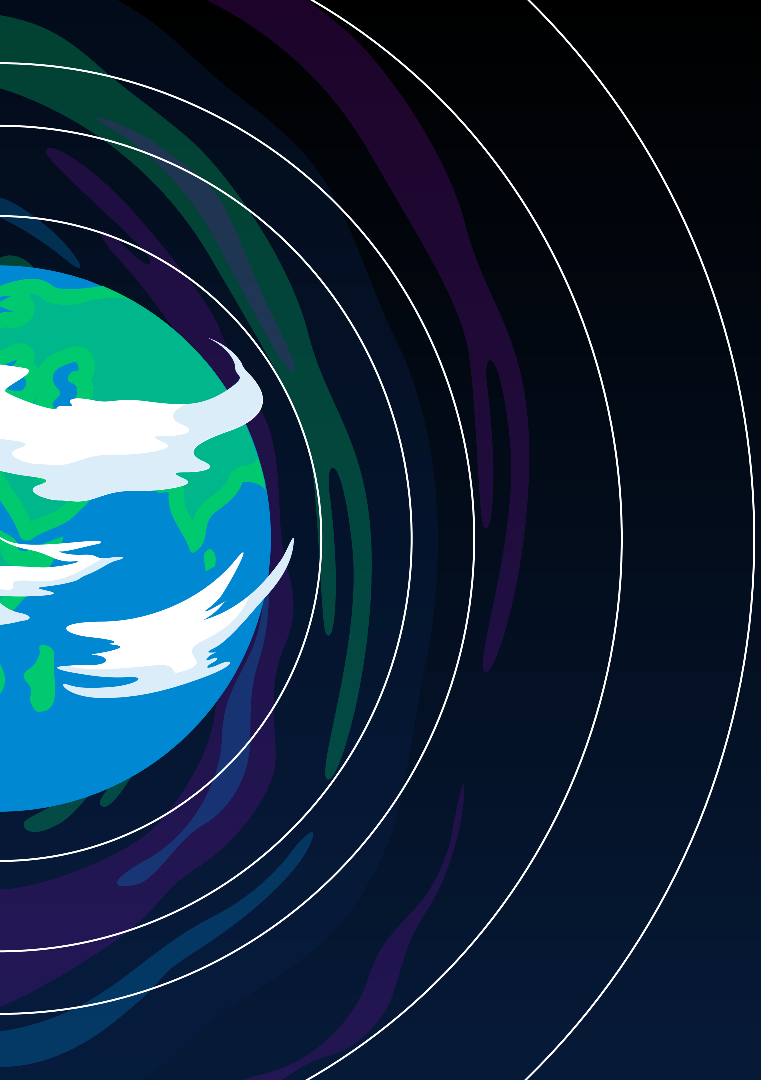
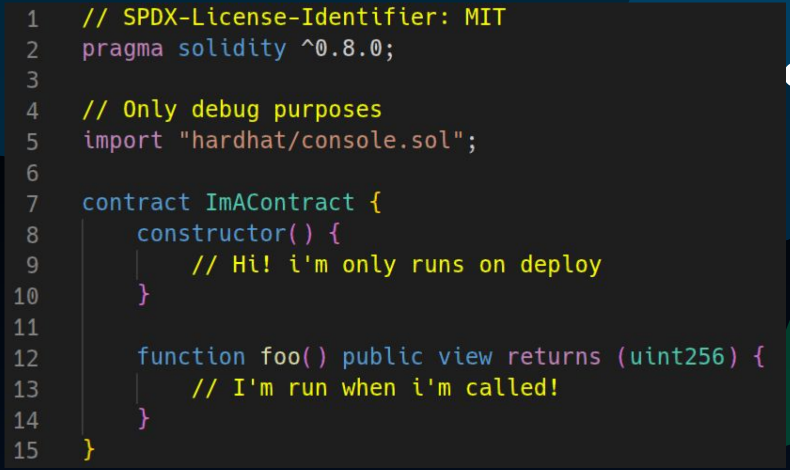


PDP 1

Learn solidity & smart contracts

Quimey Lucas Marquez
Code&Care 2022

What is an Smart Contract?



```solidity
// SPDX-License-Identifier: MIT
pragma solidity ^0.8.0;

// Only debug purposes
import "hardhat/console.sol";

contract ImAContract {
    constructor() {
        // Hi! i'm only runs on deploy
    }

    function foo() public view returns (uint256) {
        // I'm run when i'm called!
    }
}
```

A code, sequence of bits, living in the blockchain, with their own address to interact with and defined behaviour in functions.

- Solidity it's a good language to code it (there are more languages).
- Once deployed it's immutable in behaviour but no in memory, has (2^256)-1 slots of memory of 256 bit length each
- Has a maximum weight, a top of 24kb deployed
- Define a license and a compiler version to be builded

NOTE: this presentation it's ethereum oriented, some things couldn't be like this in other blockchains

