

Code Challenge



Multi-bracket Validation.

Specifications

- Read all of these instructions carefully. Name things exactly as described.
- Do all your work in a public repository called `data-structures-and-algorithms`, with a well-formatted, detailed top-level README.md.
- Create a new branch in your repo called `multi_bracket_validation`.
- Your top-level readme should contain a “Table of Contents” navigation to all of your challenges and implementations so far. (Don’t forget to update it!)
- This assignment should be completed within the `challenges` subdirectory of the repository.
- On your branch, create...
 - **C#:** a new .NET Core console project named `MultiBracketValidation`. Within your `Program.cs` create a new static method outside of `Main()` following the naming conventions below. Call your newly created method in `Main()` once complete.
 - **JavaScript:** a folder named `multiBracketValidation` which contains a file called `multi-bracket-validation.js`
 - **Python:** a folder named `multi_bracket_validation` which contains a file called `multi_bracket_validation.py`
 - **Java:** a folder named `multibracketvalidation` which contains a file called `MultiBracketValidation.java`
- Include any language-specific configuration files required for this challenge to become an individual component, module, library, etc.
 - *NOTE: You can find an example of this configuration for your course in your class lecture repository.*

Feature Tasks

- On your main file, create...
 - **C#:** a method called `public static bool MultiBracketValidation(string input)`
 - **JavaScript:** a function called `multiBracketValidation(input)`
 - **Python:** a function called `multi_bracket_validation(input)`
 - **Java:** a method `public static boolean multiBracketValidation(String input)`

Your function should take a string as its only argument, and should return a boolean representing whether or not the brackets in the string are balanced. There are 3 types of brackets:

- Round Brackets : `()`
- Square Brackets : `[]`
- Curly Brackets : `{}`

Example

Input	Output
<code>{}</code>	<code>TRUE</code>
<code>{ } { }</code>	<code>TRUE</code>
<code>() [[Extra Characters]]</code>	<code>TRUE</code>
<code>() {} [[]]</code>	<code>TRUE</code>
<code>{{Code}}[Follows]()</code>	<code>TRUE</code>
<code>[({}]</code>	<code>FALSE</code>
<code>()(</code>	<code>FALSE</code>
<code>{(})</code>	<code>FALSE</code>

Consider these small examples and why they fail.

Input	Output	Why
<code>{</code>	<code>FALSE</code>	error unmatched opening <code>{</code> remaining.
<code>)</code>	<code>FALSE</code>	error closing <code>)</code> arrived without corresponding opening.
<code>[]</code>	<code>FALSE</code>	error closing <code>}</code> . Doesn't match opening <code>(</code> .

Requirements

Ensure your complete solution follows the standard requirements.

1. Write [unit tests](#)
2. Follow the [template for a well-formatted README](#)
3. Submit the assignment following [these instructions](#)

