

Sui 101

Introduction to Sui

対象：学生、Web2開発者、暗号資産に興味がある人



学習目標

- ✓ Suiとブロックチェーンの基本を理解している
- ✓ ネットワークとやり取りするためにSlushを準備する
- ✓ SuiブロックチェーンでNFTを発行する

Agenda

- ブロックチェーンとは何か、Suiの仕組み
- Suiのオブジェクト中心モデルの特徴
- 基本的なウォレット設定、Move、トランザクション

Resources

- docs.sui.io
- [Slush](#)
- [Sui Developer Forum](#)
- [Sui Move Intro Course](#)



Suiとは?

- 高速処理と高いスケーラビリティを実現するレイヤー1ブロックチェーン
- 資産とオブジェクトを中心にしたアーキテクチャ
- Move言語を用いたスマートコントラクト開発が可能



Suiの特徴とは？

オブジェクト中心モデル

- Suiブロックチェーン上の資産は「オブジェクト」と呼ばれる
- 各オブジェクトは固有のIDを持ち、データを保存したり変更したりできる
- このモデルはアカウントベースのブロックチェーンとは異なる



ブロックチェーンとは？

- 分散型台帳 (Distributed Ledger Technology) を用いる
- 改ざんできない透明な取引記録を実現
- ネットワーク上の多数のコンピュータに取引履歴を分散保存



Move言語の紹介

Move言語は、安全性と柔軟性を備えたスマートコントラクトのプログラミング言語です。

Docs:

<https://docs.sui.io/concepts/sui-move-concepts>

SuiにおけるNFTとFTの比較

NFT (Non-Fungible Token)

- 固有のIDを持つオンチェーン上のユニークなオブジェクト
- 名前、説明、画像など豊富なメタデータを保持可能
- 各NFTは独立したオブジェクトとして存在

FT (Fungible Token)

- お金のように同じ価値を持つ単位
- Suiでは「Coin型」で表現される
- 分割したりまとめたりできる
- 特性をもつ

Move言語によるNFT の実装例

1. `key` アビリティを持つ構造体
2. メタデータを含む固有のオブジェクトを定義
3. 世界に一つだけのNFTをミント可能

```
module workshop::nft;

use std::string;
use sui::url;

public struct MyNFT has key, store {
    id: UID,
    name: string::String,
    url: url::Url,
}

public entry fun mint(name: vector<u8>, url_bytes:
vector<u8>, ctx: &mut TxContext) {
    let nft = MyNFT {
        id: object::new(ctx),
        name: name.to_string(),
        url: url::new_unsafe_from_bytes(url_bytes),
    };
    transfer::public_transfer(nft, ctx.sender());
}
```

Move言語によるFTの実装例

1. Witnessを使ってコインの種類を定義
2. コインは分割や結合が可能
3. Treasury Capによってミント権限を管理

```
module workshop::my_coin;

use sui::coin::{Self, TreasuryCap};

public struct MY_COIN has drop, store {}

fun init(witness: MY_COIN, ctx: &mut TxContext) {
    let (treasury_cap, metadata) =
coin::create_currency<MY_COIN>(
    witness,
    6,
    b"MYC",
    b"My Coin",
    b"",
    option::none(),
    ctx,
);
transfer::public_freeze_object(metadata);
transfer::public_transfer(treasury_cap, ctx.sender());
}

public entry fun mint(
    treasury_cap: &mut TreasuryCap<MY_COIN>,
    amount: u64,
    recipient: address,
    ctx: &mut TxContext,
) {
    let coin = coin::mint(treasury_cap, amount, ctx);
    transfer::public_transfer(coin, recipient);
}
```

