# Object Ownership

Every object has an owner field that dictates how you can use it in transactions. Objects can have the following types of ownership:

An address-owned object is owned by a specific 32-byte address that is either an account address (derived from a particular signature scheme) or an object ID. An address-owned object is accessible only to its owner and no others.

Go to Address-Owned Objects .

An immutable object is an object that can't be mutated, transferred, or deleted. Immutable objects have no owner, so anyone can use them.

Go to Immutable Objects .

A shared object is an object that is shared using the 0x2::transfer::share_object function and is accessible to everyone. Unlike owned objects, anyone can access shared ones on the network.

Go to Shared Objects .

In Move, you can organize data structures by putting a field of struct type in another.

Go to Wrapped Objects .

## Address-owned

An address-owned object is owned by a specific 32-byte address that is either an account address (derived from a particular signature scheme) or an object ID. An address-owned object is accessible only to its owner and no others.

Go to Address-Owned Objects .

An immutable object is an object that can't be mutated, transferred, or deleted. Immutable objects have no owner, so anyone can use them.

Go to Immutable Objects .

A shared object is an object that is shared using the 0x2::transfer::share_object function and is accessible to everyone. Unlike owned objects, anyone can access shared ones on the network.

Go to Shared Objects .

In Move, you can organize data structures by putting a field of struct type in another.

Go to Wrapped Objects .

## Immutable

An immutable object is an object that can't be mutated, transferred, or deleted. Immutable objects have no owner, so anyone can use them.

Go to Immutable Objects .

A shared object is an object that is shared using the 0x2::transfer::share_object function and is accessible to everyone. Unlike owned objects, anyone can access shared ones on the network.

Go to Shared Objects .

In Move, you can organize data structures by putting a field of struct type in another.

Go to Wrapped Objects .

## Shared

A shared object is an object that is shared using the 0x2::transfer::share_object function and is accessible to everyone. Unlike owned

objects, anyone can access shared ones on the network.

Go to Shared Objects .

In Move, you can organize data structures by putting a field of struct type in another.

Go to Wrapped Objects .

## Wrapped

In Move, you can organize data structures by putting a field of struct type in another.

Go to Wrapped Objects .