

day68-sol-inheritance

☰ 태그	
📅 날짜	@2023년 1월 4일

부모 자식 관계

부모에 있는 변수나 함수를 자식이 사용할 수 있다.

```
// SPDX-License-Identifier: GPL-3.0
pragma solidity >= 0.7.0 < 0.9.0;

contract Father{
    string public familyName = "hong";
    string public givenName = "gildong";
    uint256 public money = 100;

    function getFamilyName() view public returns(string memory){
        return familyName;
    }

    function getGivenName() view public returns(string memory){
        return givenName;
    }

    function getMoney() view public returns(uint256){
        return money;
    }
}
// is 뒤에 부모 컨트랙트명을 써주면 상속이 된다.
// Father -> Son
contract Son is Father(){
}
```

FATHER AT 0X712...B2A8C (BLOCK

Balance: 0 ETH

familyName

0: string: hong

getFamily...

0: string: hong

getGivenN...

0: string: gildong

getMoney

0: uint256: 100

givenName

0: string: gildong

money

0: uint256: 100

Low level interactions

CALLDATA

Transact

SON AT 0X813...31903 (BLOCKCH

Balance: 0 ETH

familyName

0: string: hong

getFamily...

0: string: hong

getGivenN...

0: string: gildong

getMoney

0: uint256: 100

```
// SPDX-License-Identifier: GPL-3.0
pragma solidity >= 0.7.0 < 0.9.0;
```

```

contract Father{
    string public familyName = "hong";
    string public givenName = "gildong";
    uint256 public money = 100;

    constructor(string memory _givenName) public{
        givenName = _givenName;
    }

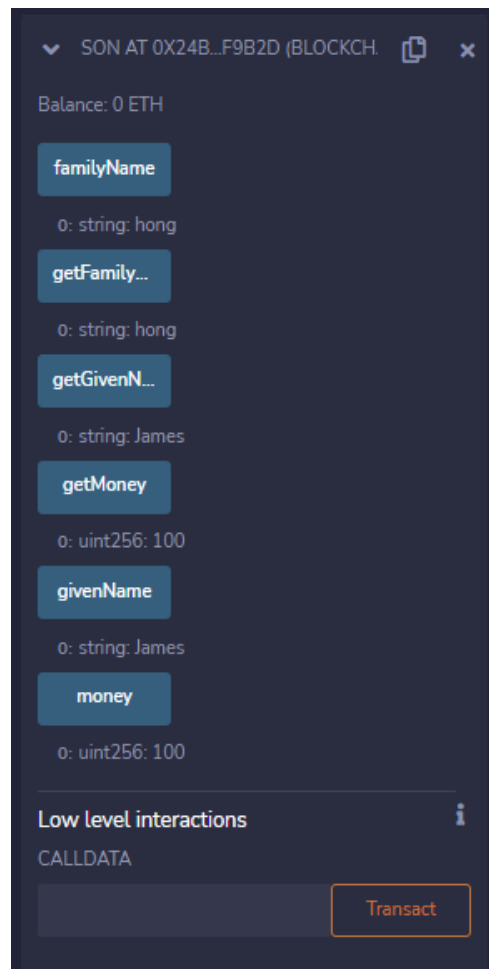
    function getFamilyName() view public returns(string memory){
        return familyName;
    }

    function getGivenName() view public returns(string memory){
        return givenName;
    }

    function getMoney() view public returns(uint256){
        return money;
    }
}
// is 뒤에 부모 컨트랙트명을 써주면 상속이 된다.
// Father -> Son
contract Son is Father("James"){

}

```



상속 오버라이딩 부모 : virtual

상속 오버라이드 자식 : override

```
// SPDX-License-Identifier: GPL-3.0
pragma solidity >= 0.7.0 < 0.9.0;

contract Father{
    string public familyName = "hong";
    string public givenName = "gildong";
    uint256 public money = 100;

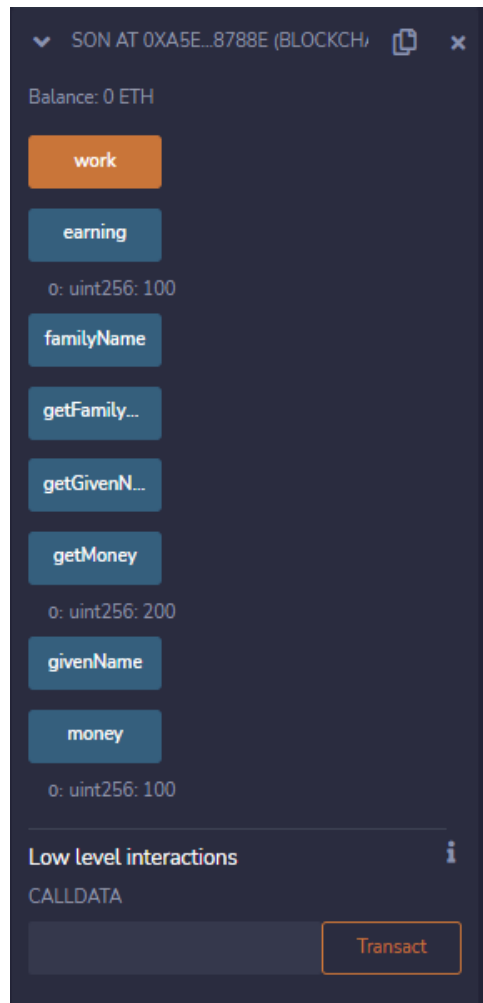
    constructor(string memory _givenName) public{
        givenName = _givenName;
    }

    function getFamilyName() view public returns(string memory){
        return familyName;
    }

    function getGivenName() view public returns(string memory){
        return givenName;
    }

    function getMoney() view public virtual returns(uint256){
        return money;
    }
}
// is 뒤에 부모 컨트랙트명을 써주면 상속이 된다.
// Father -> Son
// 오버라이딩 상속관계 부모와 자식의 함수가 이름이 같아야함
contract Son is Father("James"){
    uint256 public earning = 0;

    function work() public {
        earning += 100;
    }
    // 오버라이딩 상속관계 부모와 자식의 함수가 이름이 같아야함
    function getMoney() view public override returns(uint256){
        return money + earning;
    }
}
```



