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

1. Open brackets must be closed by the same type of brackets.
2. Open brackets must be closed in the correct order.
3. 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

Constraints:

- 1 <= s.length <= 10_4
- s consists of parentheses only '()[]{}'.

Solution:

```
class Solution {
   public boolean isValid(String s) {
      Stack<Character> stack = new Stack<>();
      char[] chars = s.toCharArray();

      for (int i = 0; i < chars.length; i++) {
         char newChar = chars[i];
         if (chars[i] == '(' || chars[i] == '{' || chars[i] == '[') {
            stack.push(chars[i]);
         } else {

            if (stack.isEmpty()) {
               return false;
            }
```

```
                Character popped = stack.pop();
                if (popped == '(') {
                    if (newChar != ')') {
                        return false;
                    }
                } else if (popped == '{') {
                    if (newChar != '}') {
                        return false;
                    }
                } else if (popped == '[') {
                    if (newChar != ']') {
                        return false;
                    }
                }
            }
        }
        return stack.isEmpty();
    }
}
```