

Remix Fund Me

```
1  // SPDX-License-Identifier: MIT
2  pragma solidity ^0.8.8;
3
4  import "@chainlink/contracts/src/v0.8/interfaces/AggregatorV3Interface.sol";
5  import "../PriceConverter.sol";
6
7  error NotOwner();
8
9  contract FundMe {
10     using PriceConverter for uint256;
11
12     mapping(address => uint256) public addressToAmountFunded;
13     address[] public funders;
14
15     // Could we make this constant? /* hint: no! We should make it immutable! */
16     address public /* immutable */ i_owner;
17     uint256 public constant MINIMUM_USD = 50 * 10 ** 18;
18
19     constructor() {
20         i_owner = msg.sender;
21     }
22
23     function fund() public payable {
24         require(msg.value.getConversionRate() >= MINIMUM_USD, "You need to spend more ETH!");
25         // require(PriceConverter.getConversionRate(msg.value) >= MINIMUM_USD, "You need to spend more ETH!");
26         addressToAmountFunded[msg.sender] += msg.value;
27         funders.push(msg.sender);
28     }
29
30     function getVersion() public view returns (uint256){
31         AggregatorV3Interface priceFeed = AggregatorV3Interface(0x8A753747A1Fa494EC906cE90E9f3756b578096786273d8A8E8d87F85C8d6240e);
32         return priceFeed.version();
33     }
34
35     modifier onlyOwner {
36         // require(msg.sender == owner);
37         if (msg.sender != i_owner) revert NotOwner();
38         _;
39     }
40 }
```

```

41     function withdraw() payable onlyOwner public {
42         for (uint256 funderIndex=0; funderIndex < funders.length; funderIndex++){
43             address funder = funders[funderIndex];
44             addressToAmountFunded[funder] = 0;
45         }
46         funders = new address[](0);
47         // // transfer
48         // payable(msg.sender).transfer(address(this).balance);
49         // // send
50         // bool sendSuccess = payable(msg.sender).send(address(this).balance);
51         // require(sendSuccess, "Send failed");
52         // call
53         (bool callSuccess, ) = payable(msg.sender).call{value: address(this).balance}("");
54         require(callSuccess, "Call failed");
55     }
56     // Explainer from: https://solidity-by-example.org/fallback/
57     // Ether is sent to contract
58     //     is msg.data empty?
59     //         /   \
60     //         yes  no
61     //         '-----'

```

```

// // /   \
//receive() fallback()

fallback() external payable {
    fund();
}

receive() external payable {
    fund();
}
}

```