Cheatography

JQ Cheat Sheet

by Orabig via cheatography.com/55627/cs/14790/

Basic filters

.	IDentity
.foo *.foo*?	Value of "*foo*" key
.[] .[]?	Array iterator. Produce each element of an input array, or each value of an object
.[*n*]	n^{th} element of an array (*n* can be negative : -1 -> last element...)
.[*n*:*m*]	Array slice : array containing n^{th} (inclusive) to m^{th} (exclusive) elements
A,*B*	Produces output of filter *A* then *B* (both *A* and *B* are fed with the same input)
A|*B*	Output of *A* is sent to *B*'s input
(A)	Grouping operator

Types and Values

[ ], { }	Array (resp. Object) construction
..	Recursive descent
+ - * / %	Basic arithmetic / string / array / object operators
length	string / array / object length
keys | keys_unsorted	The sorted/unsorted set of the input object keys.
has(*KEY*)	Whether the input object as the given *KEY*.
in(*A*)	Whether the input key is in the given *A* object.
map (*A*)	Run the *A* filter for each element of the input array. Equivalent to **[.[] | *A*]**

Types and Values (cont)

map_values(*A*)	Run the *A* filter for each element of the input object. Equivalent to **.[] |=*A***
del (*x*)	Removes a key and its value from an object
select (*foo*)	Produces input unchanged if *foo* is true for that input.
type	Returns the type of its argument as a string.
arrays, objects, iterables, booleans, numbers, normals, finites, strings, nulls, values, scalars	These built-ins select only inputs that are arrays, objects, iterables (arrays or objects), booleans, numbers, normal numbers, finite numbers, strings, null, non-null values, and non-iterables, respectively.
empty	Produces no output.
$__loc__	Produces an object with a "file" key and a "line" key
add	Produces the summed elements of the input array
any, any(*foo*)	Produces **true** if any of the elements of the input array (resp *foo*) is **true**
all, all(*foo*)	Produces **true** if all of the elements of the input array (resp *foo*) is **true**
range([*from* ;] upto [; *by*])	Produces a range of numbers (*upto* is exclusive)

Types and Values (cont)

floor, sqrt	Returns the floor (resp square root) of its numeric input
tonumber	Converts into to number
infinite, nan, isinfinite, isnan, isfinite, isnormal	Returns **true** depending of the input
sort | sort_by(*foo*)	Sorts the input array (*null* < *false* < *true* < numbers < strings < arrays < objects)
group_by(*foo*)	Groups the elements of the input array having the same *foo* value into separate arrays (sorted by *foo* values)
min | max | min_by(*foo*) | max_by(*foo*)	Finds the minimum (resp maximum) element of the input array
unique, unique_by(*foo*)	Produces an array of unique element of the input array.
reverse	Reverses an array
contains(*foo*)	Produces **true** if *foo* is completely contained within the input.
indices(*foo*)	Outputs an array containing the indices in . where *foo* occurs.
inside(*foo*)	produce true if the input is completely contained within *foo*
combinations	Production all combinations of an array

By **Orabig**
cheatography.com/orabig/

Published 5th March, 2018.
Last updated 5th March, 2018.

Cheatography

JQ Cheat Sheet

by Orabig via cheatography.com/55627/cs/14790/

SQL-Style Operators

INDEX	
JOIN	
IN	

String manipulation

tostring	JSON-encode input as a string
"\(*foo*)"	Interpolates *foo* inside a string
index(*foo*), rindex(*foo*)	Outputs the index of the first (index) or last (rindex) occurrence of *foo* in the input.
startswith(*str*)	Outputs true if . starts with the given string argument.
endswith(*str*)	Outputs true if . ends with the given string argument.
ltrimstr(*foo*), rtrimstr(*foo*)	Outputs its input with the given prefix (resp. suffix) string removed, if it starts (resp. ends) with it.
explode	Converts an input string into an array of the string's codepoint numbers.
implode	The inverse of explode.
split(*foo*)	Splits an input string on the separator argument.
join(*foo*)	Joins the array of elements given as input, using the argument as separator.
ascii_downcase, ascii_upcase	Emit a copy of the input string with its alphabetic characters (a-z and A-Z) converted to the specified case.

Path & object manipulation

path (*x*)	Output the array representation of *x* : (keys/ indices, values)
getpath(*PATHS*)	Outputs the values in . found at each path in *PATHS*
setpath(*PATH*; *VALUE*)	Set the *PATHS* in . to *VALUE*
delpaths(*PATHS*)	Removes the key at the paths in *PATHS*
to_entries	Converts from object to an array of "key":"value"
from_entries	Converts from an array of "key":"value" to an object
with_entries(*foo*)	Shortcut for **to_entries | map(*foo*) | from_entries**
flatten, flatten(*depth*)	Produces a flat array in which all arrays inside the original array have been recursively replaced by their values.

Loop control

while(cond; update)	repeatedly apply an update to . until cond is false.
until(cond; next)	repeatedly apply the expression next, initially to . then to its own output, until cond is true.
recurse(*foo* [,*cond*])	search through a recursive structure, and extract data from all levels.
walk(*foo*)	applies *foo* recursively to every component of the input entity.
bsearch(*foo*)	conducts a binary search for *foo* in the input array.

Regular expressions

test(*RE* [; *FLAGS*])	True if input string matches the given RE
match(*RE* [; *FLAGS*])	outputs an object for each match it finds.
capture(*RE* [; *FLAGS*])	Collects the named captures in a JSON object, with the name of each capture as the key, and the matched string as the corresponding value.
scan(*RE* [; *FLAGS*])	Emit a stream of the non-overlapping substrings of the input that match the regex in accordance with the flags, if any have been specified.
split|splits(*RE* [; *FLAGS*]), splits()	Splits an input string, and provides an array (resp. stream)
sub|gsub(*RE* ; *tostring* [; *FLAGS*])	Emit the string obtained by replacing the first (resp. all) match of regex in the input string with tostring, after interpolation.

FLAGS is any of "g, i, m, s, p, n, l, x"

By **Orabig**
cheatography.com/orabig/

Published 5th March, 2018.
Last updated 5th March, 2018.