

Problem Domain
Given an input string of
brackets {}, {}, [],
return a boolean value
that represents if brackets
in string are balanced

Input: [()]
Output: True

Input: [[]]
Output: False

Visualizatation
dictionary: { '[]': {}, '{}': {}, '()' : {} }
(matches)
return True
'[' does not match '}'
return False

Algorithm
Define a dictionary of
key values of the brackets
{} , [], { } .
return a boolean value
that represents if brackets
in string are balanced

for each bracket in input string
check if bracket is a valid
opening bracket and
push on stack if true
check if bracket is a valid
closing bracket +

if it is, search dictionary
to verify is key value
matches and pop off
stack

if stack is empty
returns True
else returns False

Big O()
time $O(n)$
space $O(n)$

Code

```
def multi-bracket-validation(input):  
    def multi-bracket-validation(input):  
        pairs-dict = { '[]': {}, '{}': {}, '()' : {} }  
        bracket-stack = Stack()  
        input-list = list(input)  
        opening-bracket = pairs-dict.keys()  
        closing-bracket = pairs-dict.values()  
  
        for bracket in input-list:  
            if bracket in opening-bracket:  
                bracket-stack.push(bracket)  
            elif bracket in closing-bracket:  
                for key-value in pairs-dict.items():  
                    if value == bracket:  
                        bracket-stack.pop()  
                    else:  
                        return False  
                if bracket-stack.pop():  
                    return True  
                else:  
                    return False
```

Edge Cases
input string empty
returns False