



ScaleBit

TON&Tact介绍

Jason



个人介绍

- Jason
- ScaleBit安全研究员
- 智能合约审计、漏洞挖掘、产品研发
- 审计项目数量50+



大纲

- TON生态介绍
- Tact语言介绍
- 课程学习目标
 - 完成一个Web版割草游戏
- Tact开发环境搭建
- Helloworld演示
 - 账户介绍
 - 转账
 - 合约编译部署调用
- 总结
- 作业



TON生态介绍

- Telegram于 2018 年创建了 Telegram Open Network
- ICO 获得 17 亿美元的代币销售
- SEC 命令 Telegram 停止销售之前与 TON 区块链相关的代币 \$GRAM
- 2020 年 5 月, Telegram宣布终止参与区块链的开发, 并开始向早期投资者发放退款
- 在 Telegram 放弃 TON 项目后, 一群社区开发人员将其重新命名为 The Open Network
- Telegram App整合自托管钱包“Ton Space”, 支持TON钱包
- 2023年 9 月, Telegram 已认定选择 TON 网络作为其 Web3 基础设施的区块链网络, TON 生态的多个项目和进展都将基于 Telegram 进行构建
- TON 背靠 Telegram 的 8 亿多用户支持, 有着类似微信的发展前景

Explore 551 apps in TON Ecosystem

[Exchanges CEX](#)[Staking](#)[Wallets](#)[Explorers](#)[Bridges](#)[Utilities](#)[Channels](#)[NFT Collections](#)

#1 Online Casino on TON

@Whale a gaming platform that let's you play most popular games and bet on all sports ...

[View app](#)

News

[See all](#)

- TON processed over 80 million inscriptions transactions in 15 days
TON Community is delighted to se
- Introducing fair launches in the TON Ecosystem with Ton Raffles Ton Raffles has pioneered a new er
- Layerswap integrates TON
Layerswap has integrated TON, paving the way for users to access...
- HackenProof launches new bug

Promoted Apps

[Add project](#)[Jackpot.ton](#)[Playmuse Marketplace](#)[Bitsler.com](#)[Crypto Ads Platform](#)[JetTon Games](#)

Toncoin

\$2.22 -0.92% • 24h

Market Cap

\$7.66B -11.87% • 24h



Tact语言介绍

TON 生态系统主要使用两种智能合约编程语言: FunC 和 Tact。

- FunC 是一种低级语言, 专为深入了解 TON 架构的开发人员而设计, 它在开发复杂的多合约系统时可能具有挑战性。
- Tact 是 TON 区块链的一种新编程语言, 注重效率和简单性。它的设计易于学习和使用, 并且非常适合智能合约。Tact 是一种静态类型语言, 具有简单的语法和强大的类型系统。
- Tact 提供了受 JavaScript 和 Typescript、Rust 和 Swift 启发的熟悉语法。对于新开发人员来说, 代数数据类型和编译时执行等强大功能看起来有机且友好。
- 编译器会把 Tact 代码转译为 FunC 代码, 最终编译为类似 Fift 的字节码, 在 TON VM 虚拟机上执行。

TACT

UNLEASHING TON'S POWER WITH SAFE AND SCALABLE SMART CONTRACTS

Tact code

Write Tact code below

```
1 message (0x123123) TransferMsg {
2   to: Address;
3   text: String;
4 }
5
6 contract SimpleContract {
7   init() {}
8   receive() {}
9   receive(msg: TransferMsg) {
10     send(SendParameters{
11       to: msg.to,
12       value: 0,
13       mode: SendRemainingValue,
14       body: msg.text.asComment()
15     });
16   }
17 }
```

FunC result

Tact compiler will turn the program to FunC code on the fly

```
1 include "imports/stdlib.fc";
2
3 int op::transfer_coins() asm "0x123123 PUSHINT";
4 () recv_internal(int my_balance, int msg_value, cell i
5   if (in_msg_body.slice_empty?()) { ;; ignore empty me
6     return ();
7   }
8
9   slice cs = in_msg_full.begin_parse();
10  int flags = cs~load_uint(4);
11  if (flags & 1) { ;; ignore all bounced messages
12    return ();
13  }
14
15  int op = in_msg_body~load_uint(32);
16  if(op == op::transfer_coins()) {
17    slice to = in_msg_body~load_msg_addr();
18    cell text = in msa bodv~load_ref();
19  }
```



