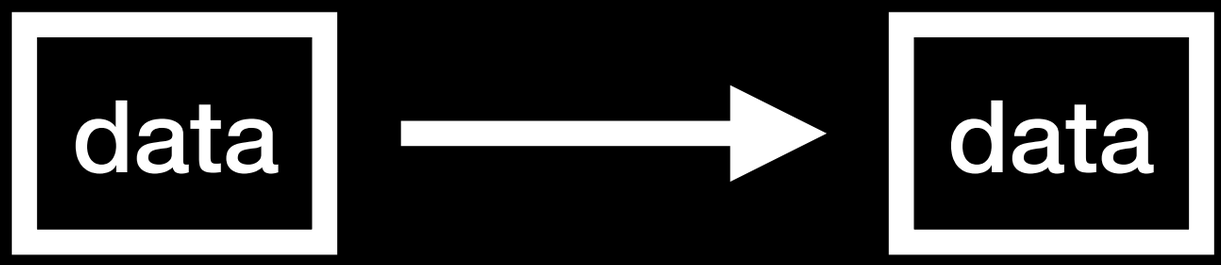
PANE 

Programming with visible data

LIVE 2018

Joshua Horowitz

data processing



Pentametron Retweeted

kyia is here for ariana♡ @oltvisuals · Oct 10

I hope Selena Gomez is okay

1 13

Pentametron Retweeted

Alessandra @_babykinxx · Oct 10

It was a slightly better day today

1 11

Pentametron Retweeted

Scott Pilgrim @scottvs_thebot · Oct 9

Scott Pilgrim vs. His Eternal Shame

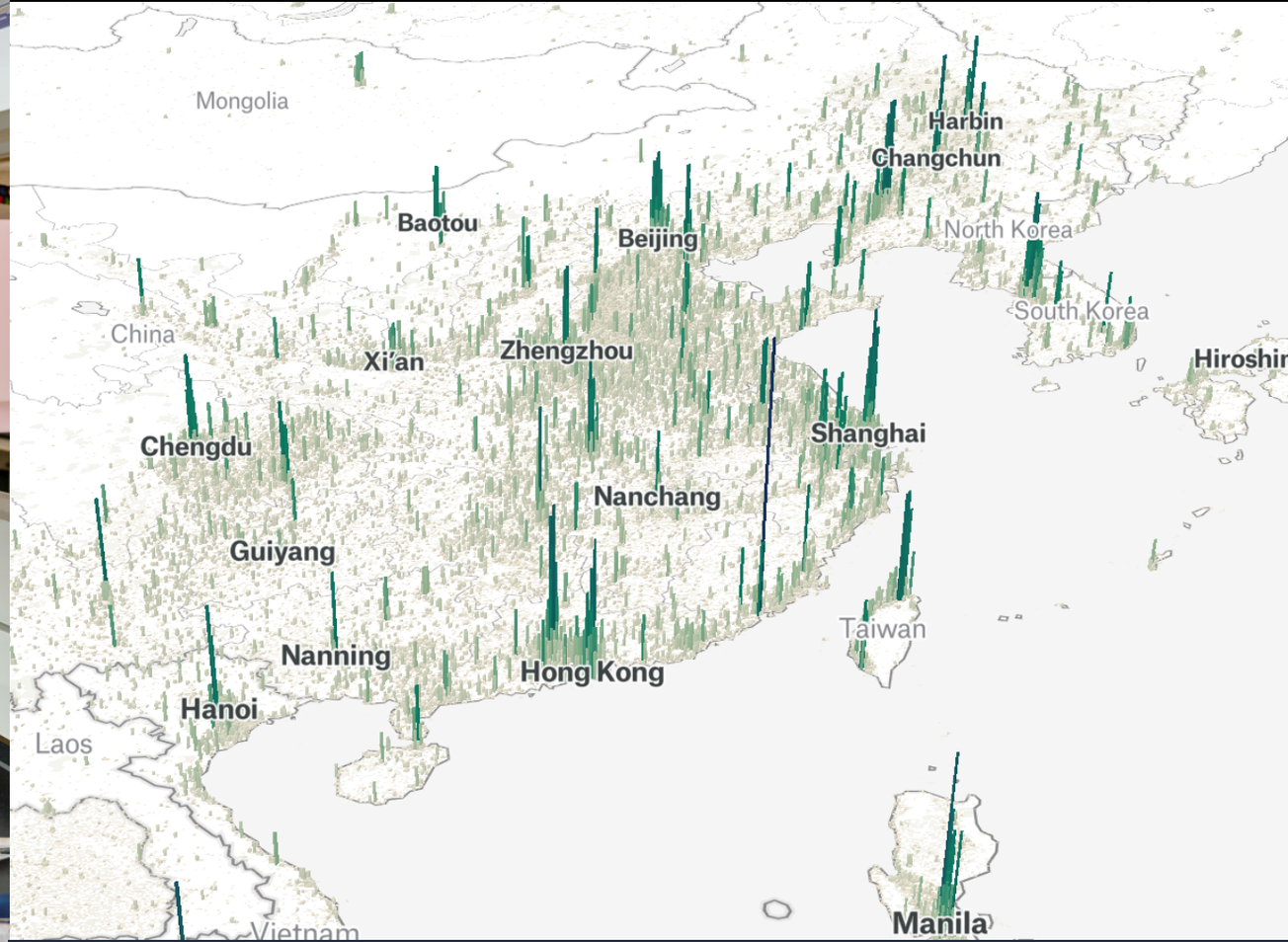
7 26

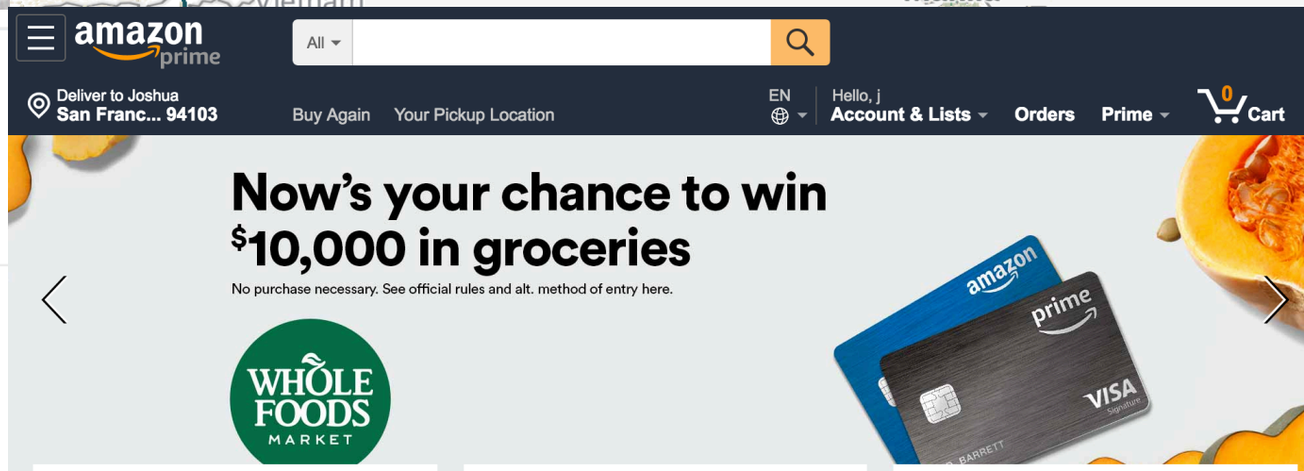
Pentametron Retweeted

Drew Ricci @Drewajr1 · Oct 8

This is a very satisfying game

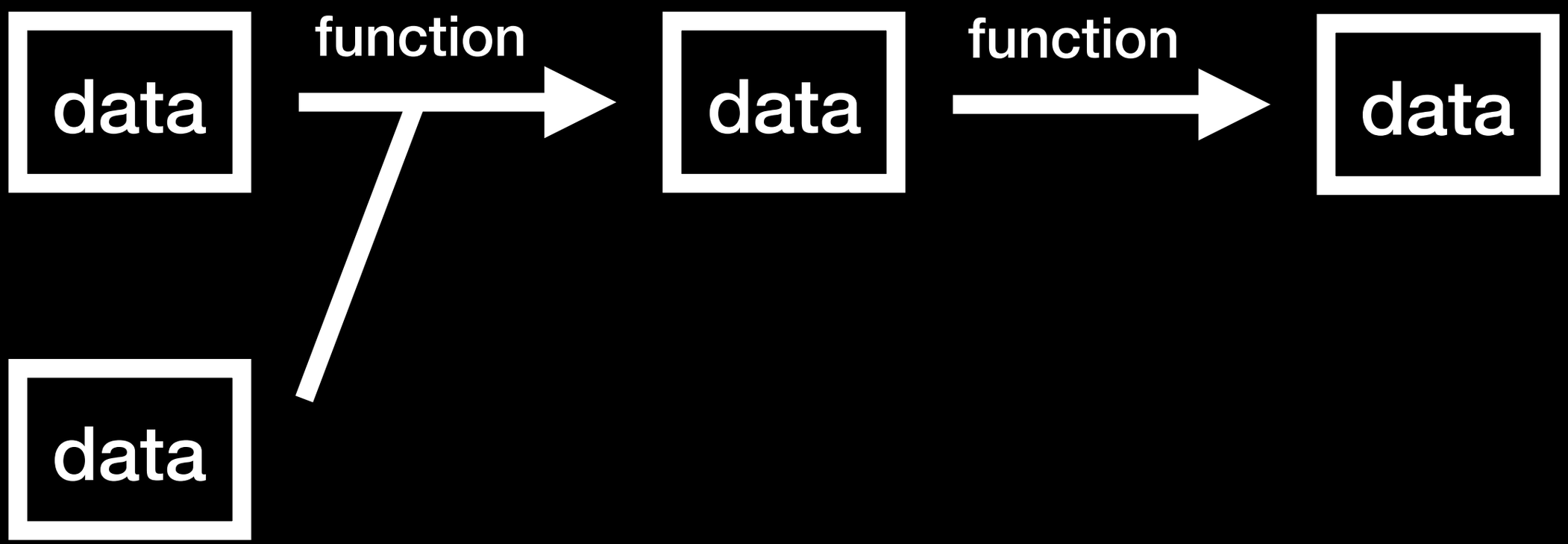
1 5

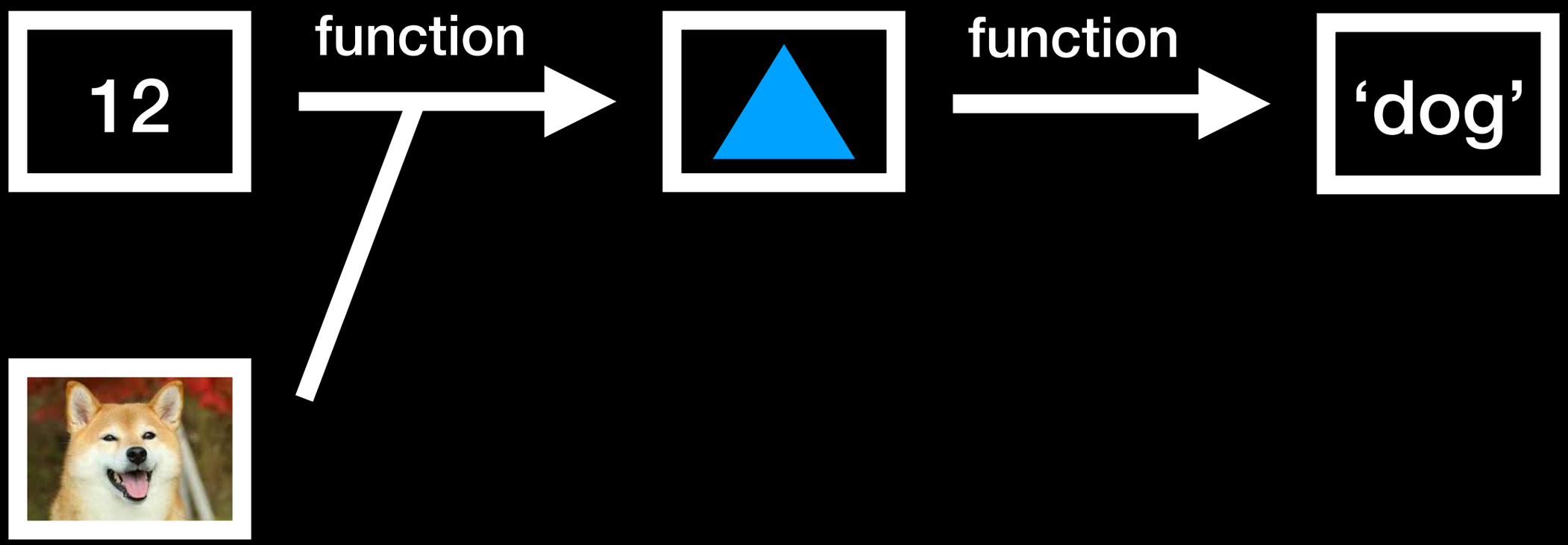




```haskell
hilbert 0 = mempty
hilbert n = hilbert' (n-1) # reflectY <> vrule 1
         <> hilbert  (n-1) <> hrule 1
         <> hilbert  (n-1) <> vrule (-1)
         <> hilbert' (n-1) # reflectX
  where
    hilbert' m = hilbert m # rotateBy (1/4)

diagram :: Diagram B
diagram = strokeT (hilbert 6) # lc silver
                              # opacity 0.3
```


data
function
data
function
data
data


12
function
function
‘dog’

```haskell
hilbert 0 = mempty
hilbert n = hilbert' (n-1) # reflectY <> vrule 1
         <> hilbert  (n-1) <> hrule 1
         <> hilbert  (n-1) <> vrule (-1)
         <> hilbert' (n-1) # reflectX
  where
    hilbert' m = hilbert m # rotateBy (1/4)

diagram :: Diagram B
diagram = strokeT (hilbert 6) # lc silver
                              # opacity 0.3
```

The Principle of Radical Visibility

All data should be visible by default.

