KeRF Reference Card

Lists

Name		Behavior
ascend(x)	<	Ascending indices / grade up.
bucketed(x,y)		Bucket y Values by x.
car(x)		First element of x.
cdr(x)		All except first element of x.
count(x)		Number of elements in x.
deal(x,y)		x unique random items of y.
descend(x)	>	Descending indices / grade dn.
distinct(x)	%	First instance of elements of x.
drop(x,y)	_	Remove first x elements from y.
enlist(x)		Wrap x in a list.
enumerate(x)	^	Range [0,x) / map keys.
except(x,y)		Remove all y from x.
explode(x,y)		Split y at instances of x.
first(x,y)	^	Take first x elements of y.
flatten(x)		Concatenate elements of x.
implode(x,y)		Join elements of y with x.
in(x,y)		Is the item x in y?
intersect(x,y)		Collect items in x and y.
join(x,y)	#	Create a list from x and y.
last(x,y)		Last x elements of y.
order(x)		Equivalent to <<x.
part(x)	&	Map unique items to indices.
range(x)		Integer vector [0, x).
range(x,y)		Numeric vector [x, y), step 1.
range(x,y,z)		Numeric vector [x, y), step z.
repeat(x,y)		List of x copies of y.
reverse(x)	/	Reverse the order of items in x.
search(x,y)		Find index of x in y or null.
shift(x,y,z)		Shift y by x, fill with z.
shuffle(x)		Randomly permute x.
sort(x)		Sort the items in x.
split(x,y)		Split x by indices in y.
trim(x)		Remove leading/trailing spaces.
union(x)		Set union of x and y.
which(x)	?	Gather nonzero indices.

Math

Name		Behavior
between(x,y)		Is x between y[0] and y[1]?
dotp(x,y)		Dot/Scalar product of x and y.
log(x)		$\log_{10}(x)$.
log(x,y)		$\log_{x}(y)$.
lsq(A,B)	\	Solve $Ax = B$ for x.
minv(A)		Inverse of matrix A.
mmul(A,B)		Multiply matrices A and B.
rand()		1 random float [0, 1).
rand(x)		1 random integer [0, x).
rand(x,y)	?	x random integers [0, y).
stamp_diff(x,y)		Stamp difference in ns.
transpose(x)	.	Matrix transpose of x.

```
+ - / * % ** !  & | ˜ :  = == < > <= >= != <> abs acos add
asin atan ceil cos cosh divide erf erfc exp floor lg ln
mod negative pow sin sinh sqrt subtract tan tanh times and
equal greater greatereq less lesseq match not noteq or
```

Aggregates

Name	Behavior
avg(x)	Arithmetic mean of elements.
count_nonnull(x)	How many items in x not null?
count_null(x)	How many items in x are null?
mavg(x,y)	x-wide moving average of y.
max(x)	Maximum element of x.
mcount(x,y)	x-wide moving count of y.
median(x)	Median of x.
min(x)	Minimum element of x.
mmax(x,y)	x-wide moving maximum of y.
mmin(x,y)	x-wide moving minimum of y.
msum(x,y)	x-wide moving sum of y.
rsum(x)	Running sum of items in x.
std(x)	Standard Deviation of elements.
sum(x)	Sum of elements.
var(x)	Variance of elements.
xbar(x,y)	Equivalent to `x * floor y/x`.

Miscellaneous

Name	Behavior
atom(x)	Is x an atom?
char(x)	Cast x to a char.
eval(x)	Evaluate the string x.
float(x)	Cast x to a float.
hashed(x)	Create an enumeration of x.
ifnull(x)	Is x null?
indexed(x)	Create an index of x.
int(x)	Cast x to an integer.
json_from_kerf(x)	Convert to JSON.
kerf_from_json(x)	Parse JSON.
kerf_type(x)	Numeric typecode of x.
kerf_type_name(x)	Human-readable type name.
now()	Current date-time.
now_date()	Current date.
now_time()	Current time.
rep(x)	String representation of x.
shell(x)	Execute x as a shell command.
seed_prng(x)	Seed the RNG used by **rand**.
sleep(x)	Wait at least x milliseconds.
string(x)	Cast x to string.
tables()	List names of loaded tables.
timing(x)	Enable or disable time logging.
type_null(x)	Type-specific null of x.

