**Find Characters with Lazy Matching**

With regular expressions, we have two types of matching : greedy one which finds the longest part of a string and the lazy one that finds the smallest possible part of that string.  
  
You can apply the regex /t[a-z]\*i/ to the string "titanic". This regex is basically a pattern that starts with t, ends with i, and has some letters in between.

Regular expressions are by default greedy, so the match would return ["titani"]. It finds the largest sub-string possible to fit the pattern.

However, you can use the ? character to change it to lazy matching. "titanic" matched against the adjusted regex of /t[a-z]\*?i/ returns ["ti"].

**Note:** Parsing HTML with regular expressions should be avoided, but pattern matching an HTML string with regular expressions is completely fine.

Example :

Fix the regex /<.\*>/ to return the HTML tag <h1> and not the text "<h1>Winter is coming</h1>". Remember the wildcard . in a regular expression matches any character.

let text = "<h1>Winter is coming</h1>";

let myRegex = /<[a-z0-9]\*?>/; // Change this line

let result = text.match(myRegex);