Développeur blockchain

Solidity

Premières approches

ALYRA

cyril@alyra.fr / benjamin.brucher@alyra.fr

Promo Buterin

Topo du live

- Aperçu généraliste du langage
- N'hésitez pas à poser des questions et a évoquer les difficultés

Le live sur Geth:

Les vidéos seront up dès que possible, après montage et export

Bien avoir en tete que le développement des clients blockchain est un travail de software dev, très différent de ce que l'on va voir par la suite, plus dev web. (Modèle osi, couche applicative)

Je vous demanderai du travail entre ce soir et mardi prochain: lire/faire le cours solidity :)

ALYRA

Solidity

Qu'est ce que c'est?

Langage de programmation orienté objet
Dédié a l'écriture de smart contract

Proposé en 2014 par Gavin Wood
(Polkadot aujourd'hui)

S'exécute sur l'EVM

Langage jeune

ALYRA

Solidity

Développer sur les blockchain

Solidity +- premier langage de smart contract
Utilisé sur Ethereum
Mais aussi sur - Binance Smart Chain,
- Tron,
- Polygon,
- Avalanche…
Et toute blockchain (publiques et privées) basée sur ETH

Un autre langage, ressemblant a python mais moins utilisé: vyper

ALYRA

Solidity

Environnement de dev:

Remix
IDE en ligne, permet de déployer des contrats…
Pas de compte donc pas pratique pour gros projet

Visual Studio Code
IDE de Microsoft, plusieurs extensions, Solidity est bien interprété
Très pratique sous environnement Windows avec WSL2, les terminaux…

ALYRA

Contrat complet

```sol
// SPDX-License-Identifier: GPL-3.0

pragma solidity 0.8.12

import {yes} from "../lib/yes.sol"

contract Main is yes {
    event ChildSet(address indexed child);

    address childManager;
    address private owner;

    constructor{
        owner=msg.sender;
    }

    modifier onlyOwner{
        require(msg.sender==owner, "not the owner");
        _;
    }

    function setChildManager(address _newManager) external onlyOwner{
        require( _newManager != 0x, "not the 0 addr");
        childManager = _newManager;
        emit ChildSet(_newManager);
    }

}
```

License

(éventuellement abi version)

Version de solidity

Import de fichier

Nom du contrat / héritage

Déclaration des événements

Déclaration des variables
constructor qui permet de déclarer que l'owner est la personne qui déploie le contract

Modifier

Condition puis _;

Fonction (nom, parametres, visibilité, modifier)

- conditions
- assign var
- évènement émis

ALYRA

Doc solidity

Très complète (mais en anglais)
Utile pour tous les éléments théoriques
Des exemples
https://docs.soliditylang.org/en/v0.8.18/

ALYRA

Structure d'un smart contract

```
pragma solidity 0.6.12;

contract contractName {

}
```

```
pragma solidity 0.6.12;

library contractName {

}
```

```
pragma solidity 0.6.12;

interface contractName {

}
```

ALYRA

Des déclarations

Variable d'état

```
// SPDX-License-Identifier: GPL-3.0
pragma solidity 0.6.12;

contract SimpleStorage {
    uint public storedData; // State variable
    // ...
}
```

fonction

```
// SPDX-License-Identifier: GPL-3.0
pragma solidity 0.6.12;

contract SimpleAuction {
    function bid() public payable { // Function
        // ...
    }
}
```

Evènements

```
// SPDX-License-Identifier: GPL-3.0
pragma solidity 0.6.12;

contract SimpleAuction {
    event HighestBidIncreased(address bidder, uint amount); // Event

    function bid() public payable {
        // ...
        emit HighestBidIncreased(msg.sender, msg.value); // Triggering event
    }
}
```

ALYRA

Des déclarations

Modifiers

```
// SPDX-License-Identifier: GPL-3.0
pragma solidity 0.6.12;

contract Purchase {
    address public seller;

    modifier onlySeller() { // Modifier
        require(
            msg.sender == seller,
            "Only seller can call this."
        );
        _;
    }

    function abort() public view onlySeller { // Modifier usage
        // ...
    }
}
```

ALYRA

Solidity

```
// SPDX-License-Identifier: GPL-3.0
pragma solidity 0.6.12;

contract Ballot {
    struct Voter { // Struct
        uint weight;
        bool voted;
        address delegate;
        uint vote;
    }
}
```

```
// SPDX-License-Identifier: GPL-3.0
pragma solidity 0.6.12;

contract Purchase {
    enum State { Created, Locked, Inactive } // Enum
}
```

Chaque smart contrat peut contenir des déclarations de :

des structures de données

des énumérations

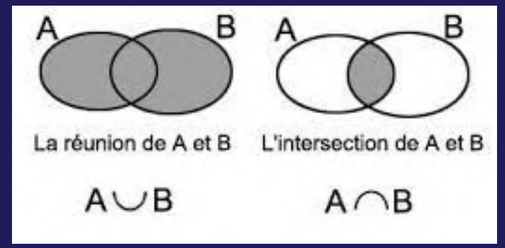
ALYRA

Solidity - Types opérateur stockages

Solidity - types, opérateurs, stockages

- bool — Vrai ou faux
- uint256, uint8, uint16 ... — Entier positif
- int256, int8 ... — Entier
- string — Texte
- address — Address
- bytes — Donnée pure

- ! ou != - Négation / Inégalité
- && - ET logique = intersection
- || - OU logique = réunion
- == - Egalité



- mapping: mapping(address => uint) values; - Paire clef valeur. Pour toute clef une valeur
- array : uint[] values; - Tableau pour indexer des valeurs. N'existent que les valeurs push

ALYRA

17/05/2023

Semaine pro on continue solidity avec de la pratique!

Merci de votre attention!

ALYRA