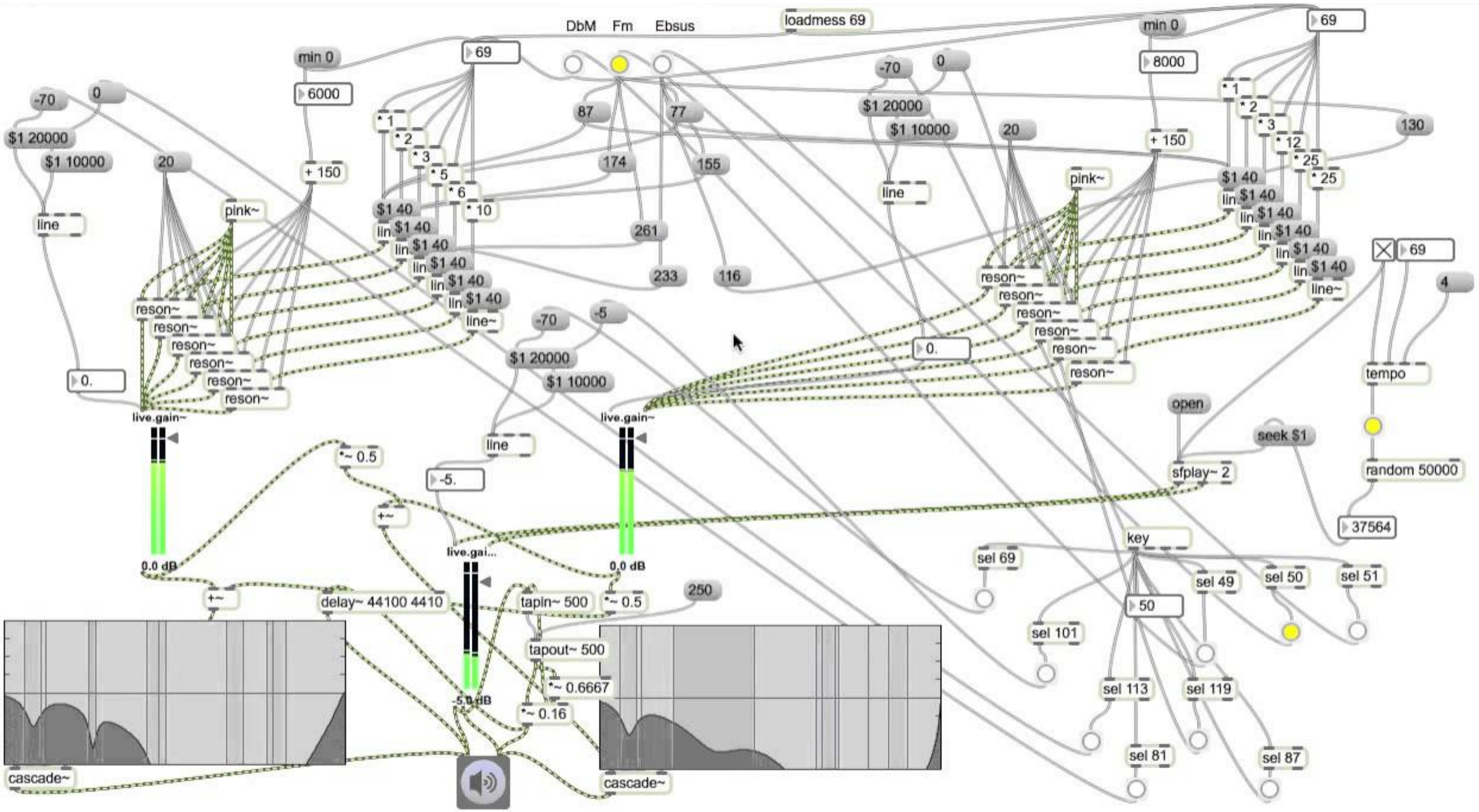
PANE 

- a prototype live programming interface
- for general data processing
- built on a functional paradigm
- in a pragmatic niche between text and VPL
- following the Principle of Radical Visibility

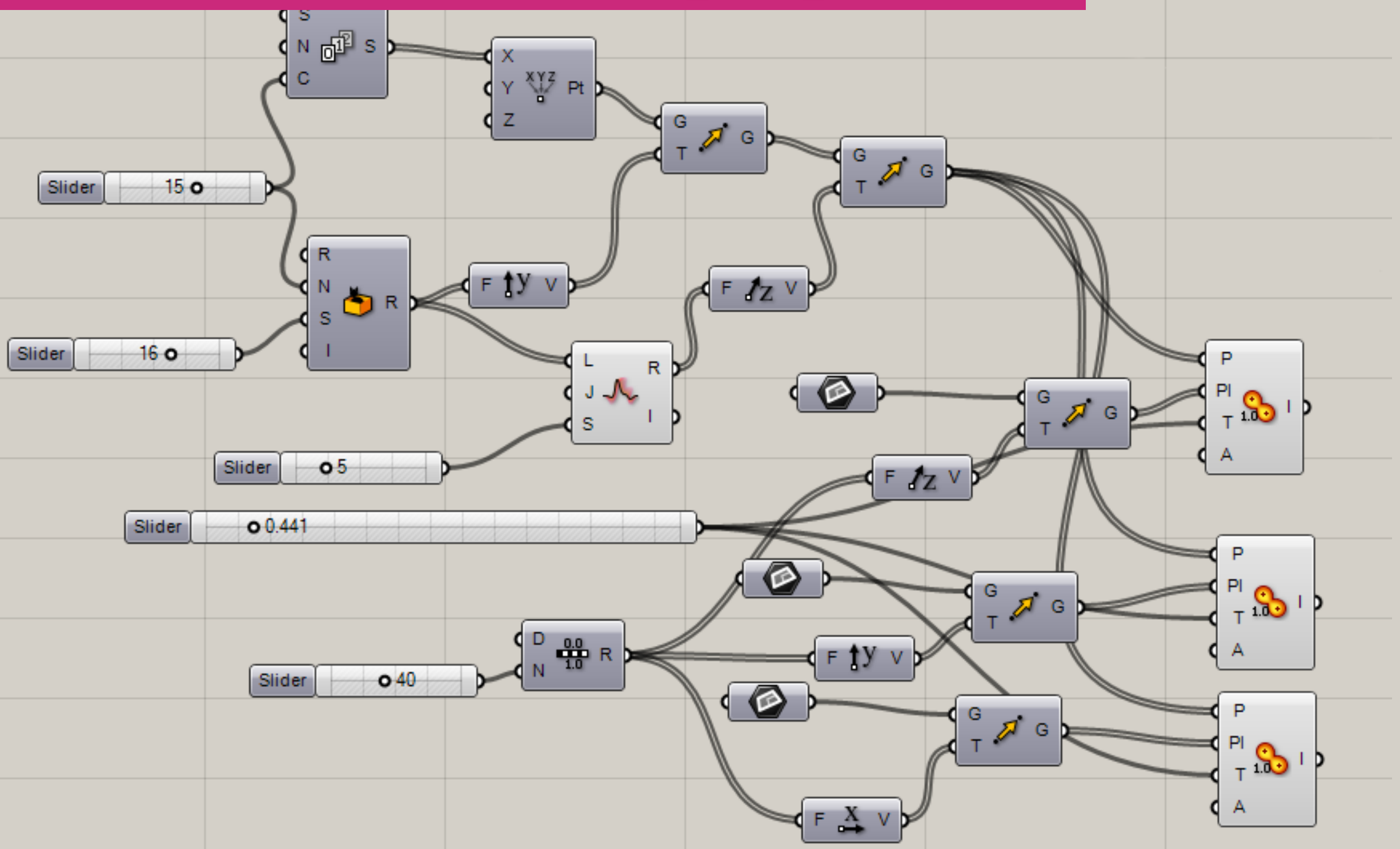
Demo

Related work & reflections

Max/MSP


DbM
Fm
Ebsus
loadmess 69
min 0
69
6000
+ 150
-70
0
$1 20000
$1 10000
20
line
pink~
reson~
live.gain~
0.0 dB
*~ 0.5
+~
delay~ 44100 4410
tapin~ 500
tapout~ 500
*~ 0.6667
*~ 0.16
250
cascade~
line~
$1 40
lin
87
77
174
155
261
233
116
-5
-5.
0.
min 0
8000
+ 150
130
* 1
* 2
* 3
* 5
* 6
* 10
* 12
* 25
4
tempo
random 50000
open
seek $1
sfplay~ 2
37564
key
sel 69
sel 49
sel 50
sel 51
50
sel 101
sel 113
sel 119
sel 81
sel 87

Grasshopper



LabVIEW

Search Customize

No Error

Close Device.vi

SH BB

CLOSE

error out

No Error

Memory Allocation

Data Acquisition

I32

0

0

Fetch Trace 32f.vi

SH BB

32f

TRACE

DBL min output

DBL max output

Waveform Graph

DBL

DBL

STOP

TF

Peak Amplitude (dBm)

SGL

Peak Frequency (Hz)