두개 이상 상속

```
// SPDX-License-Identifier: GPL-3.0
pragma solidity >=0.7.0 <0.9.0;

contract Father{
    uint256 public fatherMoney = 100;
    function getFamilyName() public pure returns(string memory){
        return "HongGilDong";
    }

    function getMoney() public view virtual returns(uint256){
        return fatherMoney;
    }
}

contract Mother{
    uint256 public motherMoney = 500;
    function getMotherName() public pure returns(string memory){
        return "Lesa";
    }

    function getMoney() public view virtual returns(uint256){
```

```

        return motherMoney;
    }
}

//오버라이딩 상속 이름이 같아야한다. 부모 virtual 자식 override
contract Son is Father,Mother{
    function getMoney() public view override(Father,Mother) returns(uint256){
        return fatherMoney+motherMoney;
    }
}

```



이벤트

solidity

event

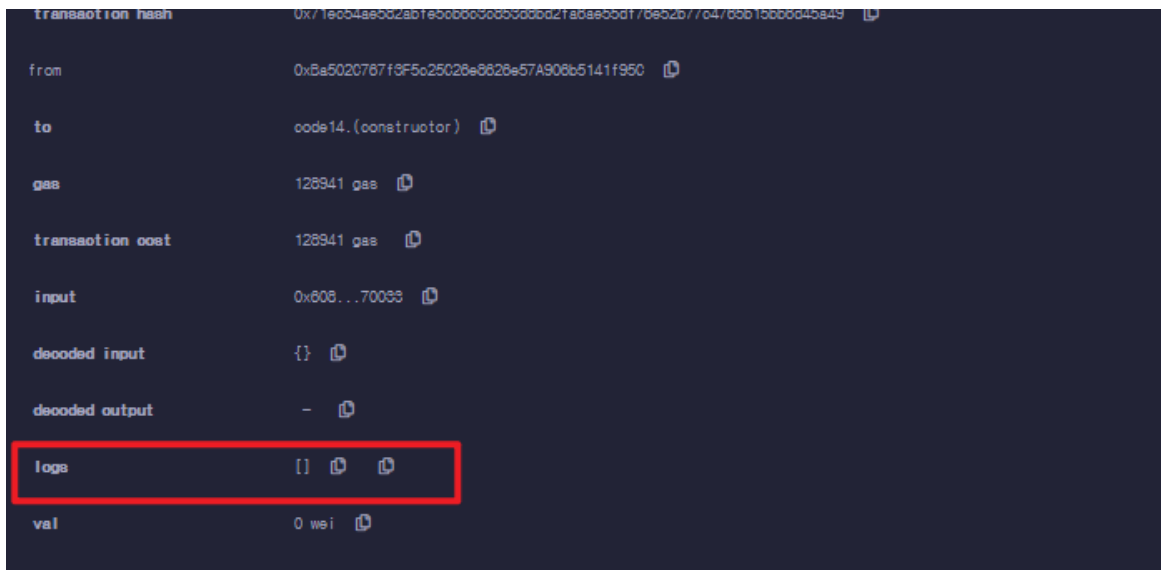
블록체인 네트워크의 블록에 특정값을 기록한다.

