TinTin

以太坊开发入门实战

owen liu from DODO

课程大纲

TinTin

第一节： 从与DApp应用交互开始，认识以太坊

第二节： 解刨合约交易，入门Solidity

第三节： **Solidity开发实战**

第四节： 源码分析与合约安全

第五节： 链上数据记录与检索

第六节： 前端与合约的交互开发

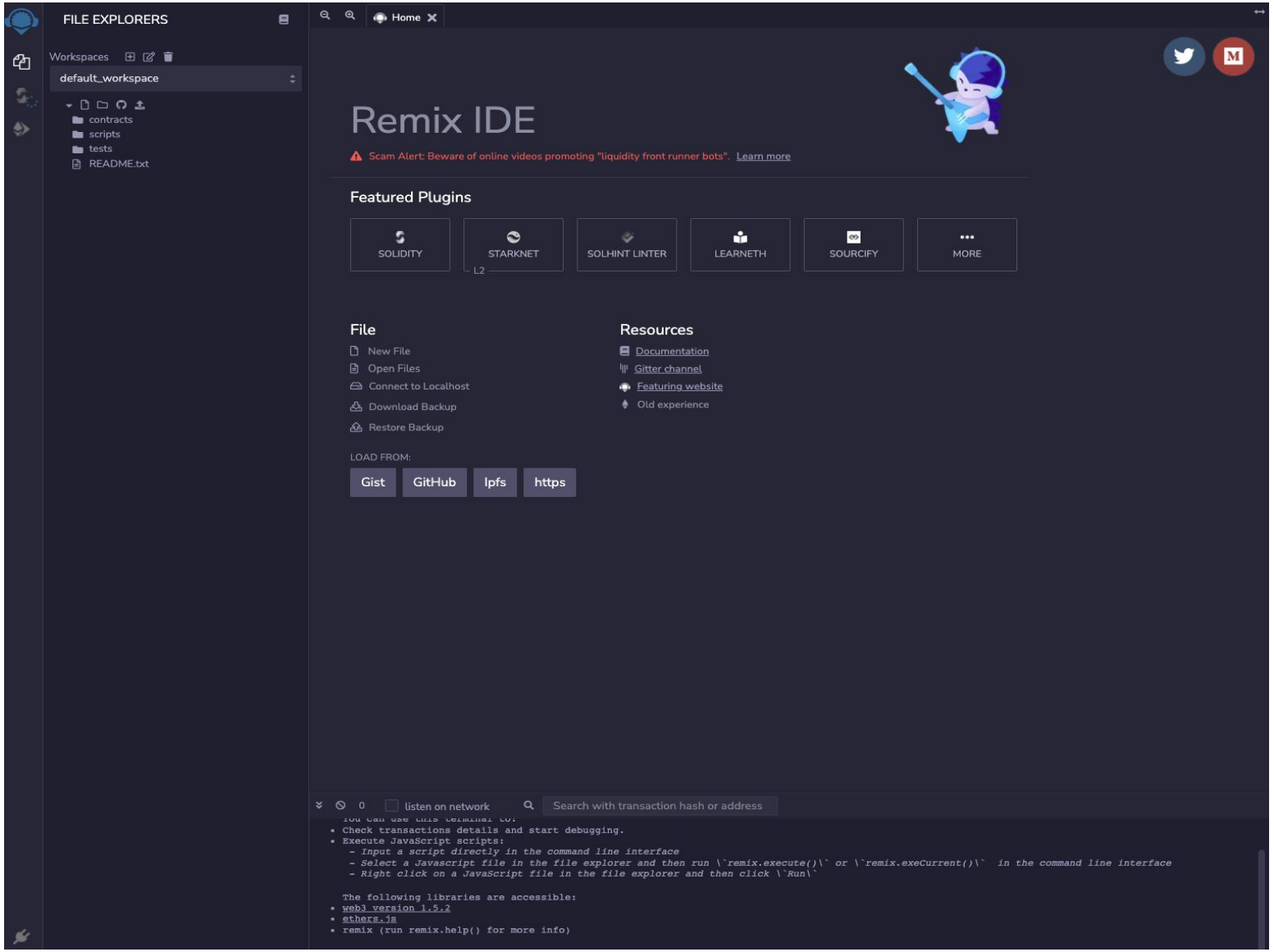
第七节： 经典业务场景的合约解析

第三节：Solidity开发实战

TinTin

Remix （http://remix.ethereum.org/）

Remix是以太坊官方开源的Solidity在线集成开发环境，可以在网页内完成智能合约的开发、编译、部署、调试与交互



TinTin

值类型

- 布尔类型　　bool true or false
- 整型:　　int/uint (int256/uint256) 8 到256 位
- 定长浮点型　　Solidity没有完全支持该类型
- 地址类型　　20字节，账户地址大小　　=》　　地址类型成员变量
- 定长字节数组　　bytes1，bytes2.....
- 变长字节数组　　bytes，string
- 枚举类型　　enum ActionChoices { GoLeft, GoRight, GoStraight, SitStill }
- 函数类型

