**Positive and Negative Lookahead**

*Lookaheads* are patterns that tell JavaScript to look-ahead in your string to check for patterns further along. This can be useful when you want to search for multiple patterns over the same string.

There are two kinds of lookaheads: *positive lookahead* and *negative lookahead*.

A positive lookahead will look to make sure the element in the search pattern is there, but won't actually match it. A positive lookahead is used as (?=...) where the ... is the required part that is not matched.

On the other hand, a negative lookahead will look to make sure the element in the search pattern is not there. A negative lookahead is used as (?!...) where the ... is the pattern that you do not want to be there. The rest of the pattern is returned if the negative lookahead part is not present.

Lookaheads are a bit confusing but some examples will help.

let quit = "qu";

let noquit = "qt";

let quRegex= /q(?=u)/;

let qRegex = /q(?!u)/;

quit.match(quRegex);

noquit.match(qRegex);

Both of these match calls would return ["q"].

A more practical use of lookaheads is to check two or more patterns in one string. Here is a (naively) simple password checker that looks for between 3 and 6 characters and at least one number:

let password = "abc123";

let checkPass = /(?=\w{3,6})(?=\D\*\d)/;

checkPass.test(password);  
  
Example :

Use lookaheads in the pwRegex to match passwords that are greater than 5 characters long, and have two consecutive digits.

let sampleWord = "astronaut";

let pwRegex = /(?=\w{6,})(?=\D\*\d{2})/; // Change this line

let result = pwRegex.test(sampleWord);