<h1 style="font-size:32px;text-align:center;color:red">区块链知识</h1>

---

**### 说到区块链，我想我们听到最多的就是比特币**

> 每当同学聚餐问我学的什么的时候，我一说区块链，他们就说自己知道：是比特币！

> 但是很明显，他们只是了解一丢丢。

---

**### 区块链是什么？**

> 官方定义已经足够充分，但我不认为好理解。

> ![区块链定义](https://gitee.com/goodflyo/cdn/raw/master/01/define.png)

**\*\*谈谈个人理解（错误不可避免，今后将会修正）\*\***

- 区块链是一个数据库的集合（超大链表），每个区块之间可以通过哈希指针连接起来

- 区块链=区块+链（其中“链”是一个区块到另一个区块链的"指针"）

- 区块=区块头+区块体

  ![区块链自我理解](https://gitee.com/goodflyo/cdn/raw/master/01/myknow.png)

- 区块体里面则包含了我们要存储的数据，区块头则：![区块体](https://gitee.com/goodflyo/cdn/raw/master/01/body.png)

- 如上所示，区块链并不是什么复杂的概念，反而像个一步步迭代过来的新型数据库，只不过这个数据库搭载了很多其他领域的知识

---

**### 区块链的应用实现**

- 比特币：区块链 1.0

  - 闪电网络

  - 数字钱包

  - 挖矿（PoW、PoS、DPoS）

- 以太坊：区块链 2.0

  - 雷电网络

  - 数字钱包

  - 挖矿

  - 智能合约

- EOS：区块链 3.0

  - 数字钱包

  - 挖矿

> 当然，以上只是最经典的应用的例举，还有很多的现代区块链落地产品。但是我们始终要知道，区块链是比特币的底层技术，比特币是区块链的应用实现。

区块链大致结构框架

![区块链框架](https://gitee.com/goodflyo/cdn/raw/master/01/fabric.jpg)

---

<h1 style="font-size:32px;text-align:center;color:red">智能合约知识</h1>

**### 关键字 constan、view、pure、payable？**

---

Solidity 中变量按作用域划分有三种，分别是

> 前提：

- **\*\*状态变量（state variable）\*\***

- **\*\*局部变量（local variable）\*\***

- **\*\*全局变量（global variable）\*\***

  - **\*\*状态变量\*\***

    状态变量是数据存储在链上的变量，所有合约内函数都可以访问 ，gas 消耗高。状态变量在合约内、函数外声明：

    ```sol

    sol contract Variables {

         uint public x = 1;

         uint public y;

         string public z; }

    ```

    我们可以在函数里更改状态变量的值：

    ```sol

        function foo() external{

                // 可以在函数里更改状态变量的值

                x = 5;

                y = 2;

                z = "0xAA";

       }

    ```

  - **\*\*局部变量\*\***

    局部变量是仅在函数执行过程中有效的变量，函数退出后，变量无效。局部变量的数据存储在内存里，不上链，gas 低。局部变量在函数内声明：

    ```sol

     function bar() external pure returns(uint){

     uint xx = 1;

     uint yy = 3;

     uint zz = xx + yy;

     return(zz);

     }

    ```

  - **\*\*全局变量\*\***

    全局变量是全局范围工作的变量，都是 solidity 预留关键字。他们可以在函数内不声明直接使用：

    ```sol

     function global() external view returns(address, uint, bytes memory){

     address sender = msg.sender;

     uint blockNum = block.number;

     bytes memory data = msg.data;

     return(sender, blockNum, data);

     }

    ```

---

1. view 修饰的函数，是 constant 的别名，只能读取 storage 变量的值，不能写入。（不能改变状态）

   > 这里特别提到 **\*\*状态变量\*\***

2. pure 修饰的函数 ，不能对 storage 变量进行读写

3. payable 当一个函数被 payable 修饰，表示调用这个函数时，可以附加发送一些 ETH（当然也可以不发）。

   没有加 payable 的函数，则没有方法接受 ETH， 附加 ETH 调用会出错。

**\*\*remix 的界面\*\***

   ![支付展示](https://gitee.com/goodflyo/cdn/raw/master/01/payableShow.png)

<h1 style="font-size:32px;text-align:center;color:red">计划</h1>

1. [solidity 官方文档](https://solidity-cn.readthedocs.io/zh/develop/index.html)

2. [了解去中心化应用程序（DAPP）](https://search.bilibili.com/all?keyword=dapp&from\_source=webtop\_search&spm\_id\_from=333.1007&search\_source=5)

3. [参考计划](https://www.zhihu.com/search?type=content&q=%E5%8C%BA%E5%9D%97%E9%93%BE%E5%AD%A6%E4%B9%A0)