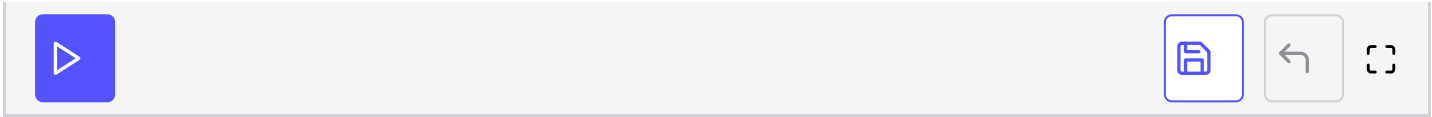
Solution Review: Balanced Brackets

This review explains the solution for the 'Balanced Brackets' exercise.

We'll cover the following

Solution

```
def check_balance(brackets):
    check = 0
    for bracket in brackets:
        if bracket == '[':
            check += 1

        elif bracket == ']':
            check -= 1

        if check < 0:
            break

    return check == 0


bracket_string = '[[[[]]'

print(check_balance(bracket_string))
```



Explanation

The solution relies on the value of the `check` variable, which is updated in each iteration. If an opening bracket is found, `check` is incremented by `1`. In the case of a closing bracket, `check` is decremented by `1`.

The logic is that check should **never** be negative because that would imply that somewhere in the string, there are more closing brackets than opening ones. The condition for being unbalanced is satisfied and we don’t need to check further.

Another case is that after the loop finishes, the value of `check` would be `0` because the brackets match. If it's not, the function simply returns `False`.