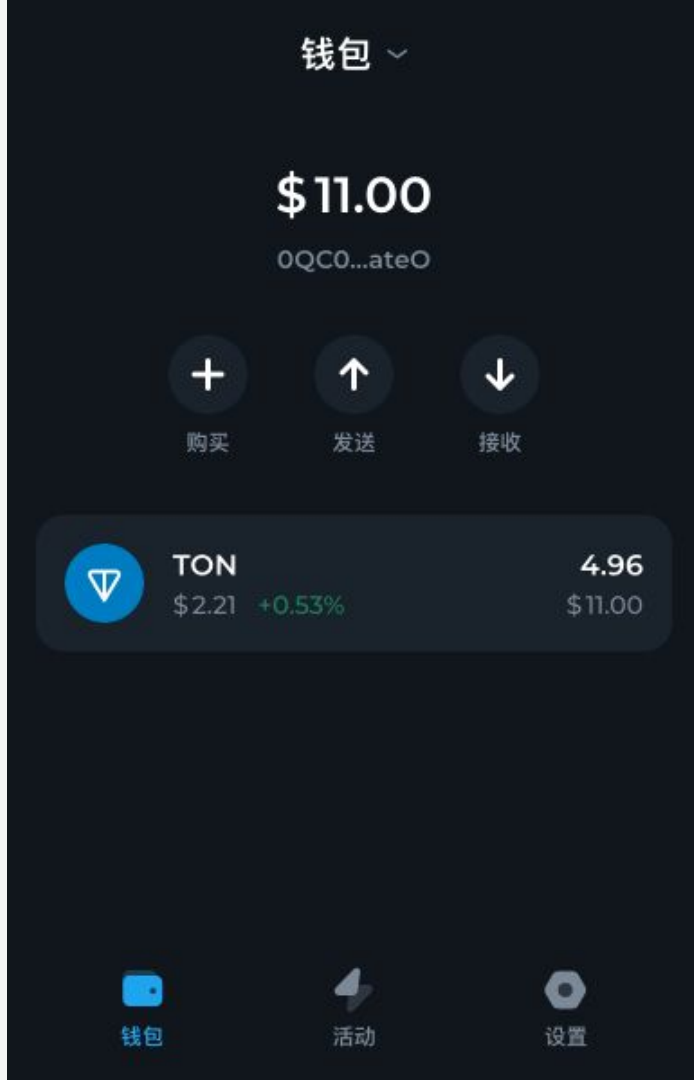

课程学习目标

课程总共6节课, 会学习Tact语法、Jetton和NFT标准, 完成一个Web版割草游戏



Tact开发环境搭建

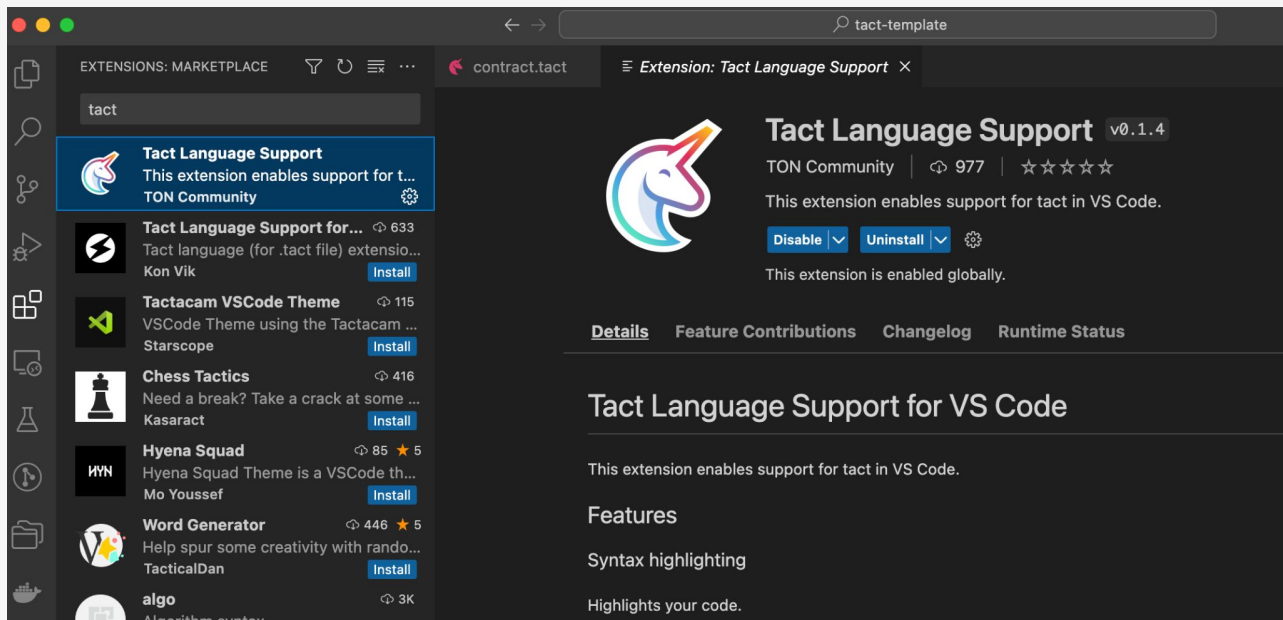
- 安装TON钱包
- 创建区块链账户
- 配置测试区块链网络
- 申领测试币





Tact开发环境搭建

- 安装Visual Studio Code插件Tact Language Support



Helloworld 合约代码

合约说明

- 从模版工程开始
 - git [git@github.com:tact-lang/tact-template](https://github.com/tact-lang/tact-template).git
- 一个示例合约SampleTactContract
- 具备状态: owner和counter
- 初始化函数: init
- 合约方法: add
- 消息处理函数: 可处理两种类型消息
- getter方法: counter

```
contract.tact ×
sources > contract.tact
Howard Peng, 4 months ago | 2 authors (Steve Korshakov and others)
1  import "@stdlib/deploy";
2
3  message Add {
4      amount: Int as uint32;
5  }
6
7  contract SampleTactContract with Deployable {
8
9      owner: Address;
10     counter: Int as uint32;
11
12     init(owner: Address) {
13         self.owner = owner;
14         self.counter = 0;
15     }
16
17     fun add(v: Int) {
18
19         // Check sender
20         let ctx: Context = context();
21         require(ctx.sender == self.owner, "Invalid sender");
22
23         // Update counter
24         self.counter = (self.counter + v);
25     }
26
27     receive(msg: Add) {
28         self.add(msg.amount);
29     }
30
31     receive("increment") {
32         self.add(1);
33         self.reply("incremented".asComment());
