

# Lab: Libraries, Using [...] for [...]

## Prerequisites

1. Chrome or Firefox browser.
2. An Internet connection
3. Open Remix with the following Smart Contract:

```
pragma solidity ^0.5.13;

contract LibrariesExample {

    mapping(address => uint) public tokenBalance;

    constructor() public {
        tokenBalance[msg.sender] = 1;
    }

    function sendToken(address _to, uint _amount) public returns(bool) {

        tokenBalance[msg.sender] -= _amount;
        tokenBalance[_to] += _amount;

        return true;
    }

}
```

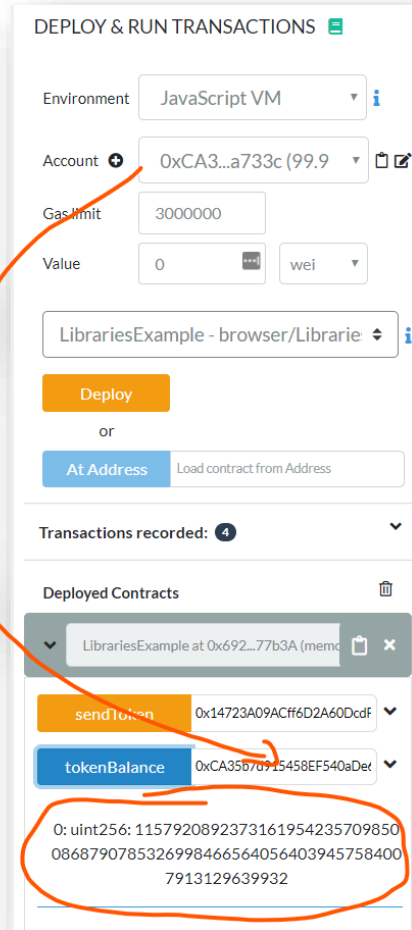
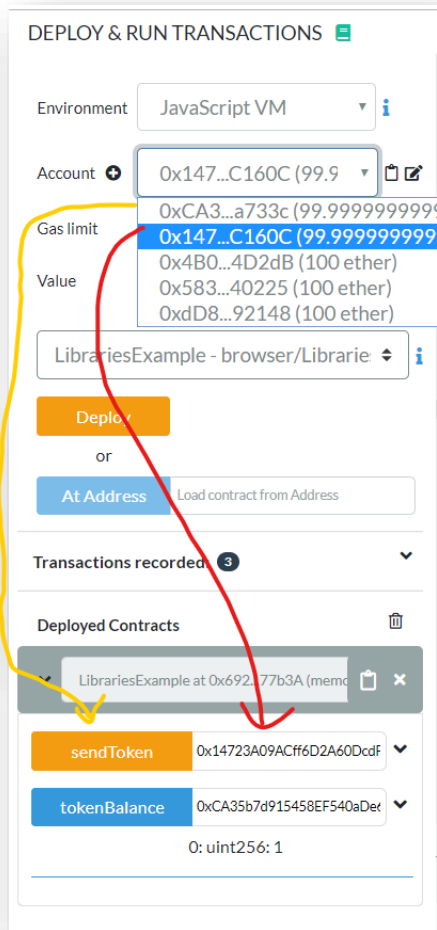
## Step by Step Instruction

*Deploy the Smart Contract in the JavaScript VM*

Open the “Deploy and Run Transactions” view in Remix with the smart contract. Deploy the Smart Contract!

*Send 2 Token to any account*

Send 2 token to any other account than the selected one and observe what happens with the tokenBalance. We already talked about Wrap Around, but there is maybe a better way to tackle this than through a lot of assert statements...



### Import a Library

Now we are going to import a library, such as the “SafeMath” Library from OpenZeppelin:

<https://raw.githubusercontent.com/OpenZeppelin/openzeppelin-contracts/master/contracts/math/SafeMath.sol>

is the raw from this file

<https://github.com/OpenZeppelin/openzeppelin-contracts/blob/master/contracts/math/SafeMath.sol>

