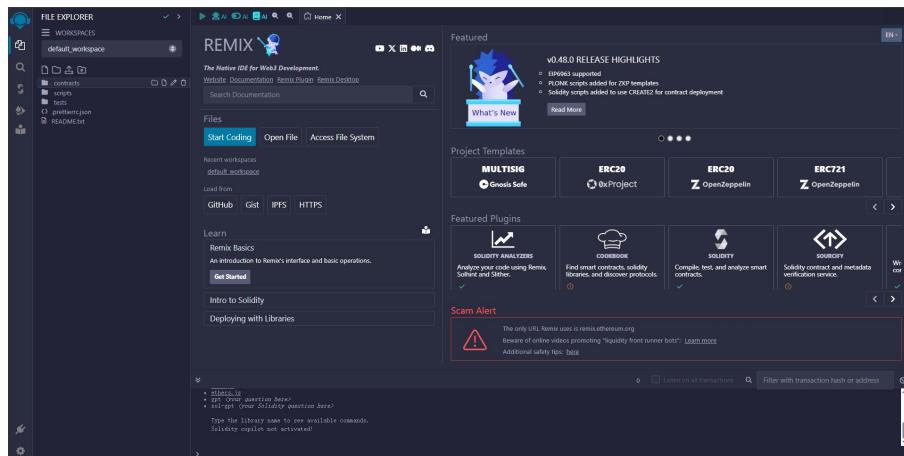
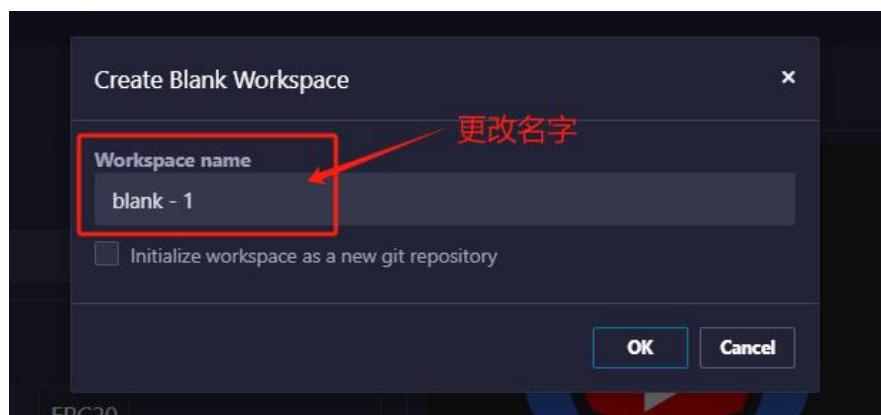
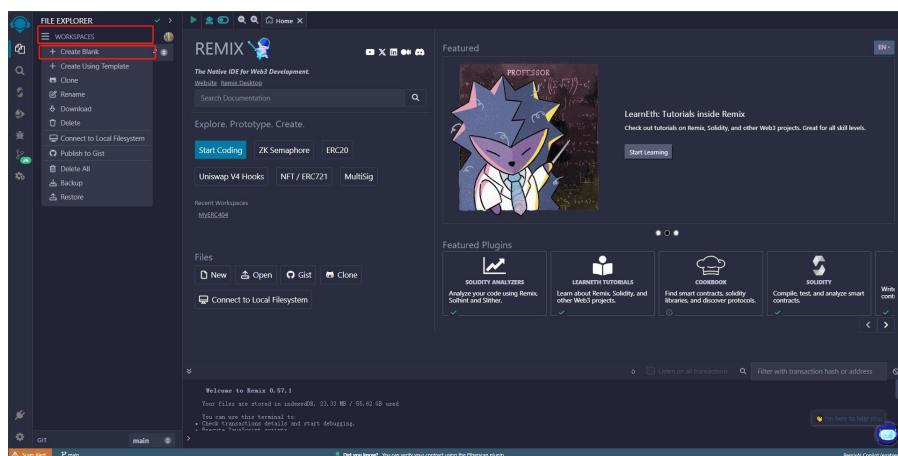


