# fx JSON viewer and processor

## CONCEPTS

Every argument is a JavaScript function; the result of one is the input x of the next. Start with . to skip x ⇒ x.

**Input** file or stdin
↓ parse
**Arg 1** .users
↓ x
**Arg 2** @.name
↓ x
**Output** JSON or text

Each argument gets the previous result as x

## RUN AND INPUT

**fx** *<file>*
Open a file interactively

*<cmd>* **| fx**
Open piped JSON interactively

**fx** *<file>* *<expr>*
Apply an expression, print result

*<cmd>* **| fx .**
Pretty-print stdin

**fx** *<file>* *<expr1>* *<expr2>*
Chain expressions

## EXPRESSIONS

**.***<key>*
Select a field

**.***<key>***[0]**
Select an array element

**x ⇒ x.***<key>*
Explicit arrow function

**'.map(x ⇒ x.***<key>***)'**
Map over an array in JS

**'.filter(x ⇒ x.***<key>* **> 1)'**
Filter an array in JS

**@.***<key>*
Shorthand for map

**?.***<key>*
Shorthand for filter

## BUILT-IN FUNCTIONS

**len**
Length of array, string or object

## BUILT-IN FUNCTIONS (cont.)

**keys / values**
Object keys or values

**list**
Print array as lines for xargs

**save**
Write edits back to the input file

## FORMATS

**--yaml**
Read YAML

## INTERACTIVE: MOVE

Arrows
Move; h j k l also work

g / G
Top / bottom

@
Fuzzy-search a path and jump

q
Quit

## INTERACTIVE: FOLD AND VIEW

Enter / l
Expand node

Backspace / h
Collapse node

e / E
Expand / collapse all

?
Show help

## INTERACTIVE: SEARCH AND OUTPUT

.
Open query input

/
Regex search

n / N
Next / previous match

P
Print value to stdout and exit