Regular Expression (Shorthand character classes)

→ These shorthand classes include:

→ \w: This is the "word character" class that represents the regex range [A-Za-z0-9_], and it will match a single uppercase character, lower-case character, digit, or underscore.

→ \d: This is the "digit character" class represents the regex range [0-9], and it will match the single-digit character.

→ \s: This is the "whitespace character" class that represents the regex range, matching a single space, ~~carriage~~ carriage return, tab, line break, form feed, or vertical tab.

Along with shorthand character classes (\w, \d, and \s), we can also use the negated shorthand character classes. These negated shorthands will match any character that is NOT in regular shorthand classes. These negated shorthand classes include:

→ \W: the "non-word character" class represents the regex range [^A-Za-z0-9_], matching any character that is not

included in the range represented by \w.

→ \D: the "non-digit character" class represents the regex range [^0-9], matching any character that is not included in the range represented by \d.

→ \S: the "non-whitespace character" class represents the regex range [^ \t\n\r\f\v], matching any character that is not included in the range represented by \s.

Character Classes (Shorthand:)

Character	DESCRIPTION	Example
→ \d	This character means any digit character; functionally equivalent to [0-9] or [[:digit:]]	\d means 1, 56, 77 etc., but not 1-7 or aa bb 6. Once or more of any digit characters.
→ \D	This character means any non-digit character; functionally equivalent to [^0-9] or [^[:digit:]]	\D matches a, ab, abcd, ab-b, but not 1.

\w	This character means any "word" character. That is any alphanumeric char. and its functionally equivalent to [_A-Za-z0-9] or [_[:ALNUM:]]	\w matches a, ab, a1, ab-c123 but not !2. One or more upper or lower case letters or digits, but not punctuation or other special symbols/characters.
\W	This character means any non-alphanumeric char.; functionally equivalent to [^_A-Za-z0-9] or [^_[:ALNUM:]]	\W matches *, &, but not SPACE or 41. One or more of any char. but upper or lower-case letters and digits.
\s	This character means any white space character; space, new line, tab, non-breaking space, etc.; functionally equivalent to [[:SPACE:]]	Vegetable\s matches "Vegetable" followed by any non-white space character.
\S	This character means any non-whitespace char.; anything other than space, newline, tab, non-breaking space, etc.; functionally equivalent to [^[:SPACE]]	Vegetable\S matches "vegetable" followed by any non-whitespace character.

Date

→ Assertions:

→ An assertion is a regular expression that will succeed if a match is found and fails if a match is not found. Assertion consists of Anchors and Lookarounds.

→ ^ : The symbol "^" matches at the beginning of the string.

→ $: The symbol "$" matches only at the end of the string.

→ \b : The character "\b" matches only at a word boundary.

→ \B : The character "\B" matches only if not at a word boundary.

→ (?= <<PATTERN>>) : This is a positive lookahead. It matches the regular expression with the pattern only if the pattern matches what comes next. The pattern is used only to look ahead but otherwise ignored.

→ (?! <<PATTERN>>) : This is the negative lookahead. The pattern is used only. It matches the regular expression with the pattern only if the pattern does not match what

comes next. The pattern is used only to look ahead but otherwise ignored.