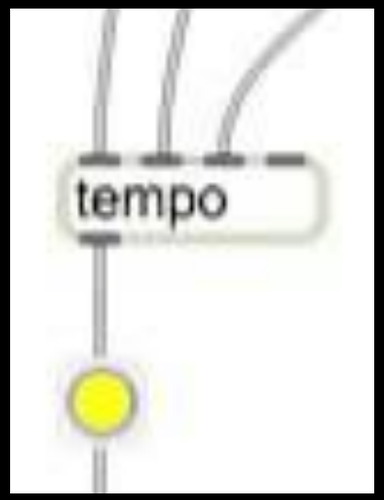
SGL

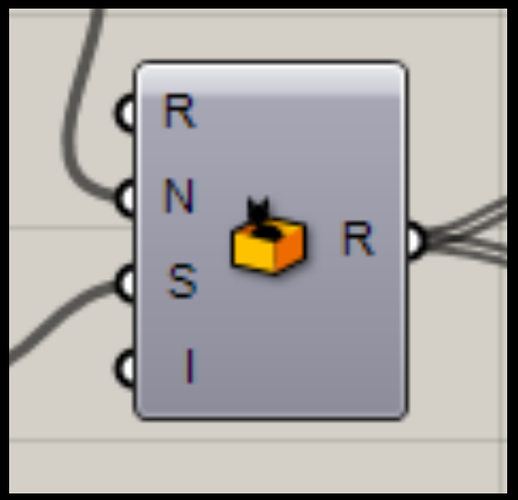
status

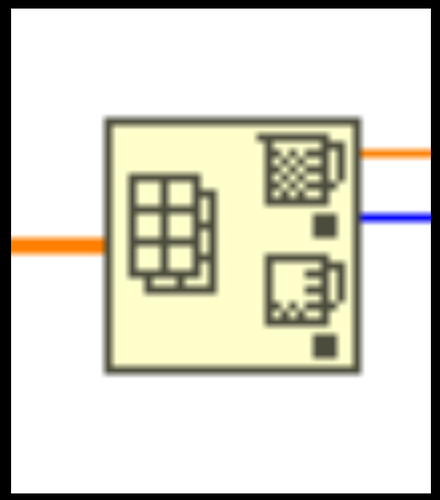
2

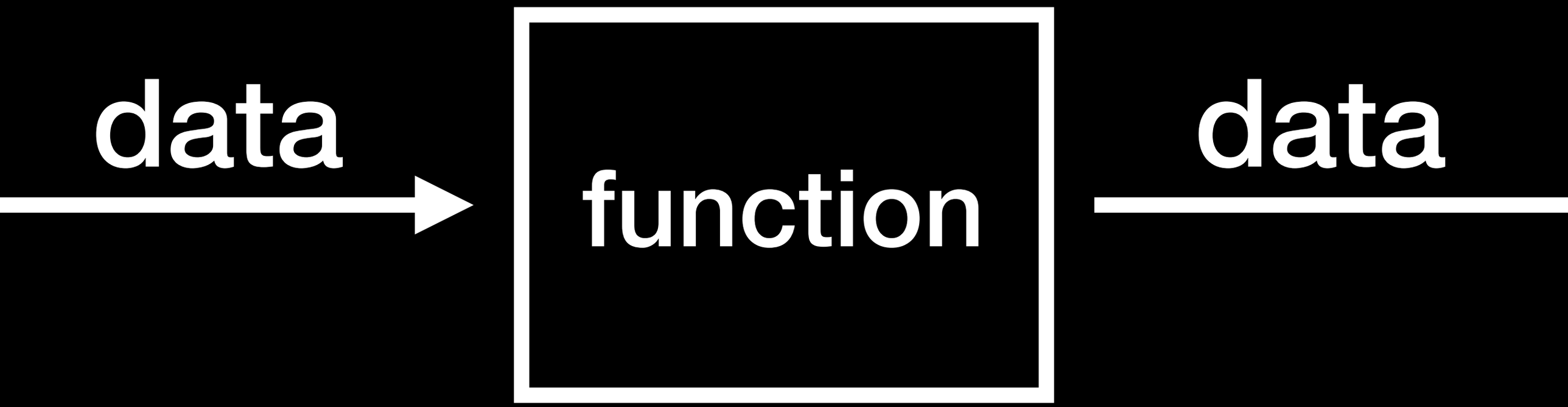
important differences

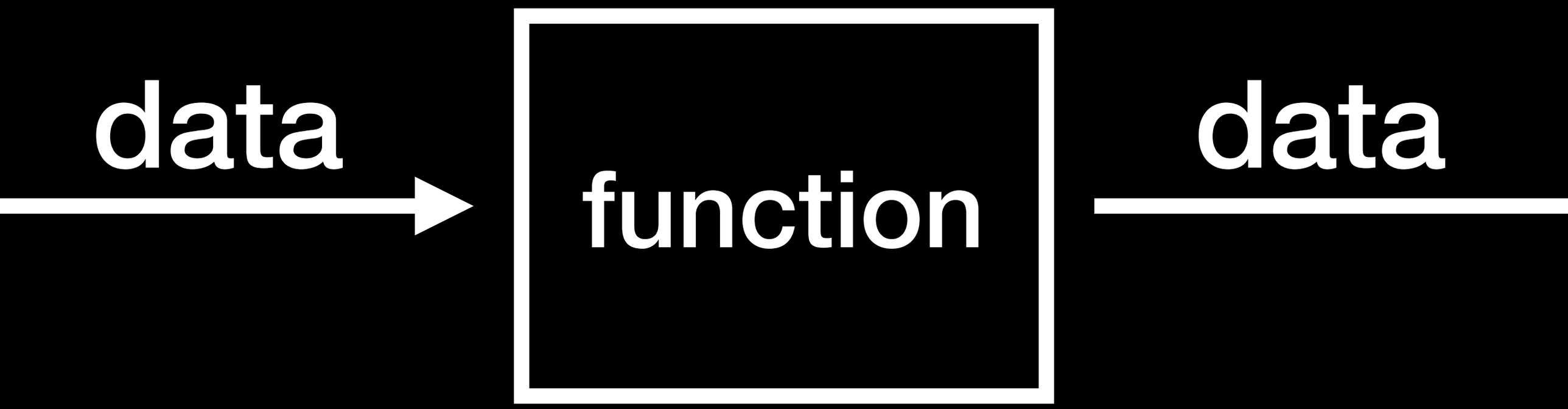
1.


