**The CEI method - Checks, Effects, Interactions**

**The Checks-Effects-Interactions (CEI) Pattern**

A very important thing to note. When developing this contract Patrick is using a style called Checks-Effects-Interactions or CEI.

The Checks-Effects-Interactions pattern is a crucial best practice in Solidity development aimed at enhancing the security of smart contracts, especially against reentrancy attacks. This pattern structures the code within a function into three distinct phases:

* Checks: Validate inputs and conditions to ensure the function can execute safely. This includes checking permissions, input validity, and contract state prerequisites.
* Effects: Modify the state of our contract based on the validated inputs. This phase ensures that all internal state changes occur before any external interactions.
* Interactions: Perform external calls to other contracts or accounts. This is the last step to prevent reentrancy attacks, where an external call could potentially call back into the original function before it completes, leading to unexpected behavior. (More about reentrancy attacks on a later date)

Another important reason for using CEI in your smart contract is gas efficiency. Let's go through a small example:

function coolFunction() public {

sendA();

callB();

checkX();

checkY();

updateM();

}

In the function above what happens if checkX() fails? The EVM goes through a function from top to bottom. That means it will execute sendA() then callB() then attempt checkX() which will fail, and then all the things need to be reverted. Every single operation costs gas, we pay for everything, and we just performed 2 operations, to revert at the 3rd. From this perspective isn't the following more logical?

function coolFunction() public {

// Checks

checkX();

checkY();

// Effects

updateStateM();

// Interactions

sendA();

callB();

}

First, we do the checks, if something goes bad we revert, but we don't spend that much gas. Then, if checks pass, we do effects, and all internal state changes are performed, these usually can't fail, or if they fail they spend an amount of gas that we can control. Lastly, we perform the interactions, here we send the tokens or ETH or perform external calls to other contracts. We wouldn't want these to happen in the absence of the checks or the state update so it's more logical to put them last.