

Exercise: Balanced Brackets

Let's check if the brackets in a string are balanced.

We'll cover the following ^

- Problem Statement
- Sample Input
- Sample Output
- Coding Challenge

Problem Statement

Given a string containing **only** square brackets, `[]`, you must check whether the brackets are balanced or not. The brackets are said to be balanced if, for every opening bracket, there is a closing bracket.

You will write your code in the `check_balance()` function, which returns `True` if the brackets are balanced and `False` if they are not.

For an empty string, the function will return `True`.

For the sake of simplicity, you can assume that the string will not contain any other characters.

`[[[]]]]`

Balanced

`[][[[]]]`

Balanced

`[[]]]][[`

Unbalanced

`] [`

Unbalanced

Sample Input

```
"[[[]]]]"
```

Sample Output

```
True
```

Coding Challenge

Take some time to figure out how this problem can be solved using loops. While this problem can be solved in other ways, you should try to use the loop properties we have learned so far.

If you feel stuck, you can always refer to the solution review in the next lesson.

Good luck!

```
def check_balance(brackets): # The argument is a string
    pass # Replace with your code
```