tempo

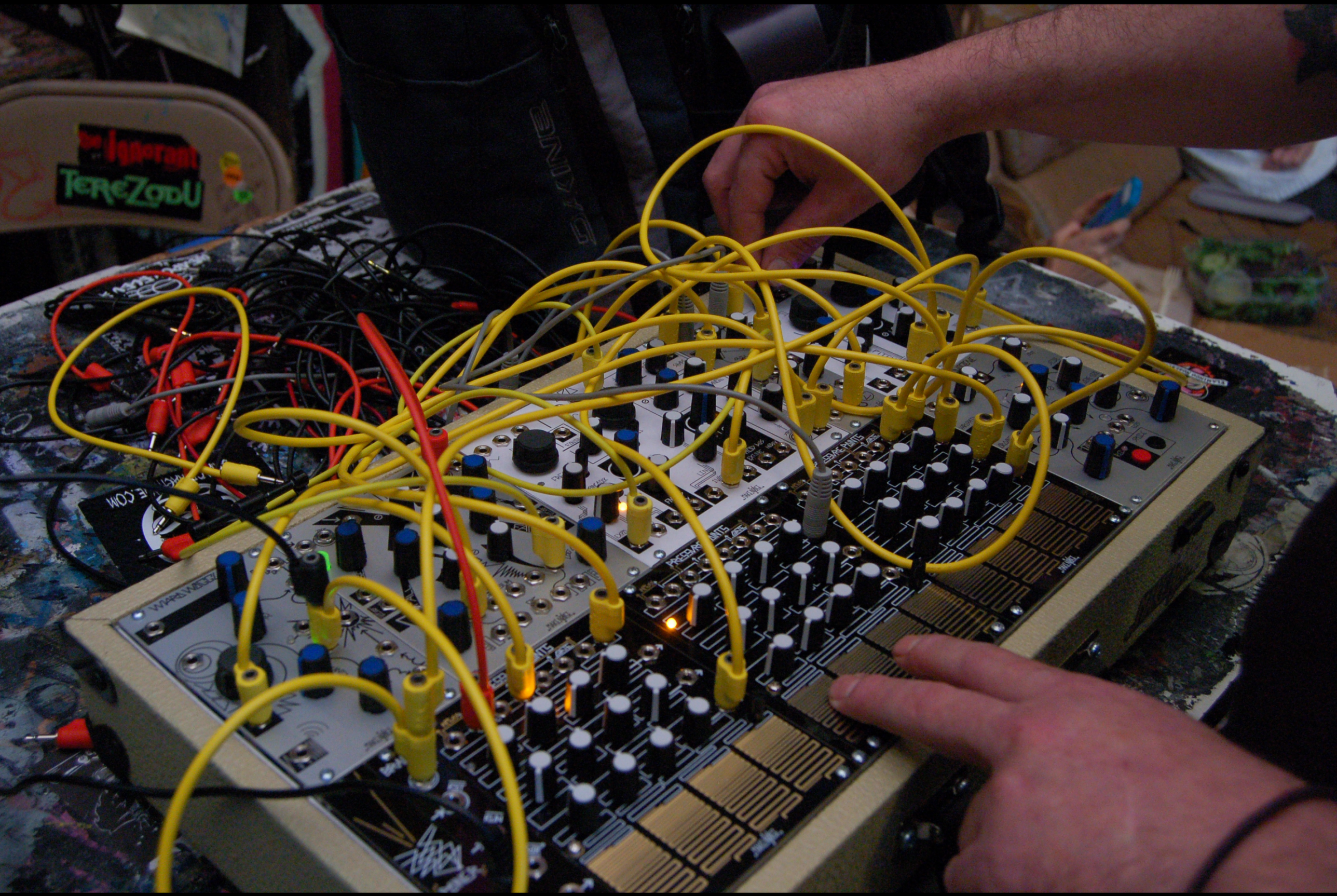

R
N
S
I
R

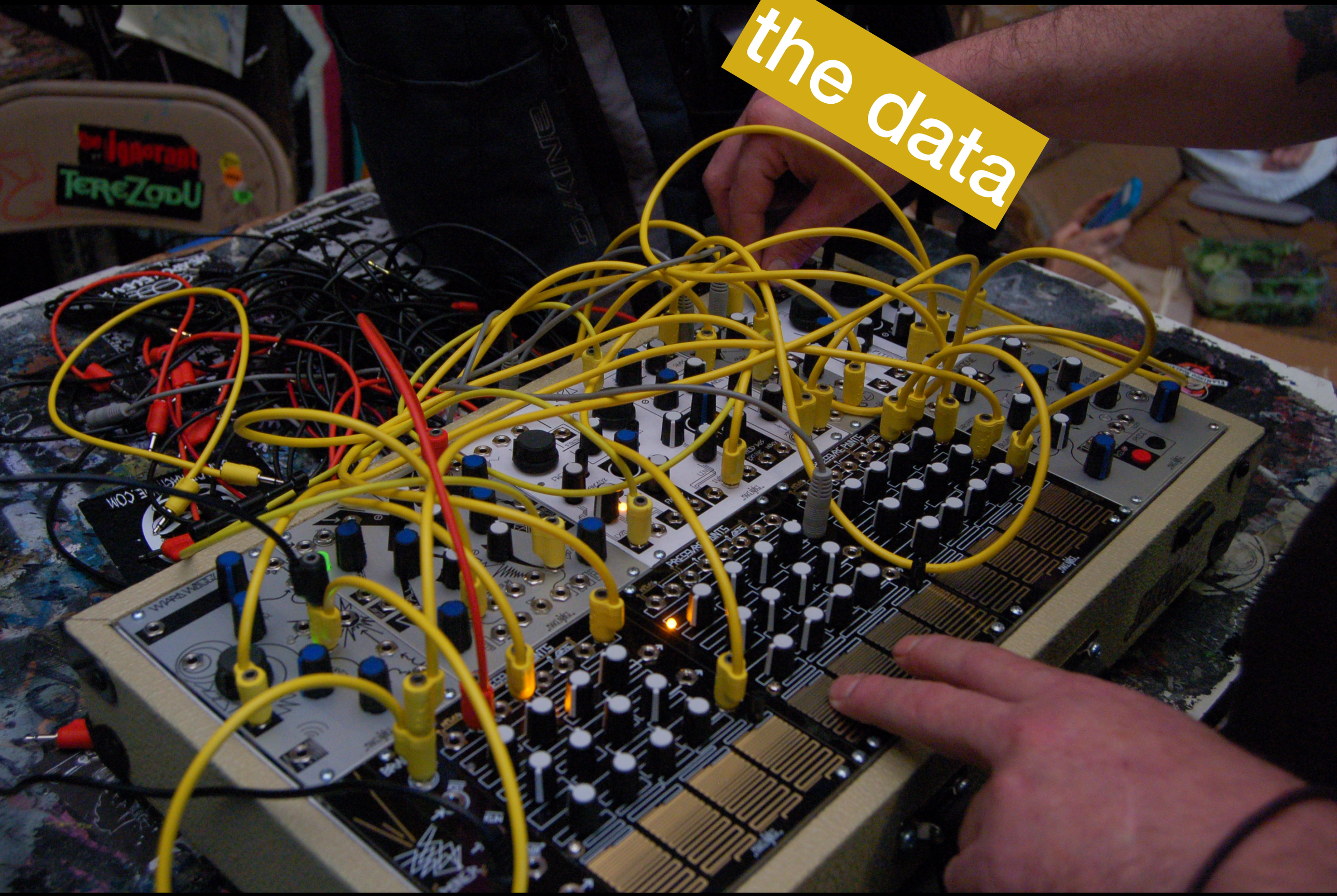


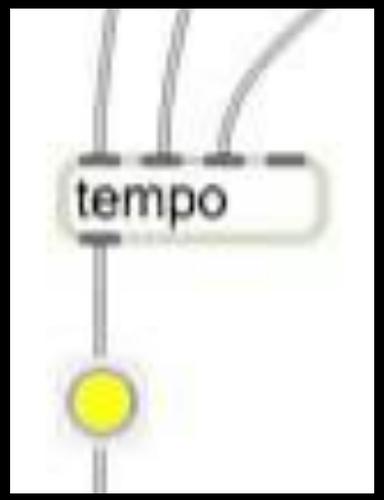

data
function
data

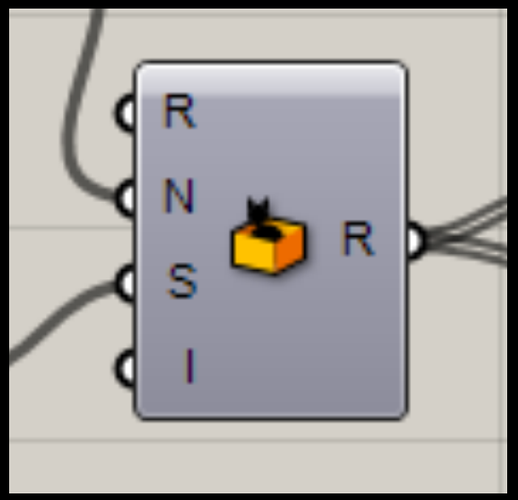


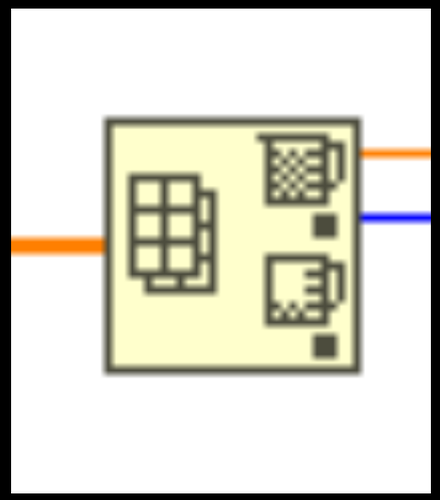
vs


TEREZODU

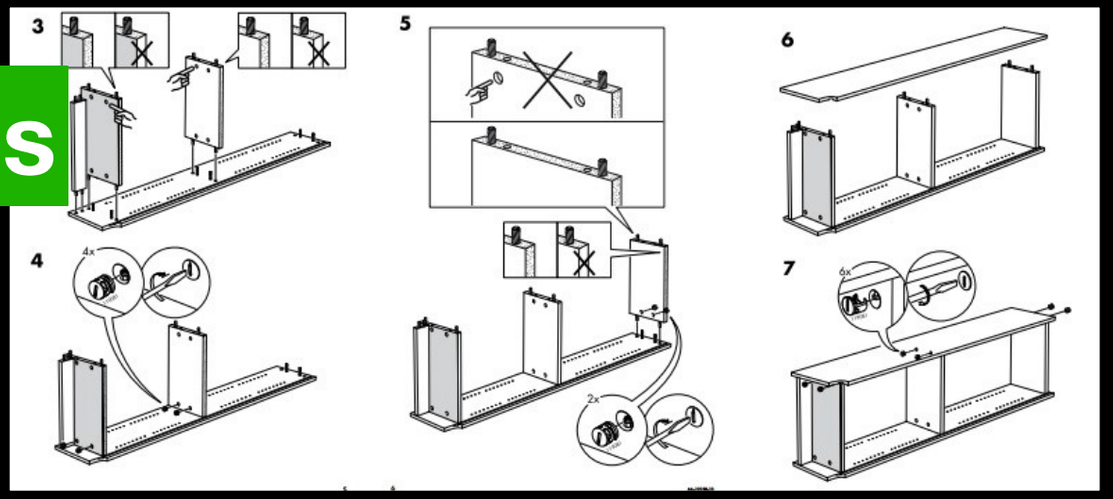

the data


tempo

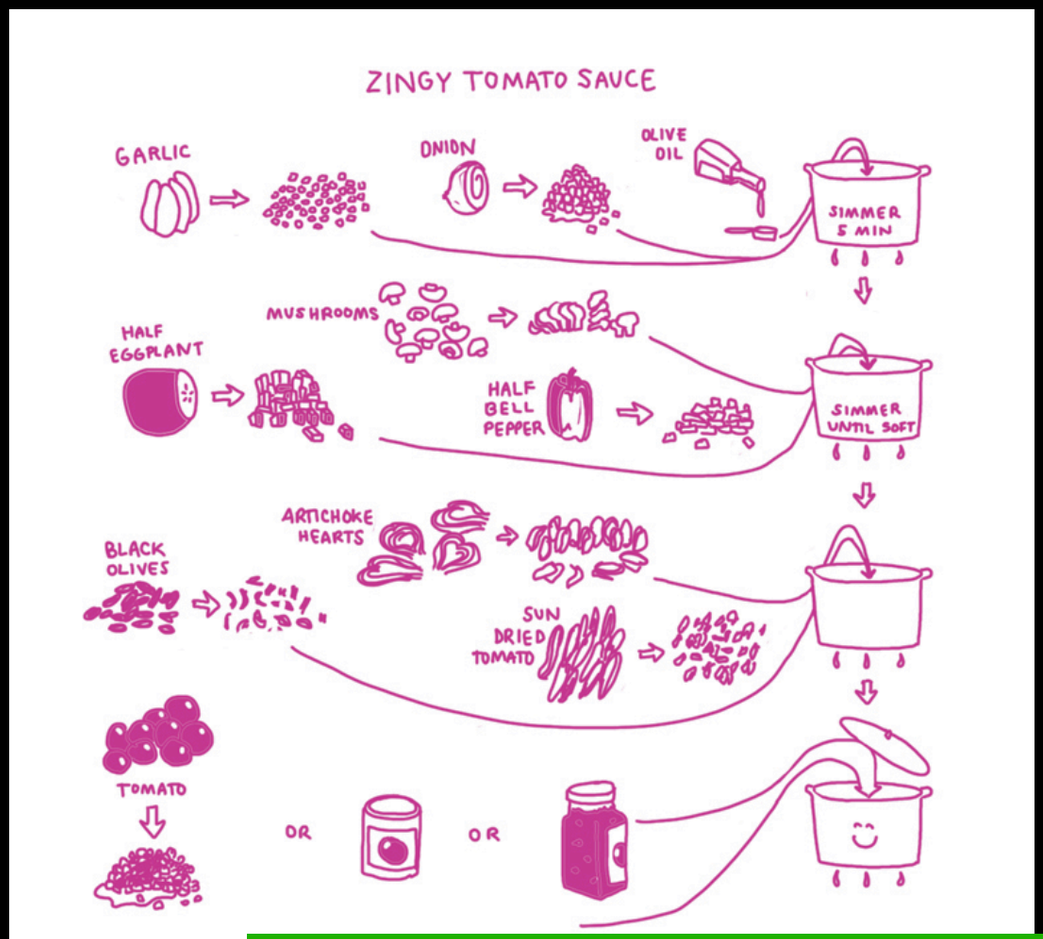

R
N
S
I
R



IKEA instructions



comic transitions (McCloud)



Picture Cook (Shelly)

PANE

adjective			
0	unrecorded	6	alive
1	bouncy		
2	lively		
3	resilient		
4	springy		
5	hot		

verb					
0	dwell	6	endure	12	know
1	inhabit	7	hold up	13	experience
2	survive	8	hold out	14	go through
3	last	9	exist	15	see
4	live on	10	subsist	16	undergo
5	go	11	be		

lists **join lists** ⊞

0	unrecorded	6	alive	12	go	18	be
1	bouncy	7	dwell	13	endure	19	know
2	lively	8	inhabit	14	hold up	20	experience
3	resilient	9	survive	15	hold out	21	go through
4	springy	10	last	16	exist	22	see
5	hot	11	live on	17	subsist	23	undergo

list **random choice** ⊞

be

1. PANE foregrounds the **data**

which provides …

- room to see the data
- handles for acting on data

2.

File Edit View Object Font Options Trace Extras Window Help

ampler9 | oliver leaman

rack random playback
mpler variables - volume
mp & slide, speed, pitch,
op time, reverb rate & time,
n & pan fade rate

all on/off | loops samples~ | MASTER VOLUME | random on/off | metro 500 | p randomiser | random 8 | metro 50 | metro 100 | metro 200 | metro 400