34     }
35
36     get fun counter(): Int {
37         return self.counter;
38     }
39 }
```

Helloworld 编译

配置说明

- 在tact.config.json文件中配置源文件和 输出目录
- 在package.json文件中配置 编译命令
- 执行命令 **yarn build** 编译

```
contract.tact  {} tact.config.json  {} package.json
{} tact.config.json > ...
Steve Korshakov, 10 months ago | 1 author (Steve Korshakov)
1  {
2    "projects": [{
3      "name": "sample",
4      "path": "./sources/contract.tact",
5      "output": "./sources/output",
6      "options": {
7      }
8    }
9  ]
10 }
Steve Korshakov, 12 months ago • feat: initial comm.
```

```
contract.tact  {} tact.config.json  {} package.json  X
{} package.json > {} dependencies
Anton Trunov, 3 weeks ago | 3 authors (Steve Korshakov and others)
1  {
2    "private": true,
3    "scripts": {
4      "build": "tact --config ./tact.config.json",
5      "test": "jest",
6      "deploy": "ts-node ./sources/contract.deploy.ts",
7      "read": "ts-node ./sources/contract.read.ts"
8    },
9    "dependencies": {
10   }
```

```
$ yarn build
yarn run v1.22.21
$ tact --config ./tact.config.json
🔧 Compiling project sample...
> SampleTactContract: tact compiler
> SampleTactContract: func compiler
> SampleTactContract: fift decompiler
> Packaging
> SampleTactContract
> Bindings
> SampleTactContract
> Reports
> SampleTactContract
🌟 Done in 2.83s.
```

