

An abstract composition on a dark gray background. It features several diagonal lines: a white line in the upper left, a light gray line in the lower left, and a bright pink line in the lower right. The word 'Substrate' is written in a bold, orange, sans-serif font, positioned between the white and light gray lines. The word 'River' is written in the same font and color, positioned between the light gray and pink lines.

**Substrate**

**River**

The background is dark gray with several diagonal lines. Two white lines are in the upper left, and two pink lines are in the lower right, all running from the bottom-left towards the top-right.

# Kitties-003

## Owning Multiple Kitties

<https://substrate.dev/substrate-collectables-workshop/#/2/owning-multiple-kitties>



目的：解决的问题

1- 同一个人可以创建多个Kitty



## 1- Tuple

Tuple一个数据结构 (,) 类似一个二维数组

让我们来思考一个问题，如何构建一个朋友列表：

1- 一个人有多少个朋友？

2- 第N个朋友是谁？

```
MyFriendsArray get(my_friends_array): map (T::AccountId, u32) => T::AccountId;  
MyFriendsCount get(my_friends_count): map T::AccountId => u32;
```

```
MyFriendsArray[AccountId][Index] -> AccountId  
MyFriendsArray[AccountId].length()
```



You turn

## 1- Storage

```
// ACTION: Rename this to `OwnedKittiesArray`/`kitty_of_owner_by_index`  
//      Have the key be a tuple of (T::AccountId, u64)  
OwnedKitty get(kitty_of_owner): map T::AccountId => T::Hash;  
// ACTION: Add a new storage item `OwnedKittiesCount` which is a `map` from `T::AccountId` to `u64`  
// ACTION: Add a new storage item `OwnedKittiesIndex` which is a `map` from `T::Hash` to `u64`
```

## 2- Module

```
// ACTION: Generate variables `owned_kitty_count` and `new_owned_kitty_count`  
//      similar to `all_kitties_count` below
```

```
// ACTION: Update this to maintain the state of our new storage items  
<OwnedKitty<T>>::insert(&sender, random_hash);
```



## 异常信息

```
<OwnedKittiesArray<T>>::insert(&sender, owned_kitties_count), random_hash);  
|                               ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^ the trait  
`core::borrow::Borrow<(<T as srml_system::Trait>::AccountId, u64)>` is not  
implemented for `(&<T as srml_system::Trait>::AccountId, u64)`
```



## 视频备注

元组概念解释的比较少，可以查询Rust中的Tuple



**Substrate**

**River**

**Thanks**