all random volumes jump on/off
all random volumes slider on/off
all random loop start/end time on/off
all random pans on/off

all reverb on/off (max)
dry/wet (max)
reverb level random

forwards & backwards | forwards only | p f~b~high | p f~b~med | p f~b~low | p f~b~v-low | p f~high | p f~med | p f~low | p f~v-low | sig~ | sel 1 | set~to~1 | p fade~selector | p random volume jump | p random~volume~slider | p random~pan | p random~pan~fade | p randomiser | random 3000 | random 6000 | select start ms | select end ms | groove~ sample~1 | edit only one end | undo | prop | record~ sample~1 | adc~ | graphics works | offset | set sample~1

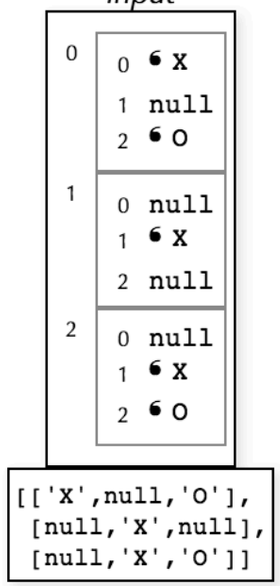
2 | 3 | 4

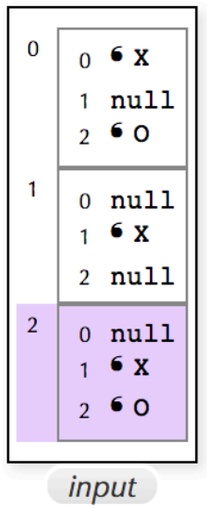
load sample~1 | load sample~2 | load sample~3 | load sample~4 | load sample~5 | load sample~6 | load sample~7 | load sample~8 | replace | buffer~ sample~1 | buffer~ sample~2 | buffer~ sample~3 | buffer~ sample~4 | buffer~ sample~5 | buffer~ sample~6 | buffer~ sample~7 | buffer~ sample~8 | double-click to open buffer~ window | s buffer_out

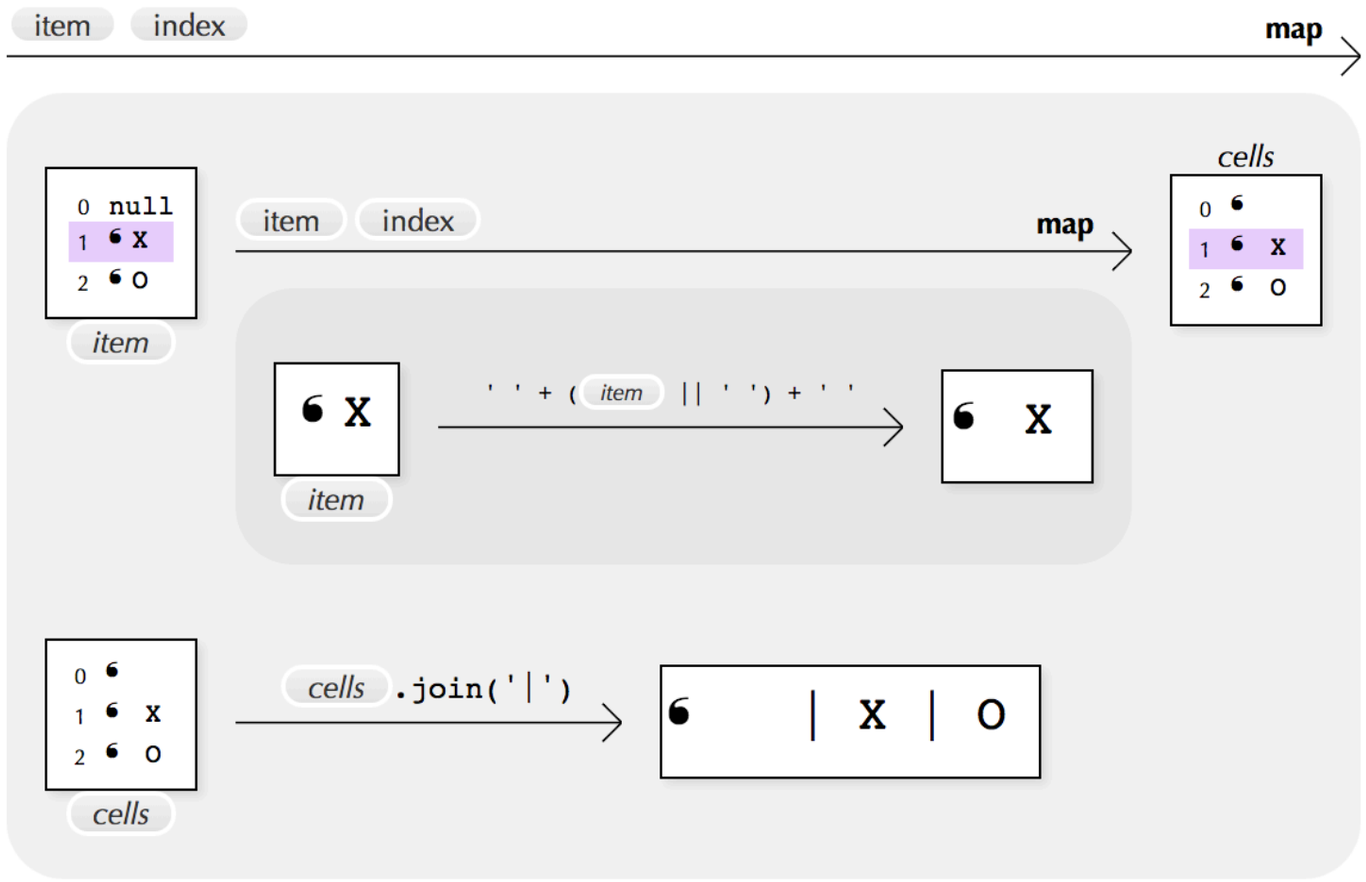
name of most recently loaded file | info~ sample~1 | info~ sample~2 | info~ sample~3 | info~ sample~4 | info~ sample~5 | info~ sample~6 | info~ sample~7 | info~ sample~8 | piano~chord1.aif | piano~chord2.aif | piano~chord3.aif | piano~chord7.aif | piano~chord8.aif | prepend set

