

- JavaScript ()
- PCRE ()
- Python ()

# Python Regex Cheatsheet

Quick Filter

<div><div>Regular Expression Basics</div><div><div>.</div><div>Any character except newline</div></div><div><div>a</div><div>The character a</div></div><div><div>ab</div><div>The string ab</div></div><div><div>a b</div><div>a or b</div></div><div><div>a*</div><div>0 or more a's</div></div><div><div>\</div><div>Escapes a special character</div></div></div>	<div><div>Regular Expression Character Classes</div><div><div>[ab-d]</div><div>One character of: a, b, c, d</div></div><div><div>[^ab-d]</div><div>One character except: a, b, c, d</div></div><div><div>[b]</div><div>Backspace character</div></div><div><div>\d</div><div>One digit</div></div><div><div>\D</div><div>One non-digit</div></div><div><div>\s</div><div>One whitespace</div></div><div><div>\S</div><div>One non-whitespace</div></div><div><div>\w</div><div>One word character</div></div><div><div>\W</div><div>One non-word character</div></div></div>	<div><div>Regular Expression Flags</div><div><div>i</div><div>Ignore case</div></div><div><div>m</div><div>^ and \$ match start and end of line</div></div><div><div>s</div><div>. matches newline as well</div></div><div><div>x</div><div>Allow spaces and comments</div></div><div><div>L</div><div>Locale character classes</div></div><div><div>u</div><div>Unicode character classes</div></div><div><div>(?iLmsux)</div><div>Set flags within regex</div></div></div>
<div><div>Regular Expression Quantifiers</div><div><div>*</div><div>0 or more</div></div><div><div>+</div><div>1 or more</div></div><div><div>?</div><div>0 or 1</div></div><div><div>{2}</div><div>Exactly 2</div></div><div><div>{2, 5}</div><div>Between 2 and 5</div></div><div><div>{2,}</div><div>2 or more</div></div><div><div>(,5)</div><div>Up to 5</div></div><div><div></div><div>Default is greedy. Append ? for reluctant.</div></div></div>	<div><div>Regular Expression Assertions</div><div><div>^</div><div>Start of string</div></div><div><div>\A</div><div>Start of string, ignores m flag</div></div><div><div>\$</div><div>End of string</div></div><div><div>\Z</div><div>End of string, ignores m flag</div></div><div><div>\b</div><div>Word boundary</div></div><div><div>\B</div><div>Non-word boundary</div></div><div><div>(?=...)</div><div>Positive lookahead</div></div><div><div>(?!...)</div><div>Negative lookahead</div></div><div><div>(?&lt;=...)</div><div>Positive lookbehind</div></div><div><div>(?&lt;!...)</div><div>Negative lookbehind</div></div><div><div>(?() )</div><div>Conditional</div></div></div>	<div><div>Regular Expression Special Characters</div><div><div>\n</div><div>Newline</div></div><div><div>\r</div><div>Carriage return</div></div><div><div>\t</div><div>Tab</div></div><div><div>\YYY</div><div>Octal character YYY</div></div><div><div>\xYY</div><div>Hexadecimal character YY</div></div></div>
<div><div>Regular Expression Groups</div><div><div>(...)</div><div>Capturing group</div></div><div><div>(?P&lt;Y&gt;...)</div><div>Capturing group named Y</div></div><div><div>(?:...)</div><div>Non-capturing group</div></div><div><div>\Y</div><div>Match the Y'th captured group</div></div><div><div>(?P=Y)</div><div>Match the named group Y</div></div><div><div>(?#...)</div><div>Comment</div></div></div>		<div><div>Regular Expression Replacement</div><div><div>\g&lt;0&gt;</div><div>Insert entire match</div></div><div><div>\g&lt;Y&gt;</div><div>Insert match Y (name or number)</div></div><div><div>\Y</div><div>Insert group numbered Y</div></div></div>

New to Debuggex? Check out the regex tester (/?flavor=python)!