pragma solidity >=0.5.3 <0.6.0;

contract MyToken {

//Public variables of the token

**string** **public** name;

**string** **public** symbol;

/\* This creates an array with all balances \*/

mapping (address => uint256) **public** balanceOf;

**event** **Transfer**(address indexed **from**, address indexed to, uint256 **value**);

/\* Initializes contract with initial supply tokens to the creator of the contract \*/

/\* Initializes contract with initial supply tokens to the creator of the contract \*/

constructor(uint256 initialSupply, **string** memory tokenName, **string** memory tokenSymbol) **public** {

balanceOf[msg.sender] = initialSupply;

// Give the creator all initial tokens

name = tokenName;

// Set the name for display purposes

symbol = tokenSymbol;

// Set the symbol for display purposes

}

/\* Send coins \*/

**function** **transfer**(address \_to, uint256 \_value) **public** **returns** (bool success) {

**require**(balanceOf[msg.sender] >= \_value);

// Check if the sender has enough

**require**(balanceOf[\_to] + \_value >= balanceOf[\_to]);

// Check for overflows

balanceOf[msg.sender] -= \_value;

// Subtract from the sender

balanceOf[\_to] += \_value;

// Add the same to the recipient

/\* Notify anyone listening that this transfer took place \*/

emit **Transfer**(msg.sender, \_to, \_value);

**return** **true**;

}

}