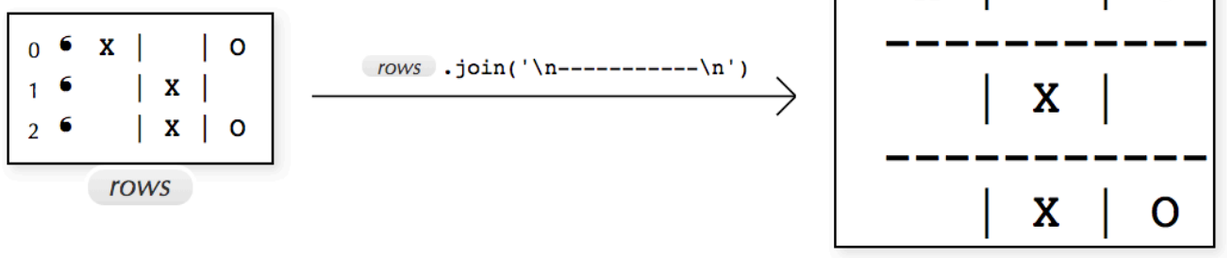
5 | 6 | 7 | 8

board to string

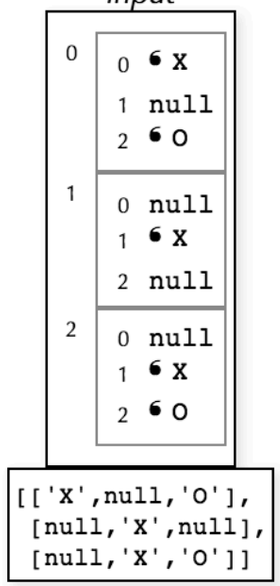


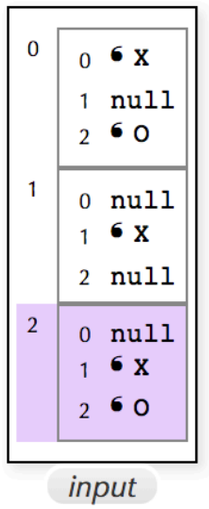


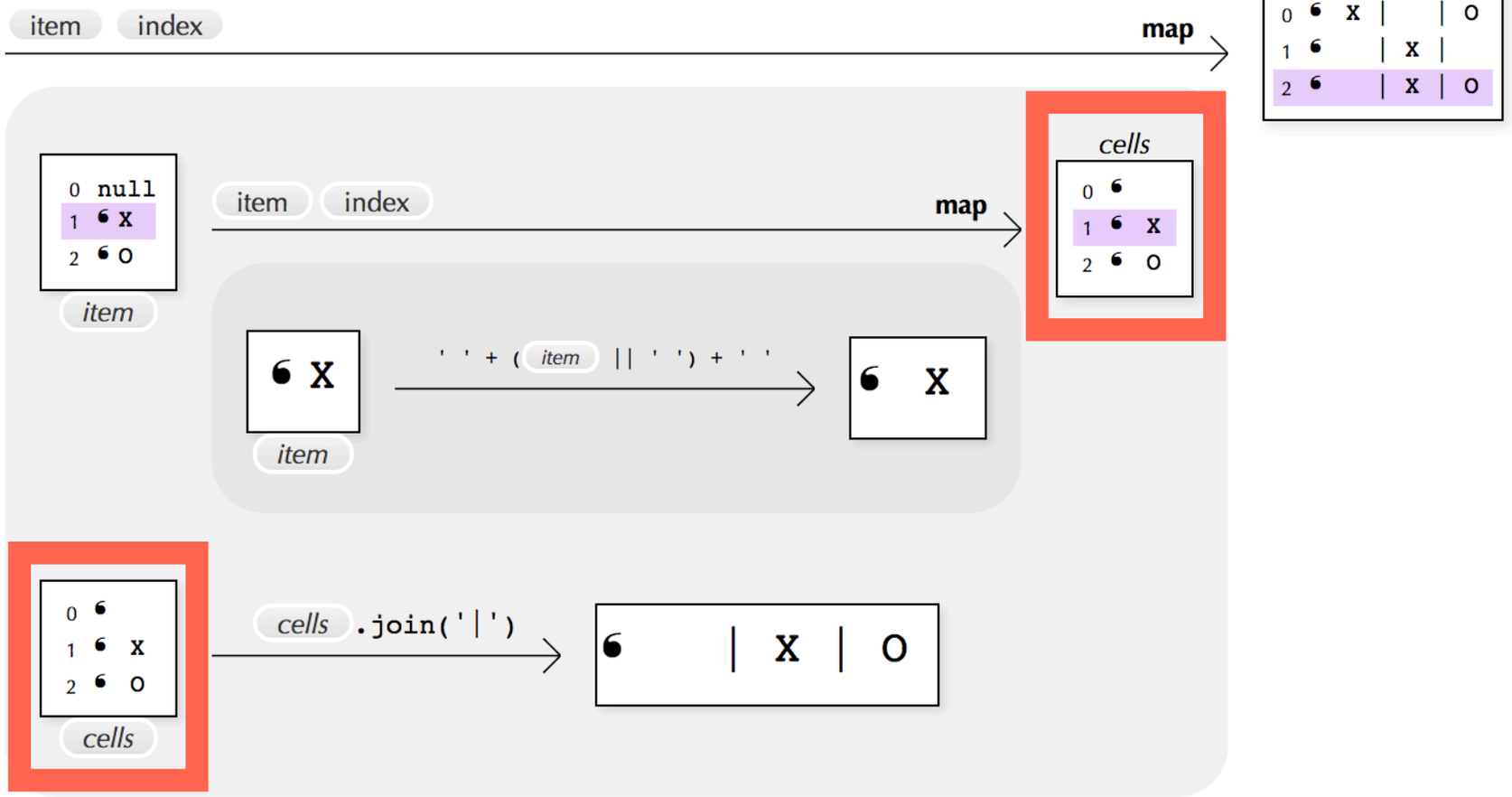


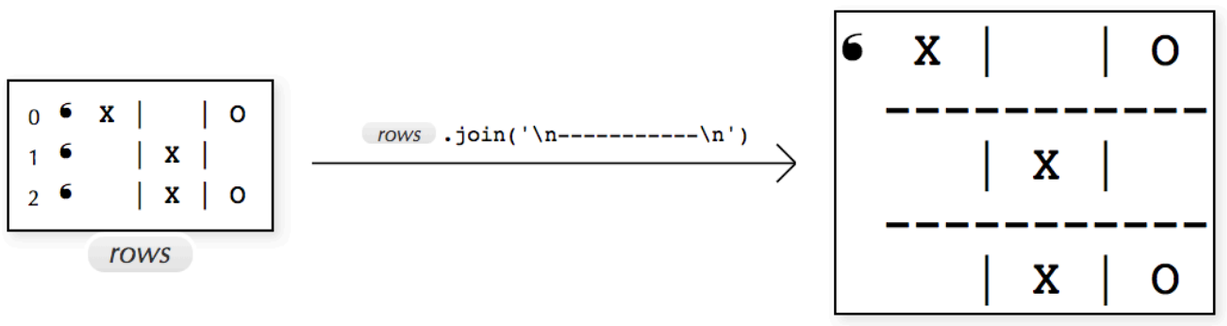


board to string









board to string

input

0	0	'X'
	1	null
	2	'O'
1	0	null
	1	'X'
	2	null
2	0	null
	1	'X'
	2	'O'

```
[['X',null,'O'],
 [null,'X',null],
 [null,'X','O']]
```

+

0	0	'X'
	1	null
	2	'O'
1	0	null
	1	'X'
	2	null
2	0	null
	1	'X'
	2	'O'

input

item index **map** → *rows*

0	'X'	'O'
1		'X'
2		'X' 'O'

item

