// SPDX-License-Identifier: MIT pragma solidity ^0.8.4; contract FunctionTypes{ uint256 public number = 5; constructor() payable {} // 函数类型 // function (<parameter types>) {internal|external} [pure|view|payable] [returns (<return types>)] // 默认function function add() external{ number = number + 1; } // pure: 纯纯牛马 function addPure(uint256 \_number) external pure returns(uint256 new\_number){ new\_number = \_number+1; } // view: 看客 function addView() external view returns(uint256 new\_number) { new\_number = number + 1; } // internal: 内部 function minus() internal { number = number - 1; } // 合约内的函数可以调用内部函数 function minusCall() external { minus(); } // payable: 递钱，能给合约支付eth的函数 function minusPayable() external payable returns(uint256 balance) { minus(); balance = address(this).balance; } }