Solidity 101, Language Basics

If

```
if (/* boolean expr */) {...}
else if (...) {...}
else {...}
```

Loops

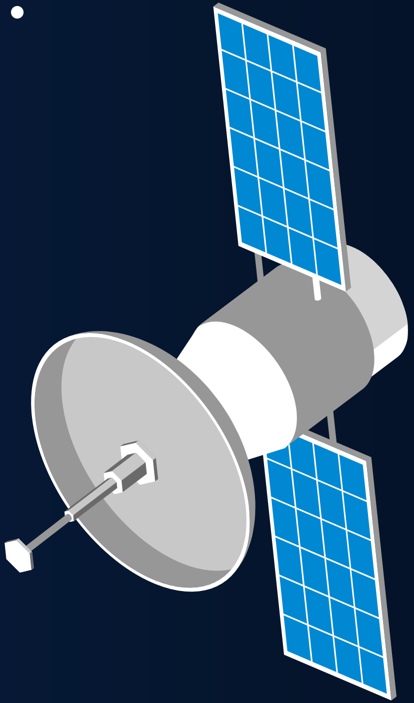
```
for (uint256 i = 0; i < 10; ++i)
{...}
```

Types

uintXX: 256 bits length, we can split it: uint256, uint8, uint32…
Strings ar not recommended
Arrays: fixed length on function level, dynamic at contract level
Mappings: kind of functions, map one type into another
Structs: just lik C

Functions modifiers

Ethereum allow us to set the type and
scope for a function



Type

- `view`: only reads from the contract
- `pure`: don't even read the contract
- `payable`: allow to accept money in the call

Scope

- `public`: anyone can call it
- `private`: just the contract can call it
- `internal`: contract and subcontracts can call it
- `external`: only the outside can call it

ethers and hardhat

Are two libraries to assist us in the contract development

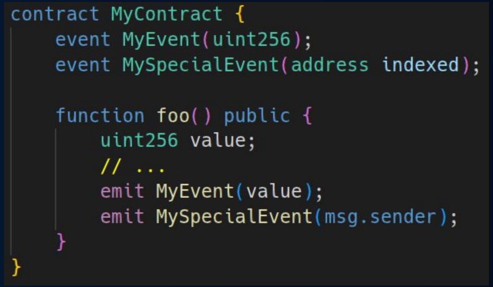
ethers: front-end library, good hardhat partner, to interact with extensions metamask like

hardhat: very useful library, help us to mount a local ethereum network, makes easier the contract development

Solidity 201, Solid than a rock

event

We can send information on it, or not, just fires the event, we can send it to anyone or only for who make the call.

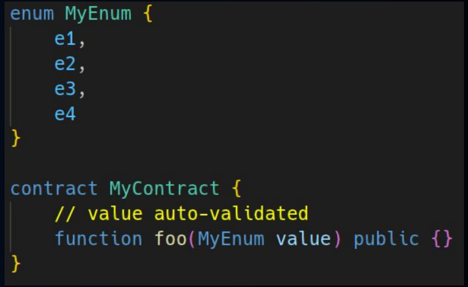


```
contract MyContract {
    event MyEvent(uint256);
    event MySpecialEvent(address indexed);

    function foo() public {
        uint256 value;
        // ...
        emit MyEvent(value);
        emit MySpecialEvent(msg.sender);
    }
}
```

enum

We can't set the value of the enum but if we receive the enum as param of a function don't require validation in our contract.

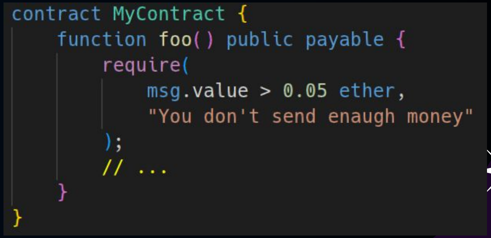


```
enum MyEnum {
    e1,
    e2,
    e3,
    e4
}

contract MyContract {
    // value auto-validated
    function foo(MyEnum value) public {}
}
```

require

Kind of assert in a function, the contract call auto reverts with the provided message if the condition don't match



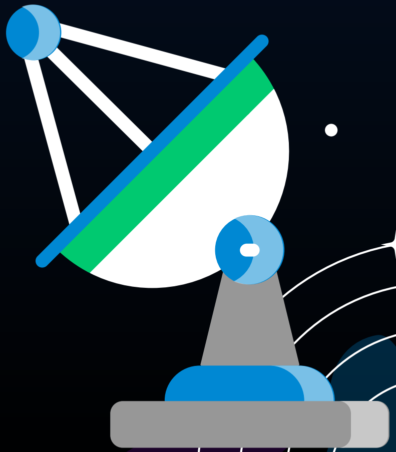
```
contract MyContract {
    function foo() public payable {
        require(
            msg.value > 0.05 ether,
            "You don't send enaugh money"
        );
        // ...
    }
}
```

Where are everyone?!

The `msg` object

When a call it’s performed to our contract we receive information about the call, who did it, if has money with it, how many gas it takes, very useful resources

`msg`.*sender* : address who call us
`msg`.*gas* : gas cost of the transaction
`msg`.*value* : money sended in the transaction
`msg`.*sig* : first 4 bytes of the call, function identifier
`msg`.*data* : the complete calldata



Diamond pattern

That definitely comes from starts

So, 24kb of top for a contract, we can’t update the contract functionality without loose money and information associated to it, don’t sound very nice. In order to fix that: **EIP-2535** and no it’s not a star name

Key concepts

- Fallbacks
- Delegate calls
- Storage/Layout

Leaving the solidity system

Presentation for PDP1 - learn about solidity and smart contracts

Reference Course:
https://coursehunter.net/course/ekskursiya-po-web-3-ethereum-i-smart-kontrakty

PDP GitHub with notes:
https://github.com/qmarquez/coursehunter.ATourWeb3EthereumSmartContracts.lessonsAndNotes

Quimey Lucas Marquez