Regular Expressions

Regular expression queries provide a powerful way of searching texts, allowing for higher-level descriptions than just literal words or characters. For example, to search for either “car” or “cars”, use the pattern `cars?` . The `?` allows the previous element (“`s`”) to occur either 0 or 1 time.

Pattern	**Description**	**Example query**
`.`	Match **any** character	`b.t`
`?`	Match the previous element **zero or one** time	`car.?`
`*`	Match the previous element **zero or more** times	`car.*`
`+`	Match the previous element **one or more** times	`car.+`
`{3,7}`	Match the previous element **between** 3 and 7 times	`.{10,12}`
`{3,}`	Match the previous element **at least** 3 times	`.{14,}`
`{,7}`	Match the previous element **no more than** 7 times (including 0)	`.{,2}`
`|`	Match the previous element **or** the subsequent element	`ab|cd`
`(...)`	**Group** the characters within the parentheses	`talk(ed|ing)`
`[ptk]`	Match **either** `p`, `t`, or `k`	`[mb]et`
`[^ptk]`	Match anything **except** `p`, `t`, or `k`	`[^mb]et`
`^a`	Match `a` only at the **beginning** of an entire query	`^met`
`a$`	Match `a` only at the **end** of an entire query	`met$`
`\w`	Match any **alphabetic or numeric** character (word-like)	`\w+`
`\W`	Match anything **except alphabetic or numeric** characters	`\W+`
`\d`	Match any **number** (digit)	`\w+\.\d+`
`\D`	Match anything **except numbers**	
`\s`	Match any **space** character (eg. space, tab)	
`\S`	Match anything **except space** characters	
`\b`	**Word boundary**	`\bhello\b`

1 Examples

Find either Meghan or Megan: `Megh?an`

Find either Katelyn or Catelyn: `[CK]atelyn`

Find either Ben or Benjamin: `Ben(jamin)?`

Find either Emily or Emilie: `Emil(ie|y)`

Find a credit card number: `\d{16}`

Find a recent year: `(19|20)\d{2}`

Find either Detmar or Dettmar or Dietmar or Dietmarr: