

SEC-205: Distributed Ledger and Blockchain

Lab/Practical Session 3: Events and Loops in Solidity

Charnon Pattiyanon, Ph.D.

charnon@cmkl.ac.th

Assistant Director of IT and Instructor

Department of Artificial Intelligence and Computer Engineering (AiCE)

CMKL University

Add Liking Functions to our Twitter Smart Contract

Count the number of likes when a user likes a tweet.

Events in Solidity

- Events in Solidity are **NOTIFICATIONS** for the blockchain.

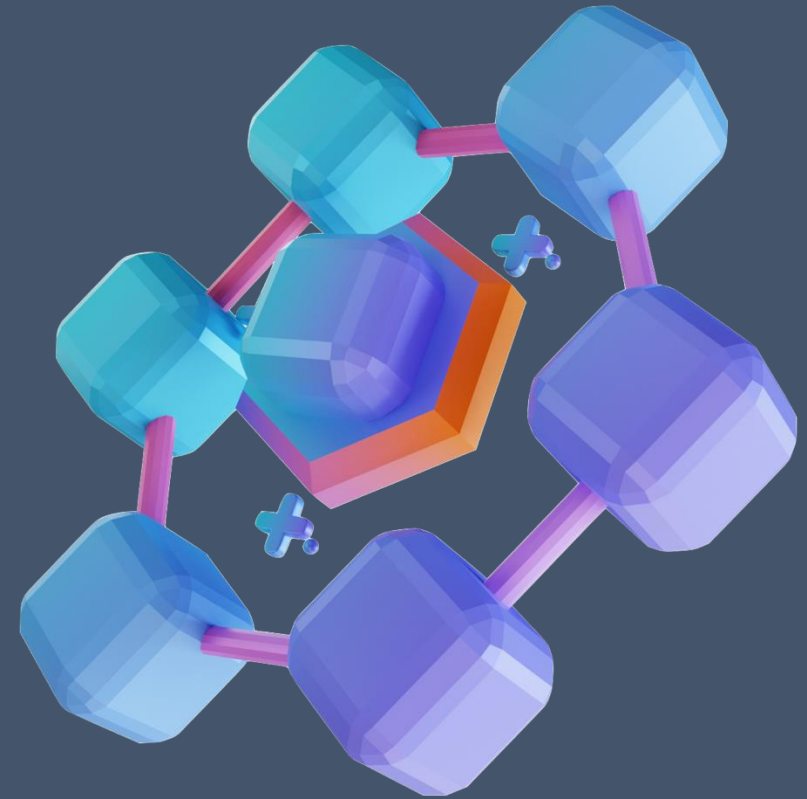
FUNCTION {

Logic...

Event

}

Notifies



Events in Solidity

- Events in Solidity are **NOTIFICATIONS** for the blockchain.



Events in Solidity

- Events in Solidity are **NOTIFICATIONS** for the blockchain. Steps to implement events:
 - **1** Create the event.
 - **2** Emit the event.

```
pragma solidity ^0.8.26

contract EventExample {

    event EventName (uint256 indexed data1, string data2);

    function someFunction () public {
        ...
        emit EventName(data1, data2);
    }
}
```

Events in Solidity (Exercise)

- Add an event to notify
when a new user is
registered successfully.

```
// SPDX-License-Identifier: MIT

pragma solidity ^0.8.26;

contract EventExample {
    // 1 Add an event called "NewUserRegistered" with 2 arguments
    // 👉 user as address type
    // 👉 username as string type
    // CODE HERE 👉

    struct User {
        string username;
        uint256 age;
    }

    mapping(address => User) public users;

    function registerUser(string memory _username, uint256 _age) public {
        User storage newUser = users[msg.sender];
        newUser.username = _username;
        newUser.age = _age;

        // 2 Emit the event with msg.sender and username as the inputs
        // CODE HERE 👉
    }
}
```

Add Events to our Twitter Smart Contract

Create an event for tweet creation and an event for tweet liking.

Loops in Solidity

- Loop is a simple way to create a **repetitive task** in your smart contract.

WHILE

Pedal is pressed

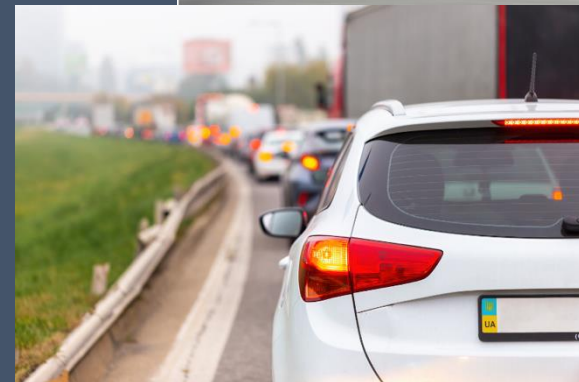


Car goes forward



OTHERWISE

Car stops



Loops in Solidity

- Loop is a simple way to create a repetitive task in your smart contract.

```

    Uint256 i = 0
FOR ( Initialization ; Condition ; Update ) {
    Logic...
}
  
```

Loops in Solidity

- Loop is a simple way to create a repetitive task in your smart contract.

```
// SPDX-License-Identifier: MIT

pragma solidity ^0.8.26;

contract LoopExample {

    function countUp (uint256 limit) public pure return (uint256) {
        uint256 sum = 0;
        for (uint256 i = 1; i <= limit; i++){
            sum += 1;
        }
        return sum;
    }
}
```

Add a Counting Function to Get All Likes for a Specific User to our Twitter Smart Contract

End of the Lab!



Please feel free to ask any questions.

If you need further discussion, please contact me:

- Email me at charnon@cmkl.ac.th
- Appoint me for 1-on-1 discussion during the office hours.