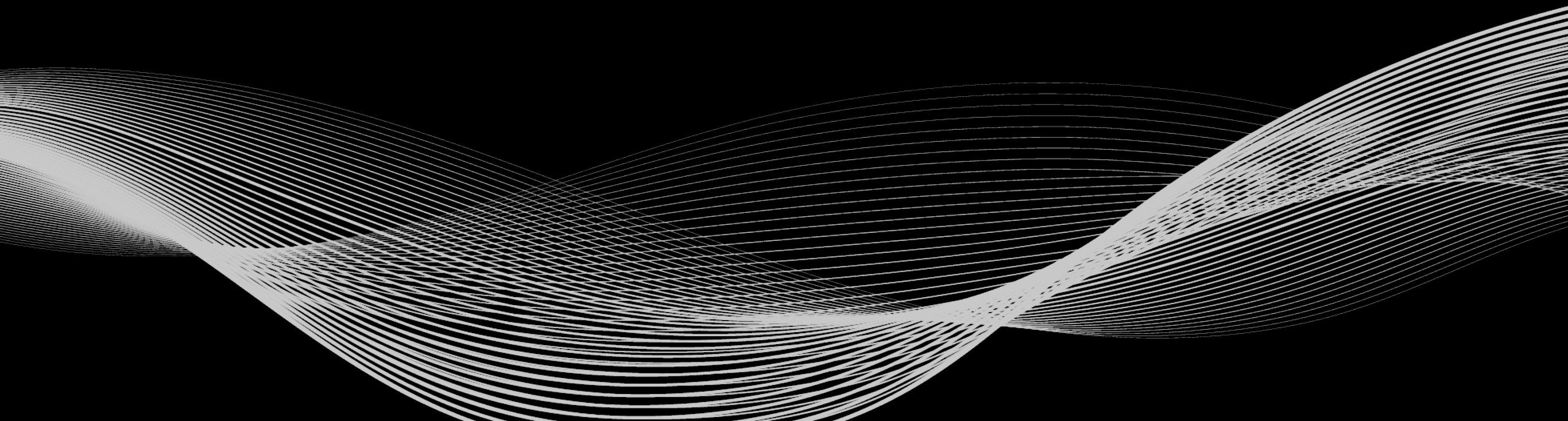


# Substrate 快速入门与实战

---

讲师: Bryan





# 第五节课



# 链表数据结构



## 链表数据结构

- KittyLinkedItem
  - prev: Option<KittyIndex>
  - next: Option<KittyIndex>
- OwnedKitties: map (AccountId, Option<KittyIndex>) => Option<KittyLinkedItem>



## 链表数据结构

- Substrate 自带了 `linked_map` 就是使用链表实现的
- `pub Validators get(validators): linked_map T::AccountId  
=> ValidatorPrefs<BalanceOf<T>>;`
- 提供了遍历所有键值得功能
- 需要额外的存储空间



## 链表数据结构

- 双向遍历
- $O(1)$  添加删除元素
- $O(n)$  随机下标读写
- `head.next -> item1.next -> item2.next -> None`
- `head.prev -> item2.prev -> item1.prev -> None`



## 链表数据结构

- 空表: []
  - head: { prev: head, next: head }
- 添加元素 `item1`: [item1]
  - head: { prev: item1, next: item1 }
  - item1: { prev: head, next: head }



## 链表数据结构

- 添加元素 `item2`: [item1, item2]
  - head: { prev: item2, next: item1 }
  - item1: { prev: head, next: item2 }
  - item2: { prev: item1, next: head }
- 添加元素 `item3`: [item1, item2, item3]
  - head: { prev: item3, next: item1 }
  - item1: { prev: head, next: item2 }
  - item2: { prev: item1, next: item3 }
  - item3: { prev: item2, next: head }





## 链表数据结构

- 删除元素 `item2`: [item1, item3]
  - head: { prev: item3, next: item1 }
  - item1: { prev: head, next: item3 }
  - item3: { prev: item1, next: head }
- 删除元素 `item3`: [item1]
  - head: { prev: item1, next: item1 }
  - item1: { prev: head, next: head }
- 删除元素 `item1`: []
  - head: { prev: head, next: head }



# 单元测试



# polkadot.js 添加自定义 类型



# srml-balances 代码分析



# 作业



## 作业

- 完成 transfer
- 完成 insert\_owned\_kitty
- 设计加密猫模块 V4
- 交易所
  - 给自己的小猫设定价钱
  - 购买小猫



## 额外作业

- 利用 polkadot.js 开发一个命令行软件
  - 创建小猫
  - 赠予小猫



一块链习

**THANK YOU!**

**Contact us:**  
**[info@yikuailianxi.com](mailto:info@yikuailianxi.com)**