Slush Walletを設定



Slushをインストール

<https://slush.app/>



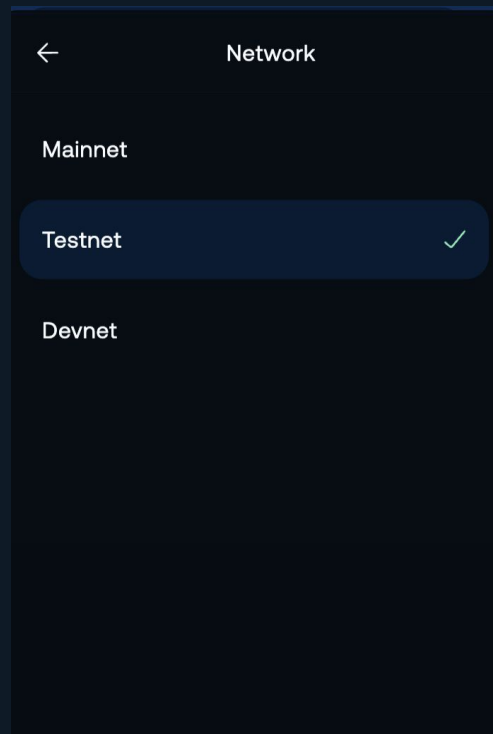
Seed Phraseを
バックアップする



テストネットSUIを
請求する

Slushでテストネットに 接続する

1. 画面右下をタップしてウォレットをバックアップしたら、設定を開く
2. "Network"をタップ
3. "Testnet"を選択する



SUIをFaucetサイトから手に入れる

1. Suiのテストネットfaucetサイトにアクセス
<https://faucet.sui.io/>
2. アドレスをペーストする
3. テストネットSUIをリクエスト

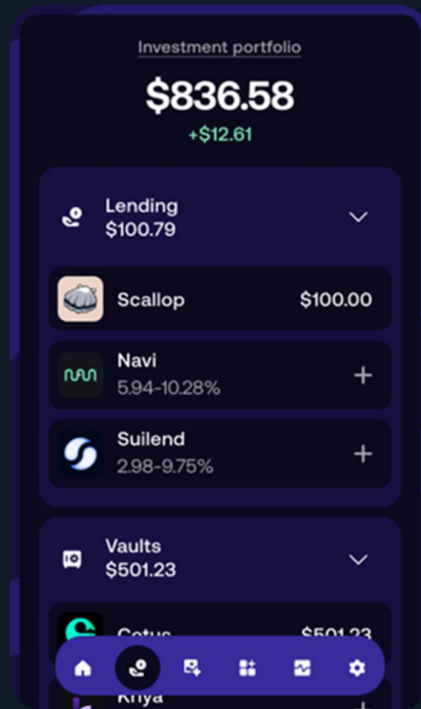


ウォレットの操作画面を 探検する

1. SUIのトークンバランスを確認できる
2. ミントしたNFTを確認できる

1

2



基本的なトランザクション: NFTのミント, トークンを送る

- SuiテストネットでサンプルNFTをMINTしてみよう
 - <https://nft-mint-sample.vercel.app/>
- 他のウォレットにトークンを送ってみよう

Suiでのガス代について

1. すべてのトランザクションにはガス代としてSUIが必要
2. ガス代の用途:
 - a. 計算処理
 - b. オブジェクトの保存
3. テストネット用SUIは無料だが上限あり
4. デプロイ前に必ず残高を確認すること

Doc:

<https://docs.sui.io/concepts/tokenomics/gas-in-sui>

Gas & Storage Fees

Gas Payment	0x163a...1bb5
Gas Budget	0.010311 sui (10,311,000 MIST)
Total Gas Fee	0.000999835 sui (999,835 MIST) ▾

*SuiSui network explorer*に表示される
トランザクションブロックのガス料金セクション

Sui CLIとは?

