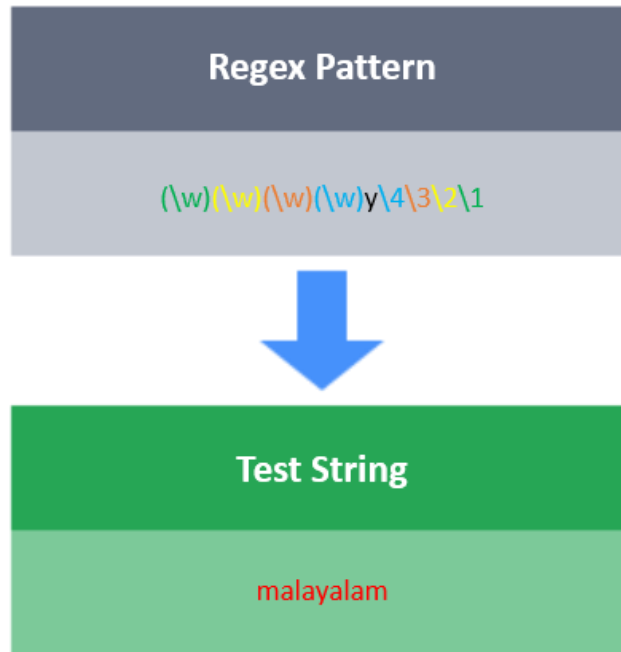


Matching Same Text Again & Again

`\group_number`

This tool (`\1` references the first capturing group) matches the same text as previously matched by the capturing group.



In the above image, the Regex Pattern is matched with the Test String.

For Example:

`(\d)\1`: It can match `00`, `11`, `22`, `33`, `44`, `55`, `66`, `77`, `88` or `99`.

Task

You have a test string `S`.

Your task is to write a regex that will match `S` with the following conditions:

- `S` must be of length: `20`
- `1st` character: `lowercase letter`.
- `2nd` character: `word character`.
- `3rd` character: `whitespace character`.
- `4th` character: `non word character`.
- `5th` character: `digit`.
- `6th` character: `non digit`.

- 7th character: `uppercase letter`.
- 8th character: `letter` (either lowercase or uppercase).
- 9th character: `vowel` (a, e, i, o, u, A, E, I, O or U).
- 10th character: `non whitespace character`.
- 11th character: should be same as `1st character`.
- 12th character: should be same as `2nd character`.
- 13th character: should be same as `3rd character`.
- 14th character: should be same as `4th character`.
- 15th character: should be same as `5th character`.
- 16th character: should be same as `6th character`.
- 17th character: should be same as `7th character`.
- 18th character: should be same as `8th character`.
- 19th character: should be same as `9th character`.
- 20th character: should be same as `10th character`.

Note

This is a regex only challenge. You are not required to write code.
You have to fill the regex pattern in the blank ().