

## Code With No Effects

### Description

In Solidity, it's possible to write code that does not produce the intended effects. Currently, the solidity compiler will not return a warning for effect-free code. This can lead to the introduction of "dead" code that does not properly performing an intended action.

For example, it's easy to miss the trailing parentheses in `msg.sender.call.value(address(this).balance)("")`;; which could lead to a function proceeding without transferring funds to `msg.sender`. Although, this should be avoided by checking the return value of the call.

### Remediation

It's important to carefully ensure that your contract works as intended. Write unit tests to verify correct behaviour of the code.

### Example:

*Code:*



```
1 pragma solidity ^0.5.0;
2
3 contract DepositBox {
4     mapping(address => uint) balance;
5
6     // Accept deposit
7     function deposit(uint amount) public payable {
8         require(msg.value == amount, 'incorrect amount');
9         // Should update user balance
10        balance[msg.sender] == amount;
11    }
12 }
```

*Explanation:*

Instead of allocating the user's balance to the balance array, the code has verified the user's balance with the balance amount, which has no impact.