

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 extending from the bottom-left towards the top-right.

# Kitties-003

## Tracking All Kitties

<https://substrate.dev/substrate-collectables-workshop/#/2/tracking-all-kitties>



## 1- Creating a List

在运行时开发中，列表迭代通常是危险的。我们将使用mapping和计数器来模拟列表。

people列表

```
decl_storage! {  
    trait Store for Module<T: Trait> as Example {  
        AllPeopleArray get(person): map u32 => T::AccountId;  
        AllPeopleCount get(num_of_people): u32;  
    }  
}
```

```
count 3  
PeopleArray  
[0,0001], [1,0002],[2,0003]
```



### 3- Updating our List in Storage

people列表

```
fn add_person(origin, new_person: T::AccountId) -> Result {  
    let sender = ensure_signed(origin)?;  
  
    let all_people_count = Self::num_of_friends();  
  
    let new_all_people_count =  
all_people_count.checked_add(1).ok_or("Overflow adding a new person");  
  
    <AllPeopleArray<T>>::insert(all_people_count, new_person);  
    <AllPeopleCount<T>>::put(new_all_people_count);  
  
    Ok()  
}
```

+1

count 4

PeopleArray

[0,0001], [1,0002],[2,0003],[3,0003]



## 如何实现

- 1- 创建了kitty的数量
- 2- 查询创建的Kitty的记录: index和kitty\_id的对应关系
- 3- Kitty\_id和index的关系



## 2- Checking for Overflow/Underflow

AllPeopleCount

checked\_add()

checked\_sub()

```
let all_people_count = Self::num_of_people();
```

```
let new_all_people_count = all_people_count.checked_add(1).ok_or("Overflow adding a new person")?;
```

```
let new_all_people_count = match all_people_count.checked_add(1) {  
    Some (c) => c,  
    None => return Err("Overflow adding a new person"),  
};
```



## You turn

### 1- Storage

```
// ACTION: Create new storage items to globally track all kitties:  
//      - `AllKittiesArray` which is a `map` from `u64` to `T::Hash`, add a getter function for this  
//      - `AllKittiesCount` which is a `u64`, add a getter function for this  
//      - `AllKittiesIndex` which is a `map` from `T::Hash` to `u64`
```

### 2- create\_kitty

```
// ACTION: Get the current `AllKittiesCount` value and store it in `all_kitties_count`  
// ACTION: Create a `new_all_kitties_count` by doing a `checked_add()` to increment  
`all_kitties_count`  
//      REMINDER: Return an `Err()` if there is an overflow
```

### 3- create\_kitty

```
// ACTION: Update the storage for the global kitty tracking  
//      - `AllKittiesArray` should use the `all_kitties_count` (remember `index` is `count - 1`)  
//      - `AllKittiesCount` should use `new_all_kitties_count`  
//      - `AllKittiesIndex` should use `all_kitties_count`
```

The background is dark gray with several diagonal lines. A white line starts from the left edge and goes towards the top right. A shorter white line is parallel to it, also sloping upwards. A gray line starts from the bottom left and goes towards the top right. A red line starts from the bottom left and goes towards the top right, positioned below the gray line. The text 'Substrate' is in orange, bold font, positioned between the two white lines.

**Substrate**

**River**

**Thanks**