**Reuse Patterns Using Capture Groups**

Say you want to match a word that occurs multiple times like below.

let repeatStr = "row row row your boat";

You could use /row row row/, but what if you don't know the specific word repeated? *Capture groups* can be used to find repeated substrings.

Capture groups are constructed by enclosing the regex pattern to be captured in parentheses. In this case, the goal is to capture a word consisting of alphanumeric characters so the capture group will be \w+ enclosed by parentheses: /(\w+)/.

The substring matched by the group is saved to a temporary "variable", which can be accessed within the same regex using a backslash and the number of the capture group (e.g. \1). Capture groups are automatically numbered by the position of their opening parentheses (left to right), starting at 1.

The example below matches a word that occurs thrice separated by spaces:

let repeatRegex = /(\w+) \1 \1/;

repeatRegex.test(repeatStr); // Returns true

repeatStr.match(repeatRegex); // Returns ["row row row", "row"]

Using the .match() method on a string will return an array with the matched substring, along with its captured groups.

Example :

Use capture groups in reRegex to match a string that consists of only the same number repeated exactly three times separated by single spaces.  
  
let repeatNum = "42 42 42";

let reRegex = /^(\d+)\s\1\s\1$/g; // Change this line

let result = reRegex.test(repeatNum);