

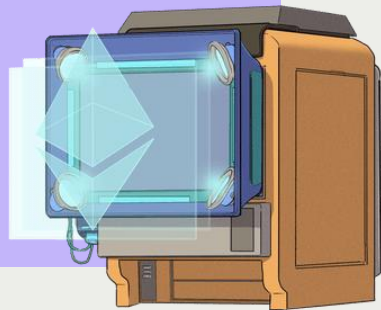
免责声明：

本课程所涉及案例仅为学习使用，不构成投资建议。
请谨慎辨别。

在中华人民共和国，区块链上的可转移数字资产（包括但不限于代币）与法币的直接或间接交换是违法行为，请遵守相关法律法规，避免参与任何非法活动。

第五讲 使用 ethers.js 和区块链交互 (下)

主讲人: steven



第五讲 使用 ethers.js 和区块链交互（下）

一个例子

```
const abi = [  
  "function decimals() view returns (uint8)",  
  "function transfer(address recipient, uint256 amount) returns (bool success)"  
]  
  
const contract = new Contract(CONTRACT_ADDRESS, abi, signer);  
const decimals = await contract.decimals();  
const amount = parseUnits("1.0", 18);  
const tx = await contract.transfer(RECIPIENT_ADDRESS, amount);  
await tx.wait();
```

第五讲 使用 ethers.js 和区块链交互（下）

订阅事件

```
const abi = [
  "event Transfer(address indexed from, address indexed to, uint amount)"
]

contract = new Contract(CONTRACT_ADDRESS, abi, provider)
contract.on("Transfer", (from, to, _amount, event) => {
  console.log(`${ from } => ${ to }: ${ formatEther(_amount, 18) }`);

  console.log("event:", event)
});
```

第五讲 使用 ethers.js 和区块链交互（下）

事件查询

```
const abi = [
  "event Transfer(address indexed from, address indexed to, uint amount)"
]

const contract = new Contract(CONTRACT_ADDRESS, abi, provider)
const filter = contract.filters.Transfer
const events = await contract.queryFilter(filter)
```

Contract 实例上的 `queryFilter()` 方法，第二和第三个参数可以指定需要查询的起始和终止区块，限定查询的区块范围。

第五讲 使用 ethers.js 和区块链交互（下）

签名与验证

```
const sig = await signer.signMessage(message);
```

```
const address = verifyMessage(message, sig);
```

第五讲 使用 ethers.js 和区块链交互（下）

以太币单位

以太坊的最小单位是 wei（在以太坊系统内，以太通常以 wei 表示，这是以太的最小单位，不可再分），1 个币即 1 ether， $1 \text{ ether} = 10^{18} \text{ wei}$ ，他们中间还有若干单位可以见下表：

单位	1 ether
wei	1^{18} wei
kwei	1^{15} kwei
mwei	1^{12} mwei
gwei	1^9 gwei
szabo	1^6 szabo
finney	1^3 finney
ether	1 ether

第五讲 使用 ethers.js 和区块链交互（下）

一个问题

在 JavaScript 中，当一个 Number 类型的数字超过 $2^{53}-1$ （约 [9.007E15](#)）后会带来诸多安全性问题。

一个以太等于 [1E18](#) wei，我们所以需要一种能处理“大数”的方案，在 ethers.js v5.x 版本中，引入了 BigNumber 类进行相关处理，而在 ethers.js v6.x 版本中，使用 [BigInt](#) 进行处理。

第五讲 使用 ethers.js 和区块链交互（下）

以太单位换算的工具函数（一）

```
formatEther(1000000000000000000n); // "1.0"
```

```
formatEther(200000000000000000n); // "0.002"
```

```
formatEther(1234000000000000000n); // "1.234"
```

```
formatUnits(1234000000000000000n); // "1.234"
```

```
formatUnits(1234000000000000000n, 'ether'); // "1.234"
```

```
formatUnits(1234000000000000000n, 'finney'); // "1234.0"
```

```
formatUnits(1234000000000000000n, 'gwei'); // "1234000000.0"
```

第五讲 使用 ethers.js 和区块链交互（下）

以太单位换算的工具函数（二）

```
parseEther("1.234"); // 1234000000000000000n
```

```
parseEther("0.002"); // 2000000000000000000n
```

```
parseUnits("1.234"); // 1234000000000000000n
```

```
parseUnits("1.234", 'ether'); // 1234000000000000000n
```

```
parseUnits("1.234", 'finney'); // 123400000000000000n
```

```
parseUnits("1.234", 'gwei'); // 1234000000n
```

谢谢观看