Helloworld 部署

准备部署代码说明

- 导入相关的TS库
- 准备合约参数
- 准备部署合约
- 计算合约地址
- 合约部署网址

```
contract.tact  {} tact.config.json  {} package.json  TS contract.deploy.ts M X  TS deployer

sources > TS contract.deploy.ts > <function>

1  import * as fs from "fs";
2  import * as path from "path";
3  import { Address, contractAddress } from "@ton/core";
4  import { SampleTactContract } from "../output/sample_SampleTactContract";
5  import { prepareTactDeployment } from "@tact-lang/deployer";
6
7  (async () => {
8      // Parameters
9      let testnet = true;
10     let packageName = "sample_SampleTactContract.pkg";
11     let owner = Address.parse("0QC0BWOHCgvMpgQPuhg43b-EWAIL-FuNmSyNpm6HnV_cate0");
12     let init = await SampleTactContract.init(owner);
13
14     // Load required data
15     let address = contractAddress(0, init);
16     let data = init.data.toBoc();
17     let pkg = fs.readFileSync(path.resolve(__dirname, "output", packageName));
18
19     // Preparing
20     console.log("Uploading package...");
21     let prepare = await prepareTactDeployment({ pkg, data, testnet });
22
23     // Deploying

PROBLEMS 1  OUTPUT  DEBUG CONSOLE  TERMINAL  GITLENS

Contract Address
=====
kQBDC5ap1-TPtLT-s_buPVagxe-8YLjyhkj04Vh6JoC7vos
=====

Please, follow deployment link
=====
https://verifier.ton.org/tactDeployer/QmXg1wY11khAUTpwn5gon9BySaoqw4Hew2NkbQ251CeqWk?testnet
=====

🌟 Done in 10.05s.
```



Helloworld 部署

浏览器打开网址

https://verifier.ton.org/tactDeployer/QmXg1wY11khAUTpwn5gon9BySaoqw4Hew2Nkb...

Sample Tact Contract

Compiler

Tact 1.1.5

Code Hash

JW74evisfNkyud3ecvl2v7cDgdMcqE5ITRjMysD8TvA=

Data Hash

kWzyg2m/q4gmlfd+kwgMkFx3jymxraC4Ou/Yqblqd1g=

Workchain

Basic Workchain (0)

Deploy

Value to initialize contract (TON)