# NFT 拍卖程序智能合约部署步骤

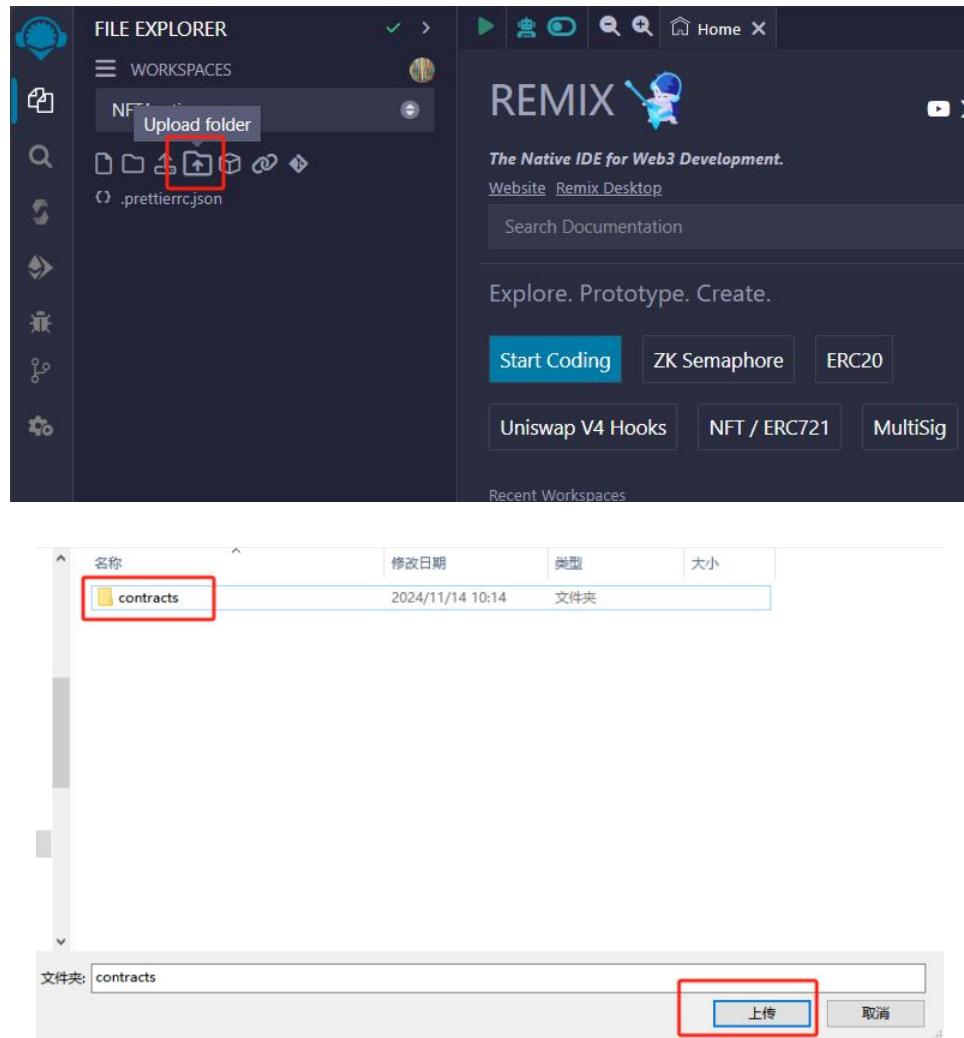
1. 打开 Remix 在线编译器。网址为: <https://remix.ethereum.org/>