Tables And Maps

Name	Behavior
asof_join(x,y,k1,k2)	Left join x and y on k1 as of k2.
atlas(x)	Create an Atlas from a map.
has_column(x,y)	Does table x have column y?
has_key(x,y)	Is y a valid index to x?
left_join(x,y,k)	Left join x and y on k.
map(x,y)	Map from keys x and values y.
xkeys(x)	Keys of x.
xvals(x)	Values of x.

IPC	
Name	Behavior
`open_socket (host, port)`	Host and port must be strings.
`close_socket(handle)`	Close an IPC socket handle.
`send_async(handle, expr)`	Doesn't block, Returns 1.
`send_sync(handle, expr)`	Returns `eval(y)` on remote host.
Launch with `-p <portnumber>` to start IPC server	

Combinators	
Name	Behavior
`u converge x`	Apply u to x until the value repeats or stops changing.
`x u converge y`	Apply u to x, y times.

`u deconverge x`
`x u deconverge x`
	As `converge`, but gather intermediate results.

`b fold x`
`x b fold y`
	Apply b between elements of x.

`b mapback x`
`x b mapback y`
	Apply b between elements of x and their predecessor.

`u mapdown x`
`x b mapdown y`
	Apply u/b to each element of x (and each of y if present).

`x b mapleft y`	Apply b to y and each of x.
`x b mapright y`	Apply b to x and each of y.

`b unfold x`
`x b unfold y`
	As `fold`, but gather intermediate results.

SQL Syntax
`INSERT INTO table VALUES data`
`DELETE FROM table [ WHERE condition ]`
`SELECT field1, field2 [ AS name ] ... FROM table [ WHERE condition ] [ GROUP BY aggregate ]`
`UPDATE table SET field1:val1, field2:val2 ... [ WHERE condition ] [ GROUP BY aggregate ]`

KeRF types		
Example	`kerf_type_name`	`kerf_type`
`[2000.01.01]`	`stamp vector`	-4
`[0.1]`	`float vector`	-3
`[1]`	`integer vector`	-2
`"A"`	`character vector`	-1
`{[x] 1+x}`	`function`	0
`` `A ``	`character`	1
`1`	`integer`	2
`0.1`	`float`	3
`2000.01.01`	`stamp`	4
`()`	`null`	5
`[]`	`list`	6
`{a:1}`	`map`	7
`#["a"]`	`enum`	8
`=[1]`	`sort`	9
`{{a:1}}`	`table`	10

IO and Scripting	
Name	Behavior
`load(filename)`	Read and execute KeRF source.
`exit(statusCode)`	Exit, status code is optional.
`out(string)`	Print a raw string x to stdout.
`display(x)`	Prettyprint x to stdout.
`dir_ls(path, showFullPaths)`	List the files in a directory path.
`lines(filename, limit)`	Read lines of a text file.
`open_table(filename)`	`mmap` serialized KeRF table.
`read_from_path(filename)`	Load a serialized KeRF object.
`read_table_from_csv (filename,fields,limit)`	Load a CSV file into a table.
`read_table_from_delimited_file (delim,filename,fields,limit)`	Load a delimited file into a table.
`read_table_from_fixed_file (filename,attr)`	Load a fixed-width file. `attr` contains field types, widths, etc.
`read_table_from_tsv (filename,fields,limit)`	Load a TSV file into a table, optionally limited to n rows.
`write_csv_from_table (filename,table)`	Write a table to a CSV file.
`write_delimited_file_from_table (delim,filename,table)`	Write a table to a delimited file.
`write_text(filename, x)`	Serialize x to a file as JSON.
`write_to_path(filename, x)`	Serialize x to a file as binary.
`dlload(filename,funcname,argc)`	Load a dynamic library.

`fixed_file` Attributes	
Name	Behavior
`fields`	Field specifier string.
`widths`	List of column widths in chars.
`line_limit`	Max row count.
`titles`	List of column names.
`header_rows`	How many rows are headers?
`line_separated`	Are rows separated by newlines?

Delimited/Fixed-Width Field Specifiers	
Symbol	Datatype
`I`	Integer
`F`	Float
`S`	String
`E`	Enumerated String
`G`	IETF RFC-4122 UUID
`N`	IP address as parsed by `inet_pton()`
`Z/z`	Custom. See `.Parse.strptime_format`
`*`	Skipped field
`Y`	NYSE TAQ symbol (Fixed-width only)
`Q`	NYSE TAQ time format (Fixed-width only)