KQL CHEAT SHEET

`chrome firefox safari`	**free text**: Matches “chrome” or “firefox” or “safari” in all fields (default)
`"win xp"`	**phrase**: Matches the exact phrase “win xp” as it appears (all fields)
`os : "win xp"`	**phrase (field-specific)**: Matches the exact phrase “win xp” as it appears in the field os
`browser : (chrome or firefox or safari)`	**or**: Matches at least one of “chrome”, “firefox” or “safari” in the field browser
`tags: ("security" and "error")`	**and**: Field tags contain both security and error as list elements
`tags: security and (warning or error)`	**grouping (values)**: Field tags matches “security” and also either “warning” or “error”
`agent :*`	**exists**: Field agent exists in any form
`agent : "MSIE 6.0"`	**phrase**: Field agent matches exact phrase “MSIE 6.0”
`machine.os* : win*`	**wildcard**: Fields beginning with machine.os matches values beginning with win
`response : 503 and agent : "MSIE 6.0"`	**and (multiple fields)**: Field response equals 503 and field agent contains the phrase “MSIE 6.0”
`response : 404 and not (browser : chrome)`	**not**: Field response equals 404 but field browser is not chrome
`response : 503 and extension : "gz" or extension: "deb"`	**precedence**: Either field response equals 503 and field extension is “gz”, or field extension is “deb” (and precedes or)
`response : 503 and (extension : "gz" or extension: "deb")`	**grouping**: Field response equals 503 and extension is either “gz” or “deb”
`response : 503 and not (extension : "gz" or extension: "deb")`	**grouping with negation**: Field response equals 503 but field extension is neither “gz” or “deb”
`@timestamp > "2020-03-02"`	**date (after)**: All events after 2 Mar 2020
`@timestamp >= "2020-03-02" and @timestamp <= "2020-03-05"`	**date (between)**: All events between 2 Mar 2020 and 5 Mar 2020 (inclusive)
`bytes > 1000`	**greater**: Field bytes has value greater than 1000
`ip: 192.168.0.0/16`	**IPV4 CIDR:** Field ip is between 192.168.0.1 and 192.168.0.254
`response : 200 agent: chrome`	✗ throws error as and / or is required between criteria in KQL. However will work in legacy Lucene syntax
`bytes :>`	✗ throws error as : is not needed in KQL
`bytes:[1000 TO 5000]`	✗ throws error, expand as bytes >= 1000 and bytes <= 5000 in KQL