REGEX - Regular Expression

Regular Expressions are patterns used to match character combinations in strings.
Examples:

- Validate an email
- Extract numbers
- Replace specific patterns in a string

Java Regex Classes (from java.util.regex)

Class	Purpose
Pattern	Compiles a regex into a pattern
Matcher	Matches a pattern against a string
PatternSyntaxException	Handles regex errors

Pattern	Meaning	Example Match
.	Any character except newline	a, 1, @
[abc]	a or b or c	a, b
[^abc]	Not a, b, or c	d, x
[a-z]	Any lowercase letter	g, z
\\d	Any digit (0â€“9)	3, 9
\\w	Word character (aâ€“z, Aâ€“Z, 0â€“9, _)	A, 7, _
\\s	Whitespace	space, tab
^	Start of string	^abc matches abc123
$	End of string	abc$ matches 123abc
a+	One or more 'a'	a, aa, aaa
a*	Zero or more 'a'	"", a, aa
a?	Zero or one 'a'	"", a
a{3}	Exactly 3 'a's	aaa
a{2,4}	2 to 4 'a's	aa, aaa, aaaa
\.	Match literal dot	.

Regex Pattern	Meaning
[xyz]	x, y, or z
[^xyz]	Any character except x, y, or z
[a-zA-Z]	Any alphabet character
[a-f[m-t]]	a to f OR m to t
[a-z && [p-y]]	Characters common to both ranges (p to y)
[a-z && [^bc]]	a to z excluding b and c
[a-z && [^m-p]]	a to z excluding m to p

Quantifiers in Java Regex:

Quantifiers control the number of repetitions of characters or groups in regex patterns.

Quantifier	Meaning	Example Pattern	Matches
*	0 or more times	a*	"", a, aaa
+	1 or more times	a+	a, aa
?	0 or 1 time(optional)	a?	"", a

{n}	Exactly n times	a{3}	aaa
{n,}	At least n times	a{2,}	aa, aaaa
{n,m}	Between n and m times	a{2,4}	aa, aaa, aaaa

Why Use Pattern and Matcher in Java?
The Pattern and Matcher classes from java.util.regex package are used for:

âœ… Searching within strings
âœ… Validating patterns (emails, passwords, phone numbers, etc.)
âœ… Extracting and replacing matching parts
âœ… Splitting strings using regex

1. Pattern Class â€“ Compiles the regex

Common Methods of Pattern:

Method	Description
compile(String regex)	Compiles regex into a Pattern object
compile(String regex, int flags)	Compiles regex with case-insensitive or multiline options (Pattern.CASE_INSENSITIVE)
matcher(CharSequence input)	Creates a Matcher for the input string
pattern()	Returns the regex string used
split(CharSequence input)	Splits string around matches
flags()	Returns flags (like case sensitivity) used in compile

2. Matcher Class â€“ Applies pattern to a string

Common Methods of Matcher:

Method	Description
matches()	Checks if entire string matches the pattern
find()	Finds next subsequence that matches
group()	Returns the current matched group
group(int)	Returns a specific group if regex uses capturing groups ( )
start() / end()	Start and end index of current match
lookingAt()	Checks if match starts at beginning (partial match)
replaceAll(String)	Replaces all matches
replaceFirst(String)	Replaces first match
reset()	Resets matcher for re-use

Examples:

âœ… 1. Email Validation
ðŸŽ¯ Task:
Validate a proper email like:
user.name123@example-domain.com

âœ… Java Regex:

```
String emailRegex = "^[a-zA-Z0-9._%+-]+@[a-zA-Z0-9.-]+\\.[a-zA-Z]{2,6}$";
```

âœ… Explanation:
^ â†’ start of string

[a-zA-Z0-9._%+-]+ â†’ username: letters, numbers, dots, dashes

@ â†’ must include @

[a-zA-Z0-9.-]+ â†’ domain part

\\. â†’ literal dot

[a-zA-Z]{2,6} â†’ domain extension (like .com, .org)

$ â†’ end of string