

How does error handling work in Solidity?

Solidity is an object-oriented programming language for implementing smart contracts on blockchains like Ethereum.

It uses state-reverting exceptions to handle errors, undoing changes made to the state in the current call and flagging an error to the caller.

Solidity ensures atomicity as a property by reverting state changes in smart contract calls when errors occur.

Developers can directly interact with other contracts by declaring interfaces.

On the Ethereum blockchain, transactions are atomic, meaning they are either fully complete or have no effect on state and are reverted entirely.

Three main Solidity error handling functions?

assert : Assert is a crucial function in programming to check for code that should never be false, preventing impossible scenarios. If it returns a true value, a terminal bug will be displayed and programs will not execute. Unlike require and revert functions, assert consumes gas supply before reversing the program to its original state. Prior to the Byzantium fork, both functions behaved identically, but compiled to distinct opcodes.

require : The require function is a gate check modifier that checks inputs and conditions before execution. It acts as a gate check, preventing logic from accessing the function and producing errors. Require statements declare prerequisites for running the function, which must be satisfied



before code execution. The function accepts a single argument and returns a boolean value of true or false. If the execution is terminated due to a false condition, unused gas is returned to the caller and the state is reversed to the original state. Custom string messages can also be added.

revert : Revert is a function that does not evaluate conditions or depend on states or statements. It handles error types like require but is more suitable for complex logic gates. When called, unused gas is returned, and the state reverts to its original state. Revert also allows adding custom messages, similar to the require function.

Require vs. Revert vs. Assert

Require:

- Used at the beginning of a function
- Validates against illegal input
- Verifies state conditions prior to execution
- Refunds leftover gas

Revert:

- Identical to require
- Useful for more complex logic flow gates (i.e., complicated if-then blocks)
- Refunds leftover gas

Assert:

- Used at the end of a function
- Validates something that is impossible
- Critical for static code analysis tools
- Does not refund leftover gas



YouTube Link:



https://www.youtube.com/embed/InmVj4G_seY

Solidity Source File

