Regex Cheat Sheet

Pattern	Description	Example
`^`	Start of string anchor – Asserts that the following pattern must occur at the beginning of the string.	`^Hello` matches "Hello world" only if "Hello" is at the start.
`$`	End of string anchor – Asserts that the preceding pattern must occur at the end of the string.	`world$` matches "Hello world" only if "world" is at the end.
`.`	Any character – Matches any single character except a newline.	`a.c` matches "abc", "a-c", or "a c".
`*`	Zero or more – Matches zero or more occurrences of the preceding element.	`ab*c` matches "ac", "abc", "abbc", etc.
`+`	One or more – Matches one or more occurrences of the preceding element.	`ab+c` matches "abc", "abbc", but not "ac".
`?`	Zero or one – Matches zero or one occurrence of the preceding element. Also makes quantifiers non-greedy when placed after them.	`colou?r` matches both "color" and "colour".
`{n}`	Exact count – Matches exactly *n* occurrences of the preceding element.	`a{3}` matches "aaa".
`{n,}`	At least n occurrences – Matches *n* or more occurrences of the preceding element.	`a{2,}` matches "aa", "aaa", "aaaa", etc.
`{n,m}`	Range of occurrences – Matches between *n* and *m* occurrences (inclusive) of the preceding element.	`a{2,4}` matches "aa", "aaa", or "aaaa".
`[abc]`	Character class – Matches any one of the characters enclosed in the brackets.	`[abc]` matches "a", "b", or "c".
`[^abc]`	Negated character class – Matches any character *not* listed between the brackets.	`[^abc]` matches any character except "a", "b", or "c".

Pattern	Description	Example
`(abc)`	Grouping and capturing – Groups the characters "abc" together, which can then be referenced later.	`(abc)+` matches "abc", "abcabc", etc.
`|`	Alternation (OR) – Matches the expression before or after the `|` symbol.	`cat|dog` matches "cat" or "dog".
`\d`	Digit character – Matches any single digit (equivalent to `[0-9]`).	`\d` matches "0", "1", …, "9".
`\D`	Non-digit character – Matches any character that is not a digit.	`\D` matches letters or symbols that are not digits.
`\w`	Word character – Matches any alphanumeric character or underscore (equivalent to `[A-Za-z0-9_]`).	`\w` matches letters, digits, and underscores.
`\W`	Non-word character – Matches any character that is not a word character.	`\W` matches punctuation, spaces, etc.
`\s`	Whitespace character – Matches spaces, tabs, newlines, and other whitespace.	`\s` matches a single space or tab.
`\S`	Non-whitespace character – Matches any character that is not whitespace.	`\S` matches letters, digits, punctuation, etc.
`\b`	Word boundary – Asserts a position between a word character and a non-word character (or the beginning/end of the string).	`\bword\b` matches "word" as a whole word (not part of "sword" or "words").
`\B`	Non-word boundary – Asserts a position that is not a word boundary.	`\Bend` might match "bend" within a longer word like "amendment".

How to Use This Cheat Sheet

- **Copy & Paste:** Copy the above HTML into your editor.
- **Save as HTML:** Save the file as `regex_cheat_sheet.html`.
- **Open in Browser:** Open the HTML file in any web browser to view the cheat sheet.

- **Print or Export:** You can also print or export the page to PDF from your browser.