function (<parameter types>) {internal|external|public} [pure|view|payable] [returns (<return types>)]

TinTin

地址类型

- balance　　作为属性，直接查询地址下的余额

- transfer　　作为函数，向该地址发送ether

```
address payable x = payable(0x123);
address myAddress = address(this);
if (x.balance < 10 && myAddress.balance >= 10) x.transfer(10);
```

- call　　外部合约调用（未在合约声明外部合约的interface，通过ABI编码的方式调用）

```
bytes memory payload = abi.encodeWithSignature("register(string)", "MyName");
(bool success, bytes memory returnData) = address(nameReg).call(payload);
require(success);
```

同时可以设置 gas与value

address(nameReg).call{gas: 1000000, value: 1 ether}(abi.encodeWithSignature("register(string)", "MyName"));

- delegatecall　　类似于call，区别在于delegatecall在自身合约环境执行外部合约逻辑，call是在外部合约环境执行逻辑

- staticcall　　类似于call，但当调用的外部合约函数修改了状态，就会回滚失败

call，delegatecall，staticcall 是非常底层的函数，只当做最后的办法使用

TinTin

- mapping（keyType => valueType） ,视作哈希表，不同的是，mapping不存储key，因此没有长度以及key或value的集合概念

- 数组　　bytes string 是特殊的数组类型
  length、push()、push(x)、pop

- 结构体　　可以作为元素用在数组或mapping中

- 数据位置　　关键字 storage（存储） or memory　（内存）。

  - 函数参数（包括返回参数）默认是memory，局部数组或结构体变量默认是storage，合约状态变量强制是storage

  - storage 与 memory 之间两两赋值，storage向状态变量赋值创建独立拷贝
  - 状态变量向storage 赋值传递引用
  - memory 向 memory赋值不会创建拷贝

  - 只有状态变量可以用mapping类型

TinTin

单位

- 以太币单位： 1 wei == 1，1 gwei = 1e9， 1 ether = 1e18

- 时间单位：秒是默认时间单位， 1 == 1 seconds （seconds，minutes，hours，days，weeks）

全局变量 - *详见文档*

- 区块与交易相关：
    - blockhash(uint blockNumber) returns(bytes32) 最近前256区块，超出默认返回0
    - block.number 区块高度
    - block.timestamp 当前区块出块时间戳
    - msg.data 上链交易的data
    - msg.value 交易发送的以太数量
    - msg.sender 当前调用的账户地址（合约or外部账户）
    - tx.origin 交易的发起账户地址

注：对于每一个外部合约调用，msg.sender, msg.value, msg.data 都会变化

- 错误处理
    - assert(bool condition)
    - require(bool condition)、require(bool condition, string memory message)
    - revert()、revert(string memory reason)

第三节：Solidity开发实战

TinTin

合约

- Getter 函数：为public状态变量自动创建。内部访问被认为是状态变量，外部访问被认为是函数
- 函数修饰器 modifier：在函数执行之前自动检查某个条件
- constant && immutable 变量： constant 编译时确定，immutable 合约创建初始化确定
- 特殊函数
    - constructor() public
    - receive() external payable {}
    - fallback() external
- 继承、抽象合约、接口
- Library：若通过delegatecall 调用，只需部署一次，无状态。若直接调用，函数为view or pure

课后扩展阅读：

- https://docs.soliditylang.org/en/develop/types.html
- https://docs.soliditylang.org/en/develop/units-and-global-variables.html
- https://docs.soliditylang.org/en/develop/contracts.html
- https://docs.soliditylang.org/en/develop/cheatsheet.html

第三节课作业：

- 编写众筹以太坊合约，提供用户参与函数，配置多众筹活动函数，提款函数。并部署开源在Rinkeby测试网

TinTin

TinTin小助手

TinTin公众号

Twitter | YouTube | Discord