Basics & Meta Characters	
Goal: Understand what regex is and learn the fundamental building blocks.	
.	Any new character except \n
^	Start of String
$	End of String
\d,\D	Digit/Non Digit
\w,\W	Word Char/Non Word Char
\s,\S	Space / Non-space
[ ]	Character Set
|	OR

Quantifiers		
Goal: Learn how to match repeated patterns.		
*	0 or more	Matches the preceding character/pattern **zero or more times**.
+	1 or more	Matches the preceding character/pattern **at least once**.
?	0 or 1	Matches the preceding character/pattern **zero or one time**.
{n}	Exactly n	Matches the preceding character/pattern **exactly n times**.
{n,}	n or more	Matches the preceding character/pattern **at least n times**.
{n,m}	Between n and m	Matches the preceding character/pattern **between n and m times** (inclusive).

Groups & Capturing		
Goal: Extract specific parts from matches.		
()	Capturing Group	Captures the matched text inside the parentheses.
(?:)	Non-capturing group	Groups things **without** capturing the match in `.groups()`.
(?P<name>)	Named group	Gives the group a name for easy access.
`.group(),`		
`.groups(),`		
`.groupdict()`		

Anchors, Boundaries & Flags		
Goal: Control where matches happen and tweak behavior.		
^,$	Anchors	match **positions**
\b	Boundaries	word boundary (transition between word char and non-word char)
\B		NOT a word boundary
re.IGNORECASE		(case-insensitive)
re.MULTILINE		^ and $ apply to every line
re.DOTALL		`. matches newlines too`

Lookarounds (Advanced)		
Goal: Match patterns based on what comes before/after without including it.		
(?=)	Positive Lookahead	Matches something **only if it is followed by** another pattern.
(?!)	Negative Lookahead	Matches something **only if it is NOT followed by** another pattern.
(?<=)	Positive Lookbehind	Matches something **only if it is preceded by** another pattern.
(?<!)	Negative Lookbehind	Matches something **only if it is NOT preceded by** another pattern.

Python Regex Functions	
Goal: Use the re module like a pro.	
re.match()	Tries to match a pattern **only at the beginning** of the string.
re.search()	Searches the **entire string** for the first match.
re.findall()	Returns **all matches** as a list of strings.
re.finditer()	Returns **all matches** as **match objects** (iterator).
re.sub()	Replace all matches with another string.
re.split()	Splits a string by regex pattern.