Contract Address

EQBDCC5ap1-TPtIT-s_buPVagxe-8YLjyhkj04Vh6JoC7kGm

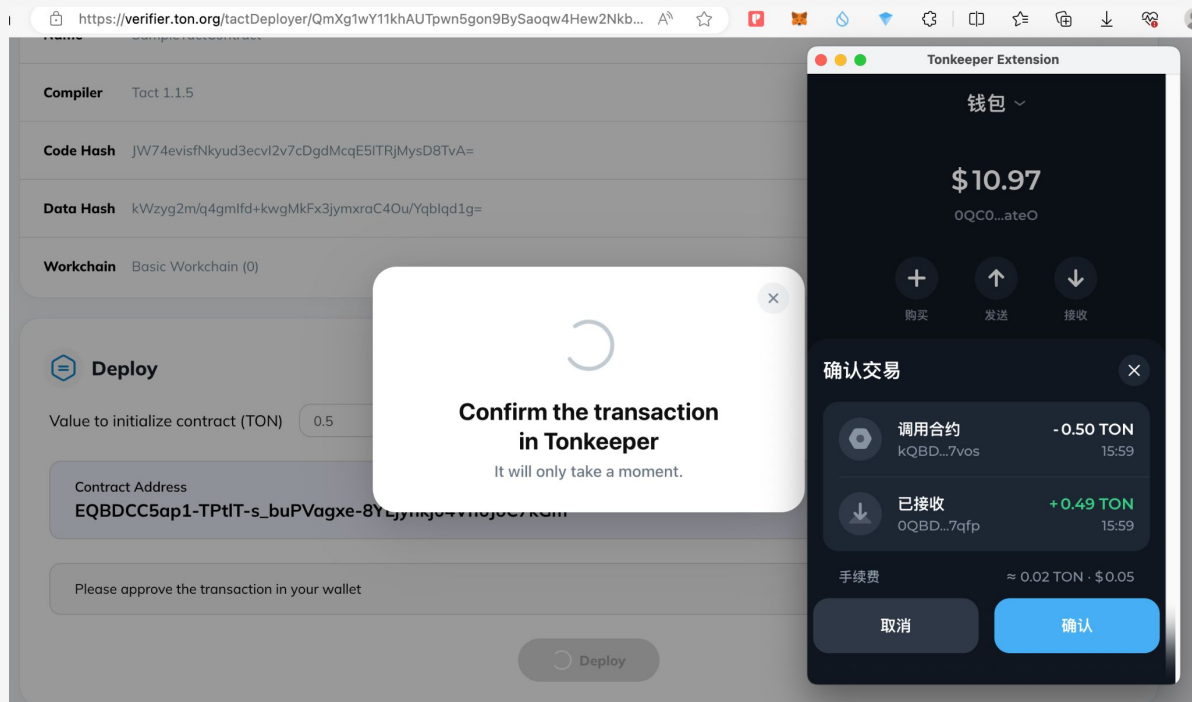
Transaction was rejected. Please retry.

Deploy



Helloworld 部署

钱包签名部署





Helloworld 部署

浏览器查看已部署合约

https://testnet.tonviewer.com/kQA4U6yQsPUjkZii2dyoiQlsfzUvyr15KIMIso1YHfWE4Y_d

Tonviewer

kQBDCC5ap1-TPtLT-s_buPVagxe-8YLjyhkj04Vh6JoC7vos

Wallet Code Methods

Address
kQBDCC5ap1-TPtLT-s_buPVagxe-8YLjyhkj04Vh6JoC7vos
[Tonscan](#) · [ton.cx](#) · [toncoin.org](#)

Balance
0 TON ≈ \$0

Status: Active Raw: 0:43082e...e89a02ee Interfaces: -

QR Code

Account history

	Date	Initiator	
2 minutes ago	</> Contract called	kQC0BWOH...nV_caopL	0x946a98b6 +0.5 TON
	↑ Sent TON	kQC0BWOH...nV_caopL	-0.491 TON
	</> Contract deploy	kQBDCC5a...6JoC7vos	-

Helloworld 调用getter

合约调用说明

- 导入TON的开发库
- 准备合约参数
- 调用getCounter读取状态值

```
{} package.json M TS contract.read.ts M X TS contract.read.ts (Working Tree) M TS contrac...

sources > TS contract.read.ts > ...

3 import { Address, contractAddress } from "@ton/core";
4 import { TonClient4 } from "@ton/ton";
5 import { SampleTactContract } from "../output/sample_SampleTactContract";
6 import { prepareTactDeployment } from "@tact-lang/deployer";
7
8 (async () => {
9   const client = new TonClient4({
10     endpoint: "https://sandbox-v4.tonhubapi.com", // Test-net API endpoint
11   });
12
13   // Parameters
14   let testnet = true;
15   let packageName = "sample_SampleTactContract.pkg";
16   let owner = Address.parse("0QC0BWOHCGVmpGQPUhg43b-EWAIL-FuNmSyNpm6HnV_cate0");
17   let init = await SampleTactContract.init(owner);
18   let contract_address = contractAddress(0, init);
19
20   // Preparing
21   console.log("Reading Contract Info...");
22   console.log(contract_address);
23
24   // Input the contract address
25   let contract = await SampleTactContract.fromAddress(contract_address);
26   let contract_open = await client.open(contract);
27   console.log("Counter Value: " + (await contract_open.getCounter()));
28 })();

PROBLEMS 1 OUTPUT DEBUG CONSOLE TERMINAL GITLENS

$ yarn read
yarn run v1.22.21
$ ts-node ./sources/contract.read.ts
Reading Contract Info...
EQBDCC5ap1-TPtLT-s_buPVagxe-8YLjyhkj04Vh6JoC7kGm
Counter Value: 0
+ Done in 2.92s.
```

