

Problem:

Valid Parentheses

Total Accepted: **82839** Total Submissions: **298410** Difficulty: **Easy**

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

The brackets must close in the correct order, "()" and "()[]{}" are all valid but "]" and "[()]" are not.

[Subscribe](#) to see which companies asked this question

Algorithm:

we can use a stack to keep track validness. stack = empty

For example the input string is s = "()"

We use a dictionary to store the parentheses, left parentheses are keys and right parentheses are values:

dic = { '(': ')', '[': ']', '{': '}' }

loop through the input string :

if the char is left parentheses, push it into the stack.

if it is not (if it is a right parentheses),

 check if it matches the top char on the stack,

 if it match, pop the stack.

 if it dose not match, return false (for example, "(]")

 if the stack is already empty, return false

after the loop, check if the stack is empty,

 if it is, true

 if not, false

Code:

```
6 class Solution(object):
7     def isValid(self, string):
8         dic = {'(': ')', '[': ']', '{': '}' }
9         stack = []
10
11         for char in string:
12             if dic.get(char) is not None:
13                 stack.append(char)
14             elif dic[stack[-1]] != char or len(stack) == 0:
15                 return False
16             else: # if the the top char on the stack match the current char
17                 stack.pop()
18
19             if len(stack) != 0:
20                 return False
21             else:
22                 return True
```