

03-AnswersEvenMoreRegex\script.js

```
1 // Password strength:
2 // function checkPasswordStrength(password) {
3 //   // Regular expression to match strong password with at least one digit
4 //   const strongRegex = /^(?=.*\d){8,}$/;
5
6 //   // Regular expression to match medium password without any digits
7 //   const mediumRegex = /^[^\d]{8,}$/;
8
9 //   if (strongRegex.test(password)) {
10 //     return "Strong";
11 //   } else if (mediumRegex.test(password)) {
12 //     return "Medium";
13 //   } else {
14 //     return "Weak";
15 //   }
16 // }
17
18 function checkPasswordStrength(password) {
19   // Check if password contains at least one digit
20   const containsDigit = /\d/.test(password);
21
22   // Check if password length is at least 8 characters
23   const isLengthValid = password.length >= 8;
24
25   // Determine strength based on conditions
26   if (containsDigit && isLengthValid) {
27     debugger;
28     return "Strong";
29   } else if (isLengthValid) {
30     return "Medium";
31   } else {
32     return "Weak";
33   }
34 }
35
36 // OK. In this version, we just explicitly check if the password contains a digit and then
37 // check to make sure it's at least 8 characters and then put the conditionals together to
38 // generate "Strong", "Weak" or "Medium"
39
40 // Test cases
41 console.log(checkPasswordStrength("password")); // Medium
42 console.log(checkPasswordStrength("abcdefg")); // Weak
43 console.log(checkPasswordStrength("secure123")); // Strong
44 console.log(checkPasswordStrength("12345678")); // Strong
45 console.log(checkPasswordStrength("pass1")); // Weak
46 console.log(checkPasswordStrength("VerySecurePassword123")); // Strong
47
48 // 2. Username Validator
```

```
47 function isValidUsername(username) {
48     // Regular expression to match valid username formats
49     const regex = /^[a-zA-Z\d](?!.*[._-]{2,})[\w.-]{2,19}[a-zA-Z\d]$/;
50
51     // Test if the username matches the regex
52     return regex.test(username);
53 }
54
55 // function isValidUsername(username) {
56 //     // Regular expression to match valid username formats
57 //     const regex = /^[a-zA-Z\d][\w.-]*[a-zA-Z\d]$/;
58
59 //     // Test if the username matches the regex
60 //     return regex.test(username) && !/^(^\.|\.$|_{2,}|-{2,}|\.+\.+*|_+_*|_+|-+|-+
61 //     *)/.test(username);
62 // }
63 // The look ahead makes sure we don't have back to back periods, hyphens, etc. We can also
64 // explicitly check for that.
65
66 // Test cases
67 console.log(isValidUsername("user123")); // true
68 console.log(isValidUsername("john.doe")); // true
69 console.log(isValidUsername("user_name")); // true
70 console.log(isValidUsername("user-name")); // true
71 console.log(isValidUsername("username_123")); // true
72 console.log(isValidUsername("user..name")); // false (consecutive periods)
73 console.log(isValidUsername("user--name")); // false (consecutive hyphens)
74 console.log(isValidUsername("user__name")); // false (consecutive underscores)
75 console.log(isValidUsername("_username")); // false (starts with underscore)
76 console.log(isValidUsername(".username")); // false (starts with period)
77 console.log(isValidUsername("-username")); // false (starts with hyphen)
78 console.log(isValidUsername("username_")); // false (ends with underscore)
79 console.log(isValidUsername("username.")); // false (ends with period)
80 console.log(isValidUsername("username-")); // false (ends with hyphen)
81 console.log(isValidUsername("a")); // false (less than 3 characters)
82 console.log(isValidUsername("very_long_username_with_more_than_20_characters")); // false
83 // (more than 20 characters)
84
85 // Test cases
86 console.log(isValidUsername("user 123")); // true
87 console.log(isValidUsername("user123")); // true
88 console.log(isValidUsername("john.doe")); // true
89 console.log(isValidUsername("user_name")); // true
90 console.log(isValidUsername("user-name")); // true
91 console.log(isValidUsername("username_123")); // true
92 console.log(isValidUsername("user..name")); // false (consecutive periods)
93 console.log(isValidUsername("user--name")); // false (consecutive hyphens)
94 console.log(isValidUsername("user__name")); // false (consecutive underscores)
95 console.log(isValidUsername("_username")); // false (starts with underscore)
96 console.log(isValidUsername(".username")); // false (starts with period)
```

```
95 console.log(isValidUsername("-username")); // false (starts with hyphen)
96 console.log(isValidUsername("username_")); // false (ends with underscore)
97 console.log(isValidUsername("username.")); // false (ends with period)
98 console.log(isValidUsername("username-")); // false (ends with hyphen)
99 console.log(isValidUsername("a")); // false (less than 3 characters)
100 console.log(isValidUsername("very_long_username_with_more_than_20_characters")); // false
    (more than 20 characters)
101
```