深圳龙霸软件 智能合约dapp开发指南

产品名称	深圳龙霸软件 智能合约dapp开发指南
公司名称	深圳龙霸网络技术有限公司
价格	100.00/件
规格参数	龙霸软件:链游源码开发 NFT 链游:软件定制 DAPP开发:软件开发
公司地址	深圳市龙华区民治街道民治社区1970科技园8栋115（注册地址）
联系电话	13632978801

产品详情

深圳龙霸软件 智能合约dapp开发指南

（详细了解欢迎电话联系）

从底层的基础设施到建立在其之上的应用层的智能合约生态，Web3领域已经发展的相当之快。基于智能合约，我们可以构建出各种各样的dApp（去中心化应用），

比如DeFi、GameFi、NFT、GameFi甚至是这两年兴起的SocialFi类应用。列举出的每一类dApp都在各自的领域内大放异彩，造就了整个Web3领域的蓬勃发展的繁荣景象。

当你熟悉智能合约kaifa之后，不论你想要kaifa任何一种（上面列出的）dApp，都已经拥有50%的基础优势，剩下的50%就是需要熟悉对应dApp领域的业务知识，下面就一起来看看智能合约kaifa的学习路线。

需要特别说明的是，本文注重实用性，不会过多介绍相关工具的历史，这方面读者可自行查询。

1.选择智能合约语言

合约编程语言是智能合约kaifa者基本的部分，你至少应该掌握一种才能编写智能合约。合约语言通常不是传统的编程语言（如C/C++，Python等），为了保证合约代码在任何一个节点上都能够得到一个确定的执行结果，

必须要求合约不能访问外部世界（如进行HTTP连接或操作文件），早期的kaifa者们设计出了专为编写合约的编程语言，如Solidity、Vyper，

后来又诞生了一些新的合约语言如Move或Rust（但包含一些使用限制）。好消息是这些语言都大量借鉴了传统编程语言的语法，如Solidity借鉴JavaScript、Vyper借鉴Python，这使得成为一名合约kaifa者的学习曲线平坦了一些。

1.1Solidity

首先，一个不用过多思考的选择是Solidity，因为目前90%的智能合约都是Solidity写的。Solidity算是早也是流行的面向对象的静态语言，初2014年是为以太坊EVM量身设计的，

后来又出现了很多兼容以太坊EVM的平台（如Ethereum、Avalanche、Moonbeam、Polygon、BSC），所以现在Solidity也可以运行在其他那些兼容EVM的上。在目前排名的Defi项目

中，有九个使用Solidity作为他们的主要编程语言。

Solidity发展至今已经快十年，生态内已经有大量的kaifa工具可供使用，包括第三方库以及IDE等（后面介绍）。另外，在EVM上运行的比Solidity更原生的语言是汇编语言Yul，进阶Solidity时你会了解到Solidity与Yul通过内联交互以提高性能的应用。

需要注意的是，Solidity在语法设计上存在一些缺陷，当然，这些年不断的被改进，在易用性和安全性上已经得到了极大的提升。

对于Solidity的学习，这里强烈推荐本仓库主页中列出的书籍智能合约技术与kaifa

，且在本仓库中也存放有笔者对该书的代码笔记。

1.2Vyper

Vyper是另一种与EVM兼容（可编译为像Solidity一样的EVM字节码）且相比Solidity更注重安全性的合约语言，它与Python的语法非常相似，但相比Python去掉了许多不必要的特性（如类继承、函数重载、运算符重载等），

减少特性可以语言变得简单，也减少了出错的机会。

另外，Vyper还旨在让任何人尽可能难以编写误导性代码。读者（即审核员）的简单性比作者（即kaifa人员）的简单性更重要。这样，将更容易识别智能合约或去中心化应用程序(dApp)中的恶意代码。

需要注意的是，Vyper不是Solidity的完全替代品，而是一种在需要高安全级别时使用的语言。用Vyper编写的项目示例包括Uniswapv1、Curve.fi和个ETH2.0存款合约。

1.3Move

Move创建于2019年，是一门相对Solidity和Vyper来说较难掌握的合约语言，它基于Rust改写，初是为Meta的Diem项目而kaifa的，在Diem项目解散之后，其创始团队出走分别创立的Aptos

与Sui，也将Move作为核心编程语言。

Move的主要特点是面向资产编程（资源是一等公民）、安全（继承了Rust诸多安全特性）以及模块化（模块可以迭代）。

相对来说，Move语言目前还十分年轻，缺乏大规模的工程化验证，并且其kaifa链尚不完善，合约规范也没有形成，所以建议只作为兴趣了解。

1.4Rust

Rust初由Mozilla员工GraydonHoare在2006年设计和发布，是一种为性能和安全性，尤其是安全并发性而设计的语言，它在语法上与C++相似。Rust

并不是一开始就为了智能合约而设计，而是作为一门传统的力求高安全性的语言而存在，由于其在安全性上的优势十分契合智能合约的应用场景，所以人们选择直接将其引入领域。

目前，Rust在各领域已经被广泛应用，如基础设施建设（Layer1）、合约编程（Layer2）等。目前将Rust语言作为核心kaifa语言的就有Polkadot、Solana、Near。

需要注意的是，Rust的语法是出了名的复杂，其学习曲线足够陡峭，其学习难度往往让人望而生畏。不过在Rust语言设计团队（LangTeam）在官方博客中公布的Rust语言2024年的更新路线图中，就昭示了降低学习难度是

Rust语言的未来发展方向。Rust在语言设计层面比较贴近C/C++等高性能语言，所以熟悉C/C++的kaifa者会有稍微一点优势。

2.部署和测试框架

2.1概览

这部分介绍用来协助部署和测试合约代码的一些框架工具。经过此领域的不断发展，如今已经有各种各样的合约框架或工具可供使用。

2.2Remix

首先是Remix，它本身不是一个框架而是一个主要基于浏览器（也支持桌面）的IDE，能够提供基于以太坊的在线智能合约编译、测试和部署功能，因为是基于浏览器的工具，所以不关心操作系统，直接开箱即用。

在Remix浏览器版本中编写的代码会保存在浏览器缓存中，所以不小心清除缓存就会导致你的工作区（workspace）被清空，这算是一个缺点，不过Remix后来也支持连接到电脑本地的工作区。

Remix是早的Solidity kaifa工具，几乎所有的合约kaifa者都是从Remix开始学习。但是当kaifa者在合约中集成更复杂的逻辑时（较大的合约项目），就需要选择自动化程度更高的框架来kaifa、测试和部署合约了。

2.3Truffle

Truffle是早出现的编写以太坊合约的框架，由Consensys在2016年创建，它是基于JavaScript编写的。官方对其的介绍是：一个用来构建、测试和部署以太坊网络应用的框架。

整个框架可以当做一个套件包含三个工具：Truffle（kaifa和测试环境）、Ganache（通过桌面版或命令行快速部署本地EVM）和Drizzle（丰富的用于构建dApp的前端UI库）。

Truffle是所有框架中历史影响大的，你可以看到他们对行业的影响，很多框架都采用了Truffle的实践做法。你会看到大部分智能合约工程师岗位都要求掌握这个框架的使用。