A가 B에게 송금했다 → EVENT 기록

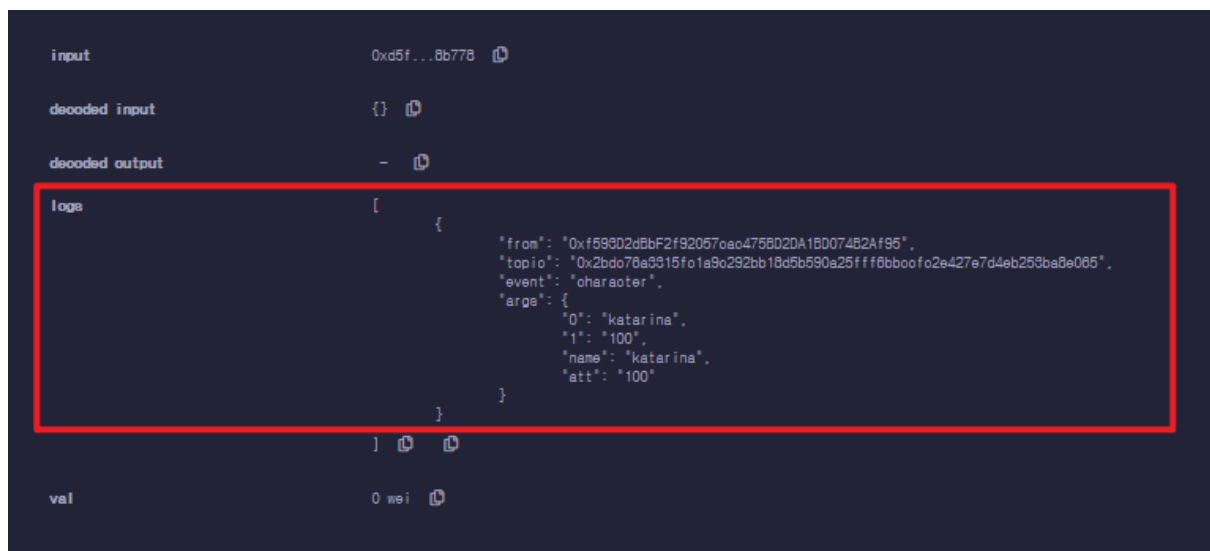
```
// SPDX-License-Identifier: GPL-3.0
pragma solidity >=0.7.0 <0.9.0;

contract code14{
    //event 이벤트의 이름( 필요한 파라미터를 넣는다)
    // event로 선언 emit으로 값을 넣는다. 짝지어서 씀
    event character(string name, uint256 att);

    function SaveCharacter() public {
        emit character("katarina", 100);
    }
}
```



실행 전



실행 후

이벤트 → emit

indexed

내 이벤트를 찾을 때 indexed 를 이용

블록들 안에 출력된 이벤트들을 필터해서 찾을 때 사용할 수 있다.

실습및 연습 복습

부모 컨트랙트

Unit{

종족

이름

미네랄

가스

}

자식 컨트랙트 유닛생성 컨트랙트를 만들어서

UnitCreate {

}

캐리어 생성해보기

리믹스ide에서 출력 해보기

상속사용

```
// SPDX-License-Identifier: GPL-3.0
pragma solidity >=0.7.0 <0.9.0;

//상속만 사용

contract Unit{
    string name = "????";
    string tribe = "????";
    uint256 mineral = 0;
    uint256 gas = 0;

    function getName() public view returns(string memory,string memory, uint256, uint256){
        return (name,tribe, mineral, gas);
    }
}

contract UnitCreate is Unit{
    function setUnit(string memory _cname, string memory _ctrIBE, uint256 _cmineral, uint256 _cgas) public{
        name = _cname;
        tribe = _ctrIBE;
        mineral = _cmineral;
        gas = _cgas;
    }
}
```


UNITCREATE AT 0XBEO...075A5 (BLOCKCHAIN)

Balance: 0 ETH

setUnit

_cname: 캐리어

_ctribe: 프로토스

_cmineral: 300

_cgas: 150

Calldata Parameters **transact**

getName

0: string: 캐리어
1: string: 프로토스
2: uint256: 300
3: uint256: 150

Low level interactions

CALLDATA

Transact

인스턴스만 사용

```
// SPDX-License-Identifier: GPL-3.0
pragma solidity >=0.7.0 <0.9.0;

contract Unit{
    string public tribe = "???";
    string public name = "???";
    uint256 public mineral = 0;
    uint256 public gas = 0;

    function ChangeUnit(string memory _tribe, string memory _name, uint256 _mineral, uint256 _gas) public{
        tribe = _tribe;
        name = _name;
        mineral = _mineral;
        gas = _gas;
    }
}

contract createUnit{
    Unit unit1 = new Unit();

    function Change(string memory Tribe, string memory Name, uint256 Mineral, uint256 Gas) public{
        unit1.ChangeUnit(Tribe, Name, Mineral, Gas);
    }

    function getUnit() public view returns(string memory, string memory, uint256, uint256) {
```

```

    }
    return (unit1.tribe(), unit1.name(), unit1.mineral(), unit1.gas());
}

```

UNIT AT 0X18F...C8156 (BLOCKCHAIN)

Balance: 0 ETH

ChangeUnit

_tribe: 저그

_name: 히드라

_mineral: 75

_gas: 50

Calldata

Parameters

transact

gas

0: uint256: 0

mineral

0: uint256: 0

name

0: string: ????

tribe

0: string: ????

Low level interactions

CALLDATA

Transact

UNIT AT 0X18F...C8156 (BLOCKCHAIN)

Balance: 0 ETH

ChangeUnit

_tribe:

저그

_name:

히드라

_mineral:

75

_gas:

50

Calldata

Parameters

transact

gas

o: uint256: 50

mineral

o: uint256: 75

name

o: string: 히드라

tribe

o: string: 저그

Low level interactions

CALLDATA

Transact