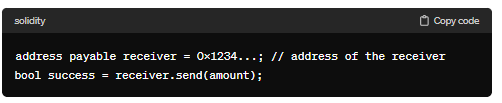
**Send Ether Send, Transfer, call:**

In Solidity, there are several ways to send ether from one address to another, each with its own characteristics and use cases:

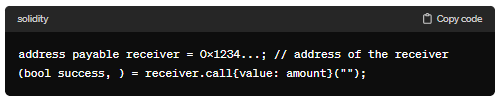
**send**: The **send** method is used to transfer ether to another address. It returns a boolean value indicating whether the transfer was successful or not. It is considered less secure because it only forwards 2300 gas to the receiving contract, which might not be enough to execute complex operations in the receiving contract.



**transfer**: The **transfer** method is similar to **send**, but it automatically forwards all available gas to the recipient. If the transfer fails (e.g., due to an out-of-gas error or if the recipient contract runs out of gas), it will revert the transaction.



**call**: The **call** method is the most flexible option and can be used to invoke arbitrary code on a contract. It allows you to specify the amount of ether to send and to provide data to the receiving contract. However, it's also the most complex and can be prone to security vulnerabilities if not used correctly.



It's important to consider the trade-offs between these methods when choosing which one to use. In most cases, **transfer** is the safest option for sending ether, as it automatically reverts the transaction if the transfer fails. However, if you need more control over gas usage or want to interact with the receiving contract in a specific way, **call** might be more suitable.

-> Send function we are using to send eth and in return we get the bool value. Limit : 2300 gas. In case if the gas limit above the defined one then in return we get the false. If the utilize gas is less then we are not getting the remaining gas.

-> Transfer function : it has the gas limit of 2300 gas. And we are not getting any revert back. It revert the change if the transaction is failed and there is not need to used require here. And remaining gas also revert back to us in this.

-> Call: In this gas limit we can define. In this return we get bool value and data in byte. Disadvantage is we need to use “require” as revert is not working here. Even remaining gas also we are not getting it back.

**Code:**

//SPDX-License-Identifier: GPL-3.0

pragma solidity ^0.8.0;

contract SendETH{

    //address payable public getter = payable();

    receive() external payable{}

    function checkbal() public view returns(uint){

        return address(this).balance;

    }

    event log(uint value);

    function SEND(address payable getter) public payable {

        emit log(msg.value);

        bool sent=getter.send(msg.value);

        require(sent,"tran is failed");

    }

    function TRANSFER(address payable getter) public payable {

        emit log(msg.value);

        getter.transfer(msg.value);

    }

    function CALL(address payable getter) public payable {

        emit log(msg.value);

        (bool sent,) = getter.call{value:msg.value}("");

        require(sent,"tran is failed");

    }

}

contract GetETH{

    receive() external payable{}

    function checkbal() public view returns(uint){

        return address(this).balance;

    }

}

**Output:**

