

Python Coding Language

Regular Expression - Common Terms

Term	Syntax	Meaning	Example	Matches
Core Regular Expression Terms				
Literal	abc	Exact text	cat	cat
Any character	.	Any char except newline	c.t	cat, cut
Digit	\d	Any digit 0-9	\d	7
Non-digit	\D	Any non-digit	\D	A
Whitespace	\s	Space, tab, newline	\s	
Non-whitespace	\S	Anything but whitespace	\S	A
Word char	\w	Letter, digit, underscore	\w	a, _
Non-word char	\W	Not letter/digit/_	\W	@
Anchors (Position, Not Characters)				
Start of string	^	Beginning	^Hello	Hello world
End of string	\$	End	world\$	Hello world
Word boundary	\b	Edge of a word	\bcat\b	cat
Non-boundary	\B	Inside word	\Bcat\B	educate
Character Classes				
Character set	[abc]	Any one listed	[aeiou]	a
Range	[a-z]	Any in range	[0-9]	5
Negated set	[^abc]	Anything but listed	[^0-9]	A
Quantifiers (How Many)				
Zero or more	*	0+ times	a*	"", aaa
One or more	+	1+ times	a+	a, aaa
Optional	?	0 or 1	colou?r	color, colour
Exact count	{n}	Exactly n	\d{4}	2025
Range	{n,m}	Between n and m	\d{2,4}	22, 2025

Python Coding Language

Regular Expression - Common Terms

Term	Syntax	Meaning	Example	Matches
At least	{n,}	n or more	\d{2,}	12345
Grouping & Alternation				
Group	(abc)	Captures group	(ab)+	abab
Non-capturing	(?:abc)	Group, no capture	(?:ab)+	abab
Alternation	'`'	'`'	OR	`cat
Lookarounds (Advanced but Essential)				
Lookahead	(?=x)	Followed by x	\d(?=px)	5 in 5px
Negative lookahead	(?!x)	Not followed by x	\d(?!px)	5 in 5em
Lookbehind	(?<=x)	Preceded by x	(?<=\\$)\d+	100 in \$100
Negative lookbehind	(?<!x)	Not preceded by x	(?<!\\$)\d+	100
Escaping & Flags				
Escape	\	Literal special char	\.	
Case-insensitive	i	Ignore case	/cat/i	
Global	g	All matches	\d/g	
Multiline	m	^ and \$ per line	/^cat/m	#NAME?