# Regex reference

**Unix shell file selection wildcards:**

- `*` matches zero or more characters
- `?` matches exactly one character
- `[ ]` matches a character from a list or range of contained options
- `[! ]` matches a character NOT in a list or range of contained options
- `{ }` expands to produce forms of all listed contained options

---

**Extended Regular Expression Syntax:**

- `|` acts as an OR between options
- `( )` allows grouping, e.g. for OR modifier, with quantifiers, etc..
- `[ ]` matches a character from a list or range of contained options
- `[^ ]` matches a character NOT in a list or range of contained options
- `^` at the start of a regex means match at start of line
- `$` at the end of a regex means match at end of line
- `.` is the match-all (any single character) wildcard
- `?` quantifies previous character or group as occurring zero or one time
- `*` quantifies previous character or group as occurring zero or more times
- `+` quantifies previous character or group as occurring one or more times
- `{n,m}` quantifies previous character or group as occurring between n and m times
- *Quantifiers are greedy- will always match longest possible fit.*
- `[:alpha:] [:alnum:] [:digit:] [:upper:] [:lower:] [:punct:] [:space:]`
- `\w` : Word character [a-zA-Z0-9] OR a _ (underscore)
- `\W` : `[^\w]` Inverse of \w, any non-word character
- `\s` : Spaces, tabs, in some contexts new-lines
- `\S` : `[^\s]` Inverse of \s, any non-space character
- `\b` : Boundary between adjacent word and space, 0-length anchor
- `\B` : `[^\b]` In the middle of a word or multiple spaces, 0-length anchor
- `\<` : Boundary at *start* of word between word and space, 0-length anchor
- `\>` : Boundary at *end* of word between word and space, 0-length anchor
- `\1`, `\2`, etc.. : Back-reference- refer back to an exact copy of a matched (group)