Helloworld 发送消息

合约调用说明

- 导入TON的开发库
- 准备合约参数
- 调用send方法发送消息

```
TS contract.write.ts U X TS contract.deploy.ts M TS index.d.ts TS contract.spec.ts TS sample...
sources > TS_contract.write.ts > ...
1 import { Address, contractAddress, toNano } from "@ton/core";
2 import { TonClient4, WalletContractV4 } from "@ton/ton";
3 import { SampleTactContract } from "../output/sample_SampleTactContract";
4 import { mnemonicToPrivateKey } from "@ton/crypto";
5
6 const Sleep = (ms: number) => {
7   return new Promise(resolve => setTimeout(resolve, ms));
8 }
9
10 (async () => {
11   const client = new TonClient4({
12     endpoint: "https://sandbox-v4.tonhubapi.com", // 🟡 Test-net API endpoint
13   });
14
15   // open wallet v4 (notice the correct wallet version here)
16   const mnemonic = "c ... .. company suggest she
17   const key = await mnemonicToPrivateKey(mnemonic.split(" "));
18   const wallet = WalletContractV4.create({ publicKey: key.publicKey, workchain: 0 });
19
20   // open wallet and read the current seqno of the wallet
21   const walletContract = client.open(wallet);
22   const walletSender = walletContract.sender(key.secretKey);
23
24   // open the contract address
25   let owner = Address.parse("0QAVIKI250id19N8CvORmrmoj114PqcOMnPcaQb4LeAR-VR");
26   let init = await SampleTactContract.init(owner);
27   let contract_address = contractAddress(0, init);
28   let contract = await SampleTactContract.fromAddress(contract_address);
29   let contract_open = await client.open(contract);
30
31   // send message to contract
32   await contract_open.send(walletSender, { value: toNano(1) }, "increment");
33
34   await Sleep(3000);
35   console.log("Counter Value: " + (await contract_open.getCounter()));
36 })();
37
38
```

PROBLEMS OUTPUT DEBUG CONSOLE **TERMINAL** GITLENS

```
$ yarn write
yarn run v1.22.21
$ ts-node ./sources/contract.write.ts
Counter Value: 1
Done in 17.84s.
```



总结

- TON生态背靠Telegram
- TON的合约语言有FunC和Tact
- 在TON上操作, 需要钱包和TON代币, 开发时建议使用TON测试网络
- Tact开发环境搭建
- Tact Helloworld合约编写, 编译、部署、和调用



参考资料

- <https://docs.ton.org/>
- <https://tact-lang.org/>
- <https://github.com/tact-lang/awesome-tact>
- <https://docs.tact-lang.org/>
- <https://ton.app>
- <https://foresightnews.pro/article/detail/21329>
- <https://www.fx168news.com/article/392979>
- <https://www.panewslab.com/zh/articledetails/ty2spsy.html>
- https://www.techflowpost.com/article/detail_14664.html



作业

修改 SampleTactContract 合约, 增加一个 getter 方法, 重新部署, 并调用此新方法。

修改 SampleTactContract 合约, 增加一个消息处理方法, 重新部署, 并发送此新消息。



ScaleBit

Thanks

Contact us:

- Twitter: @scalebit_
- Email: contact@scalebit.xyz

More information : www.scalebit.xyz