**Address**[**ℑ**](https://docs.soliditylang.org/en/v0.8.15/types.html?highlight=size#address)

The address type comes in two flavours, which are largely identical:

* address: Holds a 20 byte value (**size** of an Ethereum address).
* address payable: Same as address, but with the additional members transfer and send.

The idea behind this distinction is that address payable is an address you can send Ether to, while you are not supposed to send Ether to a plain address, for example because it might be a smart contract that was not built to accept Ether.

Type conversions:

Implicit conversions from address payable to address are allowed, whereas conversions from address to address payable must be explicit via payable(<address>).

### Members of Address Types[ℑ](https://docs.soliditylang.org/en/v0.8.15/units-and-global-variables.html#members-of-address-types)

**<address>.balance (uint256)**

balance of the [Address](https://docs.soliditylang.org/en/v0.8.15/types.html#address) in Wei

**<address>.code (bytes memory)**

code at the [Address](https://docs.soliditylang.org/en/v0.8.15/types.html#address) (can be empty)

**<address>.codehash (bytes32)**

the codehash of the [Address](https://docs.soliditylang.org/en/v0.8.15/types.html#address)

**<address payable>.transfer(uint256 amount)**

send given amount of Wei to [Address](https://docs.soliditylang.org/en/v0.8.15/types.html#address), reverts on failure, forwards 2300 gas stipend, not adjustable

**<address payable>.send(uint256 amount) returns (bool)**

send given amount of Wei to [Address](https://docs.soliditylang.org/en/v0.8.15/types.html#address), returns false on failure, forwards 2300 gas stipend, not adjustable

**<address>.call(bytes memory) returns (bool, bytes memory)**

issue low-level CALL with the given payload, returns success condition and return data, forwards all available gas, adjustable

**<address>.delegatecall(bytes memory) returns (bool, bytes memory)**

issue low-level DELEGATECALL with the given payload, returns success condition and return data, forwards all available gas, adjustable

**<address>.staticcall(bytes memory) returns (bool, bytes memory)**

issue low-level STATICCALL with the given payload, returns success condition and return data, forwards all available gas, adjustable

in order to make safe Ether transfers, always check the return value of send, use transfer or even better: Use a pattern where the recipient withdraws the money.

It is possible to query the balance of an address using the property balance and to send Ether (in units of wei) to a payable address using the transfer function:

address payable x = payable(0x123);

address myAddress = address(this);

if (x.balance < 10 && myAddress.balance >= 10) x.transfer(10);

The transfer function fails if the balance of the current contract is not large enough or if the Ether transfer is rejected by the receiving account. The transfer function reverts on failure.

NOTE:

If x is a contract address, its code (more specifically: its [Receive Ether Function](https://docs.soliditylang.org/en/v0.8.15/contracts.html#receive-ether-function), if present, or otherwise its [Fallback Function](https://docs.soliditylang.org/en/v0.8.15/contracts.html#fallback-function), if present) will be executed together with the transfer call (this is a feature of the EVM and cannot be prevented). If that execution runs out of gas or fails in any way, the Ether transfer will be reverted and the current contract will stop with an exception.

* send

Send is the low-level counterpart of transfer. If the execution fails, the current contract will not stop with an exception, but send will return false.

**Warning**

There are some dangers in using send: The transfer fails if the call stack depth is at 1024 (this can always be forced by the caller) and it also fails if the recipient runs out of gas. So in order to make safe Ether transfers, always check the return value of send, use transfer or even better: use a pattern where the recipient withdraws the money.