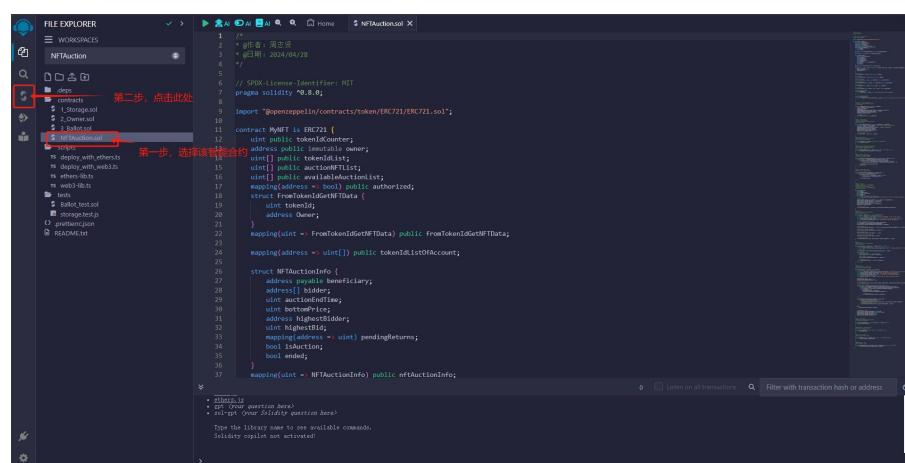
2. 新建工作空间。



### 3. 导入智能合约。



### 4. 编译智能合约。



最新版本的编译器

```

1 // 智能合约：NFT Auction
2 * 创建者：匿名
3 * 日期：2024/04/28
4 *
5 // SPDX-License-Identifier: MIT
6 pragma solidity >=0.8.0;
7
8 import "openzeppelin/contracts/token/ERC721/ERC721.sol";
9
10 contract MyFT is ERC721 {
11     uint public tokenIdCounter;
12     address public immutable owner;
13     uint[] public tokenIds;
14     uint[] public auctionList;
15     uint[] public availableAuctionList;
16     mapping(address => bool) public authorized;
17     struct FromtokenIdData {
18         uint tokenId;
19         address owner;
20     }
21     mapping(address => uint) public tokenIdListOrAccount;
22
23     struct NFIAuctionInfo {
24         address payable beneficiary;
25         address[] bidder;
26         uint auctionDeadline;
27         uint bottomPrice;
28         address highestBidder;
29         uint highestBid;
30         mapping(address => uint) pendingReturns;
31         bool bidAction;
32         bool endBid;
33     }
34     mapping(uint => NFIAuctionInfo) public nftAuctionInfo;
35
36     mapping(uint => uint) public tokenIdListOrAcount;
37
38     event NFIAuctionCreated(
39         address indexed beneficiary,
40         address indexed bidder,
41         uint indexed auctionDeadline,
42         uint indexed bottomPrice,
43         address indexed highestBidder,
44         uint indexed highestBid,
45         uint indexed pendingReturns,
46         bool indexed bidAction,
47         bool indexed endBid
48     );
49     mapping(uint => NFIAuctionInfo) public nftAuctionInfo;
50
51     event NFIAuctionUpdated(
52         address indexed beneficiary,
53         address indexed bidder,
54         uint indexed auctionDeadline,
55         uint indexed bottomPrice,
56         address indexed highestBidder,
57         uint indexed highestBid,
58         uint indexed pendingReturns,
59         bool indexed bidAction,
60         bool indexed endBid
61     );
62     mapping(uint => NFIAuctionInfo) public nftAuctionInfo;
63
64     event NFIAuctionCancelled(
65         address indexed beneficiary,
66         address indexed bidder,
67         uint indexed auctionDeadline,
68         uint indexed bottomPrice,
69         address indexed highestBidder,
70         uint indexed highestBid,
71         uint indexed pendingReturns,
72         bool indexed bidAction,
73         bool indexed endBid
74     );
75     mapping(uint => NFIAuctionInfo) public nftAuctionInfo;
76
77     event NFIAuctionCompleted(
78         address indexed beneficiary,
79         address indexed bidder,
80         uint indexed auctionDeadline,
81         uint indexed bottomPrice,
82         address indexed highestBidder,
83         uint indexed highestBid,
84         uint indexed pendingReturns,
85         bool indexed bidAction,
86         bool indexed endBid
87     );
88     mapping(uint => NFIAuctionInfo) public nftAuctionInfo;
89
90     event NFIAuctionFinalized(
91         address indexed beneficiary,
92         address indexed bidder,
93         uint indexed auctionDeadline,
94         uint indexed bottomPrice,
95         address indexed highestBidder,
96         uint indexed highestBid,
97         uint indexed pendingReturns,
98         bool indexed bidAction,
99         bool indexed endBid
100    );
101    mapping(uint => NFIAuctionInfo) public nftAuctionInfo;
102}

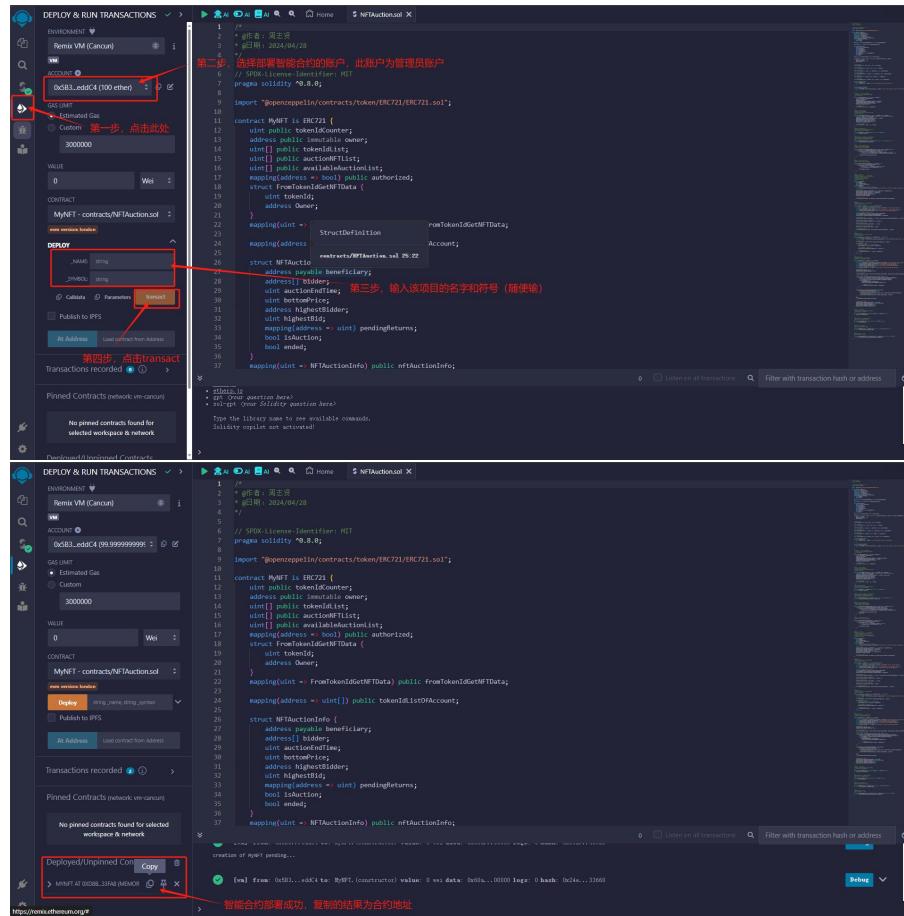
```

