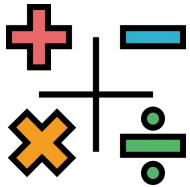
Access Control Contract

- Control Access to Resources and Functionality
- Selective Restrictions and basic authorization control
    - Eg. Whitelisting
- Restricting
    - Voting
    - Minting
    - Sending Transactions
- Provides a level of security and Flexibility



Math Contract

SafeMath - Provide Mathematical functions with **Safety Check** to prevent **overflow** and **underflow** attacks



Reentrancy Guard Contract

Protect against **reentrancy** attacks

Pausable Contract

Pause and **Unpause** a contract to temporarily halt functionalities

Crowdsale Contract

- Manage Crowdsale, include functionalities
  - Token Distribution, Whitelisting and Hard-capping supply of tokens
- Extend Base contract to satisfy your Crowdsale's requirements
  - With Token Emission, Validation and Distribution
- Applicable to ICO*
- https://docs.openzeppelin.com/contracts/2.x

Using OpenZeppelin Contracts on the Remix IDE

General Steps:

1. Go to the OpenZeppelin Contracts website and download the latest version of the contracts.
2. Open the Remix IDE in your browser and create a new file by clicking on the "+" button.
3. In the new file, import the OpenZeppelin Contracts by adding the following line at the top of your file:

```
import "https://github.com/OpenZeppelin/openzeppelin-contracts/contracts/token/ERC20/SafeERC20.sol";
```

4. You can also import other contracts from the OpenZeppelin library as needed for your smart contract.
5. Write your smart contract code, making sure to inherit from the imported OpenZeppelin Contracts as necessary.
6. Once you have written your code, you can use the Remix IDE's built-in compiler to compile and test your smart contract. You can also use the Remix IDE's built-in test functionality to test your smart contract's functionality.
7. Once you are satisfied with the functionality and security of your contract, you can deploy it to the Ethereum blockchain using a compatible wallet or blockchain explorer.

Conclusion