Errors in Solidity provide a convenient and gas-efficient way to explain to the user why an operation failed. They can be defined inside and outside of contracts (including interfaces and libraries).

They have to be used together with the [revert statement](https://docs.soliditylang.org/en/v0.8.15/control-structures.html#revert-statement) which causes all changes in the current call to be reverted and passes the error data back to the caller.

Errors cannot be overloaded or overridden but are inherited. The same error can be defined in multiple places as long as the scopes are distinct. Instances of errors can only be created using revert statements.

 Note that an error can only be caught when coming from an external call, reverts happening in internal calls or inside the same function cannot be caught.

*// SPDX-License-Identifier: GPL-3.0*

**pragma solidity** ^**0.8.4**;

*/// Insufficient balance for transfer. Needed `required` but only*

*/// `available` available.*

*/// @param available balance available.*

*/// @param required requested amount to transfer.*

error InsufficientBalance(uint256 available, uint256 required);

**contract** **TestToken** {

mapping(address => uint) balance;

function transfer(address to, uint256 amount) public {

if (amount > balance[**msg.sender**])

revert InsufficientBalance({

available: balance[**msg.sender**],

required: amount

});

balance[**msg.sender**] -= amount;

balance[to] += amount;

}

*// ...*

}

**te**

It is possible for a contract to revert with different errors of the same name or even with errors defined in different places that are indistinguishable by the caller. For the outside, i.e. the ABI, only the name of the error is relevant, not the contract or file where it is defined.

The statement require(condition, "description"); would be equivalent to if (!condition) revert Error("description") if you could define error Error(string). Note, however, that Error is a built-in type and cannot be defined in user-supplied code.

Similarly, a failing assert or similar conditions will revert with an error of the built-in type Panic(uint256).