□□□□□□□□ '('□')'□'{'□'}'□'['□']' □□□□□□□□□□□□□□□□□□

□□□□□□□□□

1. □□□□□□□□□□□□□□□□□□□
2. □□□□□□□□□□□□□□□□

□□□□□□□□□□□□□□□□□□□□

□□ 1:

: "()"
: true

□□ 2:

: "()[]{}"
: true

□□ 3:

: "(]"
: false

□□ 4:

: "([)]"
: false

□□ 5:

: "{[]}"
: true

□□□

□□□□□□□□□□□□□□□□□□□□□□□□

□□□□□□□□□Java□□□□□□□□import java.util.Stack;

□□□□□□□□□□□□

- push—□□□
- pop—□□
- isEmpty—□□□□□□□
- peek—□□□□□□

□□□□□□□□□□□□□□□□□□□□(),{}|,[]□

□□

□□□□□□□□□□□□□□□□□□

□□□□□□□□□□□□□□□□□□□

□□□□□□□□□□

- □□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□
- □□
- □□□□□□□□□□□□□□□□□□□□true□□□□□□□□false;

□□□□□□□□

```java
import java.util.Stack;
class Solution {
    public boolean isValid(String s) {
        Stack<Character> stack = new Stack<>();
        char[] chars = s.toCharArray();
        for (char aChar : chars) {
            if (stack.size() == 0) {
                stack.push(aChar);
            } else if (isSym(stack.peek(), aChar)) {
                stack.pop();
            } else {
                stack.push(aChar);
            }
        }
        return stack.size() == 0;
    }

    private boolean isSym(char c1, char c2) {
        return (c1 == '(' && c2 == ')') || (c1 == '[' && c2 == ']') || (c1 == '{' && c2 ==
    }
}
```

□□□

- char□□□□□□□□□□Character;
- □□□□□□□□□□□□□stack.size()
- □□□□□□□□□□□□char[] chars = s.toCharArray();