20. Valid Parentheses

Given a string s containing just the characters '(', ')', '{', '}', '[' and ']', determine if the input string is valid.

An input string is valid if:

Open brackets must be closed by the same type of brackets.

Open brackets must be closed in the correct order.

Every close bracket has a corresponding open bracket of the same type.

Example 1:

Input: s = "()"

Output: true

Example 2:

Input: s = "()[]{}"

Output: true

Example 3:

Input: s = "(]"

Output: false

Example 4:

Input: s = "([])"

Output: true

Constraints:

1 <= s.length <= 104

s consists of parentheses only '()[]{}'.

SOLUTION:



Explanation:

1. We first initialized a stack to keep track of what closing brackets we expect



1. Then we created a mapping of opening and closing brackets

A black background with text and symbols

AI-generated content may be incorrect.

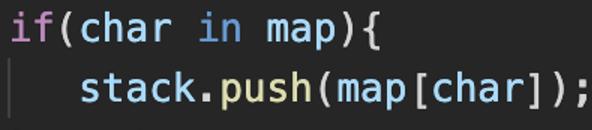
1. We then loop through each character in the string.

A computer code on a black background

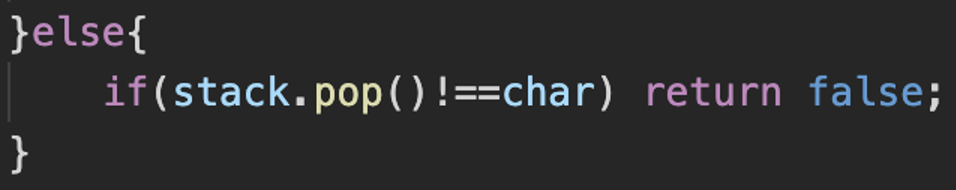
AI-generated content may be incorrect.

* For example, if s = "{[]}", you go one by one: '{', '[', ']', '}'.

1. If the character is a opening bracket we push the expected closing bracket into the stack



1. If the character is the closing bracket we pop it from the stack and check if it matches.



1. We now do the final check i.e. if the stack is empty at the end, it means all brackets matched correctly.

A close up of a logo

AI-generated content may be incorrect.

* Return true if the stack is empty, otherwise return false.

