Here's a table that outlines common **regular expressions** (regex) in JavaScript, along with their syntax, examples, and explanations:

| **Pattern** | **Syntax** | **Example** | **Explanation** |
| --- | --- | --- | --- |
| **Literal Characters** | /abc/ | /hello/ | Matches the exact characters in the pattern. |
| **Wildcard (any character)** | /a.b/ | /a.b/ → Matches aab, acb, a1b, etc. | The dot (.) matches any character except newlines. |
| **Character Set** | /[abc]/ | /[aeiou]/ → Matches any vowel | Matches any single character in the set [abc]. |
| **Negated Character Set** | /[^abc]/ | /[^0-9]/ → Matches any character except a digit | Matches any character not in the set [abc]. |
| **Character Range** | /[a-z]/ | /[A-Za-z]/ → Matches any letter (lowercase or uppercase) | Matches any letter in the specified range. |
| **Quantifiers** | /a{2,4}/ | /o{2,3}/ → Matches oo or ooo | Matches between 2 and 4 occurrences of a. |
| **Optional** | /a?/ | /colou?r/ → Matches color or colour | The preceding character or group is optional (matches 0 or 1). |
| **Zero or More** | /a\*/ | /ba\*/ → Matches b, ba, baa, baaa, etc. | Matches 0 or more occurrences of a. |
| **One or More** | /a+/ | /ba+/ → Matches ba, baa, baaa, etc. | Matches 1 or more occurrences of a. |
| **Anchors (Start/End of String)** | /^a/ and /a$/ | /^hello/ → Matches hello only at the beginning | ^ matches the start of the string, $ matches the end. |
| **Word Boundaries** | /\bword\b/ | /\bcat\b/ → Matches the word cat as a whole word | \b matches word boundaries (like spaces or punctuation). |
| **Escape Special Characters** | /\\. | /\\d/ → Matches the literal character \d | Escape special characters like . (dot), \*, +, etc. |
| **Digit** | /\d/ | /\d+/ → Matches 123, 4567, etc. | \d matches any digit (equivalent to [0-9]). |
| **Non-Digit** | /\D/ | /\D+/ → Matches abc, hello, etc. | \D matches any non-digit character. |
| **Whitespace** | /\s/ | /\s+/ → Matches any sequence of whitespace characters | \s matches spaces, tabs, and line breaks. |
| **Non-Whitespace** | /\S/ | /\S+/ → Matches any non-whitespace characters | \S matches any non-whitespace character. |
| **Word Character** | /\w/ | /\w+/ → Matches words like hello, abc123 | \w matches any word character (letters, digits, and \_). |
| **Non-Word Character** | /\W/ | /\W+/ → Matches any non-word characters like !@# | \W matches anything that's not a word character. |
| **Groupings** | `/a(bc | de)/` | `/a(bc |
| **Lookahead (Positive)** | /(?=abc)/ | /\d(?=\D)/ → Matches a digit followed by a non-digit | Positive lookahead (?=) asserts that a given pattern follows. |
| **Lookahead (Negative)** | /[^abc](?=xyz)/ | /\d(?!\D)/ → Matches a digit that is not followed by a non-digit | Negative lookahead (?!) asserts that a given pattern does not follow. |
| **Replace Example** | string.replace(regex, replacement) | "hello world".replace(/\bworld\b/, "everyone") → hello everyone | Replaces a match with a given string. |

**Additional Examples:**

1. **Matching a phone number (xxx-xxx-xxxx)**:

const regex = /^\d{3}-\d{3}-\d{4}$/;

console.log(regex.test('123-456-7890')); // true

1. **Validating an email**:

const emailRegex = /^[a-zA-Z0-9.\_%+-]+@[a-zA-Z0-9.-]+\.[a-zA-Z]{2,}$/;

console.log(emailRegex.test('test@example.com')); // true

1. **Matching a URL**:

const urlRegex = /^(https?:\/\/)?([a-zA-Z0-9.-]+)(\.[a-zA-Z]{2,})/;

console.log(urlRegex.test('http://example.com')); // true

Regular expressions in JavaScript are quite powerful and can be used for a variety of string matching, searching, and manipulation tasks.