20. Valid Parentheses

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

```python
class Solution(object):
    def isValid(self, s):
        """

        :type s: str

        :rtype: bool
        """

        var_dict = {}

        var_dict['}']='{'

        var_dict[')']='('

        var_dict[']']='['

        stack = list()

        for i in s:

            if i in var_dict.values():

                stack.append(i)

            else:

                if len(stack) !=0:

                    if var_dict.get(i) == stack[len(stack)-1]:

                        stack.pop()

                    else:

                        return False

                else:

                    return False

        if len(stack)==0:

            return True

        else:

            return False
```

```
11          stack = list()
12 ▾        for i in s:
13 ▾            if i in var_dict.values():
14                  stack.append(i)
15 ▾            else:
16 ▾                if len(stack) !=0:
17 ▾                    if var_dict.get(i) == stack[len(stack)-1]:
18                          stack.pop()
19 ▾                    else:
20                          return False
21 ▾                else:
22                      return False
23 ▾        if len(stack)==0:
24              return True
25 ▾        else:
26              return False
```

Your previous code was restored from your local storage. Reset to default

| Testcase | Run Code Result | Debugger 🔒 |

Accepted   Runtime: 24 ms

Your input   `"()"`

Output   true

Expected   true

Success   Details ›

Runtime: **24 ms**, faster than **45.74%** of Python online submissions for Valid Parentheses.

Memory Usage: **12.7 MB**, less than **83.93%** of Python online submissions for Valid Parentheses.

Next challenges:

Generate Parentheses   Longest Valid Parentheses

Remove Invalid Parentheses   Check If Word Is Valid After Substitutions

Show off your acceptance: f 🐦 in

Time Submitted	Status	Runtime	Memory	Language
a few seconds ago	Accepted	24 ms	12.7 MB	python
a minute ago	Accepted	24 ms	12.7 MB	python