VM版本选择london

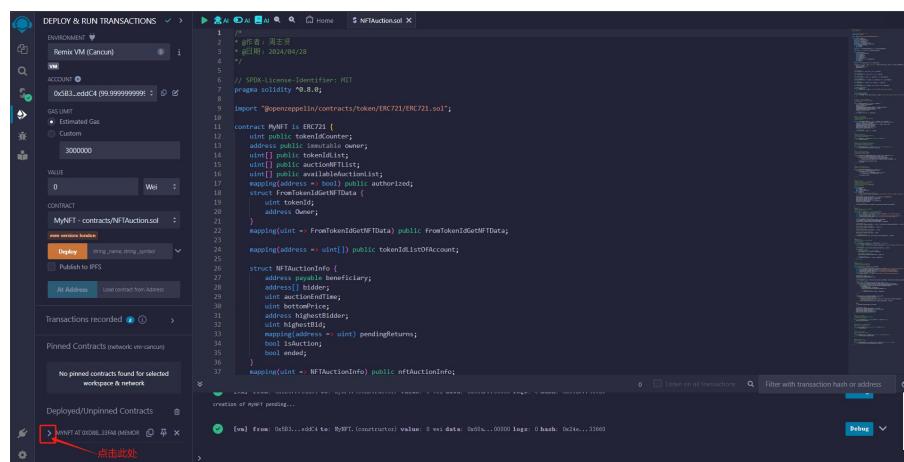
然后点击此处，编译智能合约

智能合成功能已启用

## 5. 部署智能合约。



## 6. 调试智能合约。



The screenshot shows the Truffle UI interface. On the left, there's a sidebar with a search icon and a dropdown menu titled "Deployed/Unpinned Contracts". This dropdown is currently expanded, showing a list of functions: approve, bid, transfer, mintERC721, transferFrom, renounceOwnership, safeTransferFrom, setApprovalForAll, startAuction, tenderERC20, transferERC721, transferFrom, safeTransfer, auctionERC721, authorized, availableForSale, and balanceOf. The "bid" function is highlighted with a red box. The right side of the interface shows the Solidity code for the "NFTAuction.sol" contract.

```
1 // SPDX-License-Identifier: MIT
2
3 // Copyright 2024. All rights reserved.
4
5 import "openzeppelin/contracts/token/ERC721/ERC721.sol";
6
7
8 contract MyNFT is ERC721 {
9     uint public tokenIdCounter;
10    address payable owner;
11    uint[] public auctionIdList;
12    uint[] public availableAuctionList;
13    struct FromTokenIdSaleInfo {
14        uint tokenId;
15        address owner;
16    }
17
18    mapping(uint => uint) public tokenIdsToAuction;
19
20    struct NFTAuctionInfo {
21        address payable beneficiary;
22        address bidder;
23        uint startingPrice;
24        uint bottomPrice;
25        address highestBidder;
26        uint endPrice;
27        mapping(address => uint) pendingReturns;
28        bool auction;
29        bool ended;
30    }
31
32    mapping(uint => NFTAuctionInfo) public nftAuctionInfo;
33
34
35
36
37 }
```

In the center of the screen, there's a text input field with placeholder text: "在想要测试的函数中输入对应的参数即可 调试智能合约". Below the code editor, there are buttons for "Deploy" and "Run".