Sui CLIは、Suiネットワークとやり取りするためのコマンドラインツールです。

Docs: <https://docs.sui.io/references/cli/client>

Sui CLIのインストール

macOS

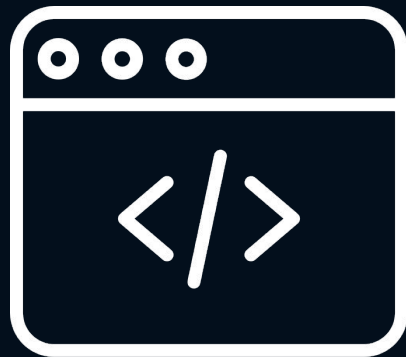
```
brew install sui
```

Windows

```
choco install sui
```

Linux

```
brew install sui
```



設定の確認・テストネットへの切り替え

設定の確認

(正常にインストールされていれば、このコマンドで Suiバイナリのバージョンが表示されます)

```
sui --version
```

Testnet環境に切り替える

```
sui client switch --env testnet
```

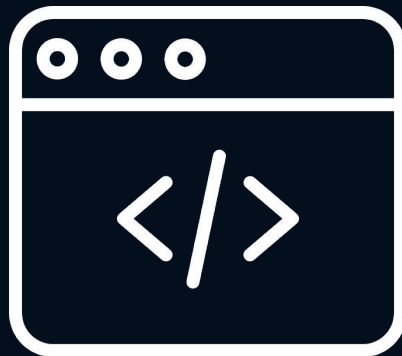
Sui CLIでアドレスのインポート

Slushで作ったアドレスをインポート

```
sui keytool import INPUT KEY_SCHEME
```

インポートしたアドレスが有効になっているか確認

```
sui client active-address
```

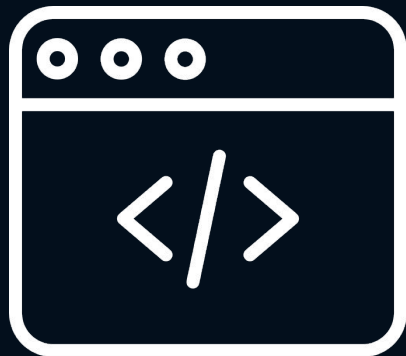


Docs:<https://docs.sui.io/references/cli/cheatsheet>

Sui CLIでウォレットの中身を確認

アクティブアドレスにガス代があるか確認

```
sui client gas
```



Docs:<https://docs.sui.io/references/cli/cheatsheet>

Packageとは？

- 1つ以上のモジュールをまとめた単位
- Suiブロックチェーン上でデプロイ・管理される
- バージョン管理が可能

例：NFTマーケット用の一連のモジュールを含むパッケージ

EXPLORER

 0x65d4...df64



MOVEMODULE

0x65d5...37dc



Moduleとは？

- Move言語で書かれたコードの再利用可能な単位
- 関数や構造体などの定義をまとめたもの
- スマートコントラクトのロジックを実装する場所

例：NFTをミントする関数、コインを転送する関数などを含む

EXPLORER

 0x65d4...df64



MOVEMODULE

0x65d5...37dc



Moduleを確認する方法

1. Sui Explorer というサイト

<https://suiscan.xyz/testnet/home>

2. Move moduleを検索する

a. サンプルを見てみよう:

0x9f992cc2430a1f442ca7a
5ca7638169f5d5c00e0ebc
3977a65e9ac6e497fe5ef

EXPLORER

 0x65d4...df64



MOVEMODULE

0x65d5...37dc



QuizとハンズオンChallenge

Mini Quiz

- Suiのオブジェクトモデルと
は何でしょうか？
- MoveはSolidityとはどのよ
うに違うでしょうか？



Challenge

- NFTのスマートコントラクトをデプロイしてみよう！
- スマートコントラクトの関数を呼び出してNFTをミントしてみよう！

Repo:

https://github.com/pnha2411/mint_nft

スクリーンショットを提出して下さい

セットアップガイド： NFTのスマートコントラクトのデプロイ

1/ リポジトリをクローンする

```
git clone https://github.com/pnha2411/mint_nft
```

2/ ディレクトリに移動

```
cd move/nft-move
```

3/ Testnetにデプロイ

```
sui client publish --gas-budget 1000000000
```

よくある質問

- Suiウォレットの復元手順はどうなっていますか？
- SuiウォレットをdAppsに接続する方法は？
- Moveについてさらに学べるリソースはどこで見つけられますか？
- このコースは本当に初心者向けですか？

Next: Builderトラックの概要

- コントラクトの開発とデプロイを中心に学習
- ガス代やストレージの仕組みを理解する

•

Thank You.

•