In Remix we can conveniently import it into our smart contract:

```
pragma solidity ^0.5.13;

import "https://github.com/OpenZeppelin/openzeppelin-
contracts/contracts/math/SafeMath.sol";

contract LibrariesExample {
    using SafeMath for uint;

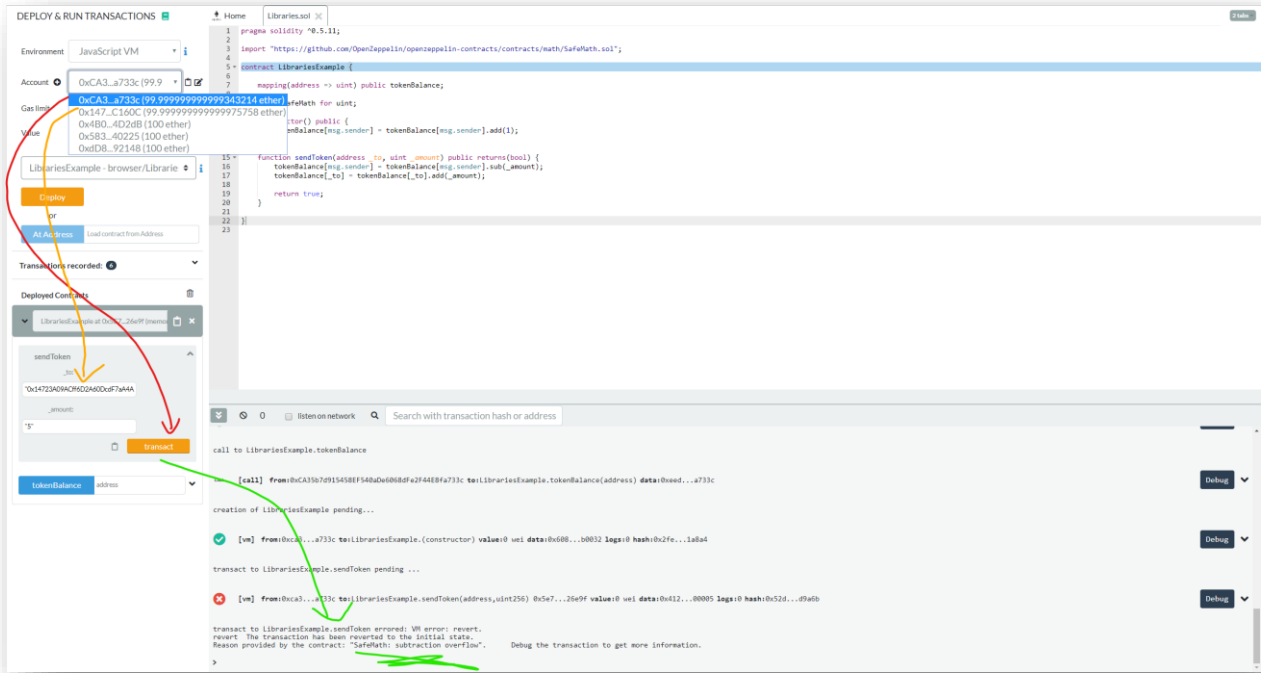
    mapping(address => uint) public tokenBalance;

    constructor() public {
        tokenBalance[msg.sender] = tokenBalance[msg.sender].add(1);
    }

    function sendToken(address _to, uint _amount) public returns(bool) {
        tokenBalance[msg.sender] = tokenBalance[msg.sender].sub(_amount);
        tokenBalance[_to] = tokenBalance[_to].add(_amount);

        return true;
    }
}
```

*Execute the Transfer of 2 tokens again*



You will see it's inherently not possible.

Now, is it good? No! Because the feedback for the user is very “low level” and it could be a better error message, but at least there is no Wrap Around

### Using For vs. Not Using For

Consider this Library and the Smart Contract:

```
pragma solidity >=0.4.16 <0.7.0;

library Search {
    function indexOf(uint[] storage self, uint value)
        public
        view
        returns (uint)
    {
        for (uint i = 0; i < self.length; i++)
            if (self[i] == value) return i;
        return uint(-1);
    }
}

contract NotUsingForExample {
    uint[] data;

    function append(uint value) public {
        data.push(value);
    }

    function replace(uint _old, uint _new) public {
        // This performs the library function call
        uint index = Search.indexOf(data,_old);
        if (index == uint(-1))
            data.push(_new);
        else
            data[index] = _new;
    }
}
```

Here we have one uint[] array, but to search and use the “indexOf” function we need to explicitly call the Library function.

Consider now the following small Change:

```
pragma solidity >=0.4.16 <0.7.0;

library Search {
    function indexOf(uint[] storage self, uint value)
        public
        view
        returns (uint)
    {
        for (uint i = 0; i < self.length; i++)
            if (self[i] == value) return i;
        return uint(-1);
    }
}

contract UsingForExample {
    using Search for uint[];
    uint[] data;

    function append(uint value) public {
        data.push(value);
    }

    function replace(uint _old, uint _new) public {
        // This performs the library function call
        uint index = data.indexOf(_old);
        if (index == uint(-1))
            data.push(_new);
        else
            data[index] = _new;
    }
}
```

As you can see, the small change made it possible to use the library in the context of the uint array itself.

This is directly from the example in the solidity docs:

<https://solidity.readthedocs.io/en/v0.5.13/contracts.html#libraries>

## Congratulations, LAB is completed



---

From the Course “Ethereum Blockchain Developer – Build Projects in Solidity”



FULL COURSE:

<https://www.udemy.com/course/blockchain-developer/?referralCode=E8611DF99D7E491DFD96>