0	null
1	'X'
2	'O'

item index **map** → *cells*

0	
1	'X'
2	'O'

item

'X' → ' ' + (*item* || ' ') + ' ' → 'X'

cells

0	
1	'X'
2	'O'

cells `.join('|')` →

```
'  |X|O'
```

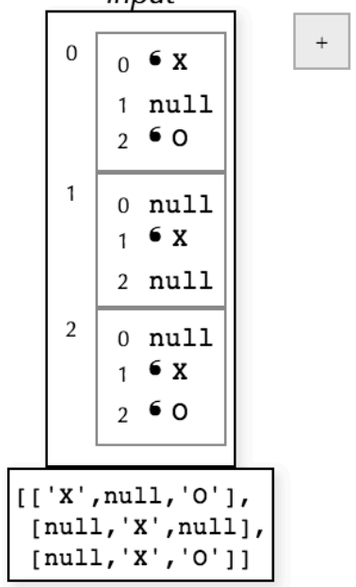
rows

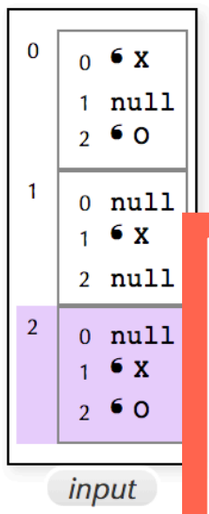
0	'X'	'O'
1		'X'
2		'X' 'O'

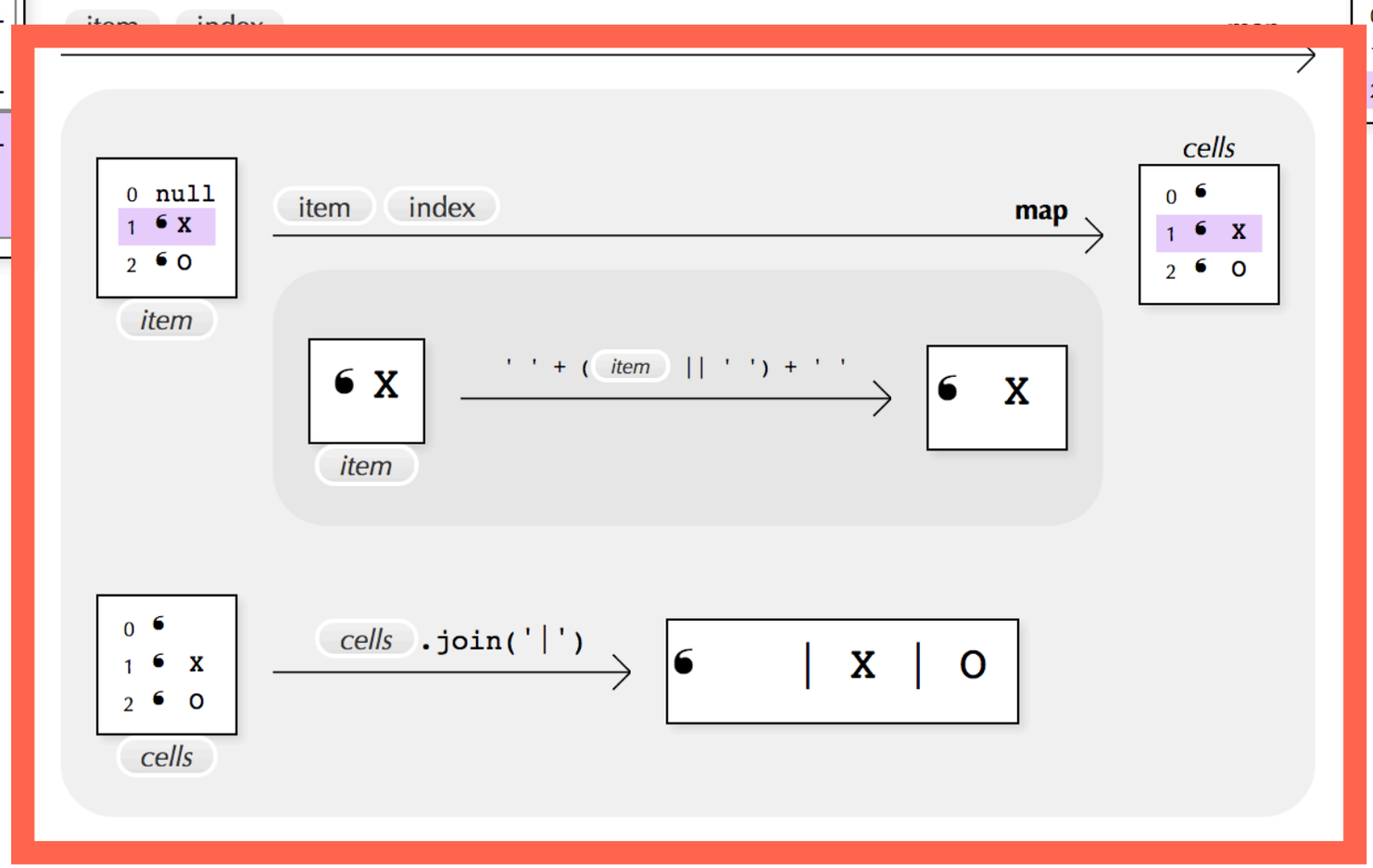
rows `.join('\n----------\n')` →

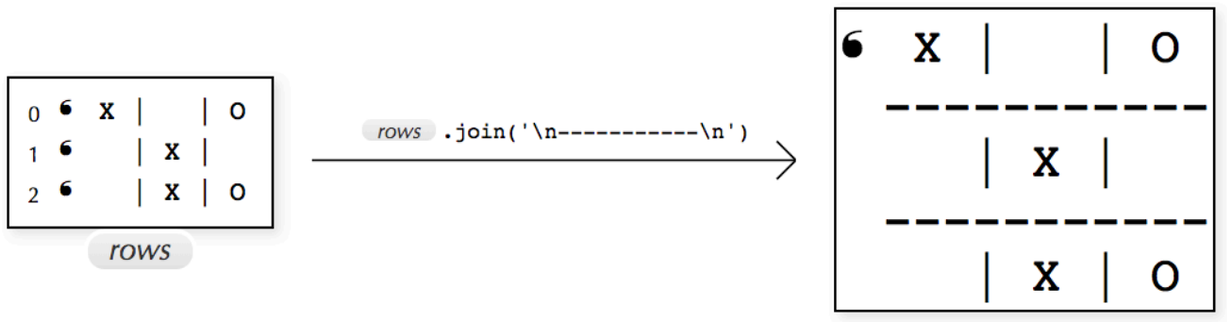
```
'X |  | O
----------
  | X |
----------
  | X | O'
```

board to string









2. PANE uses hybrid **metaphors**

which provides …

- legibility
- scalability

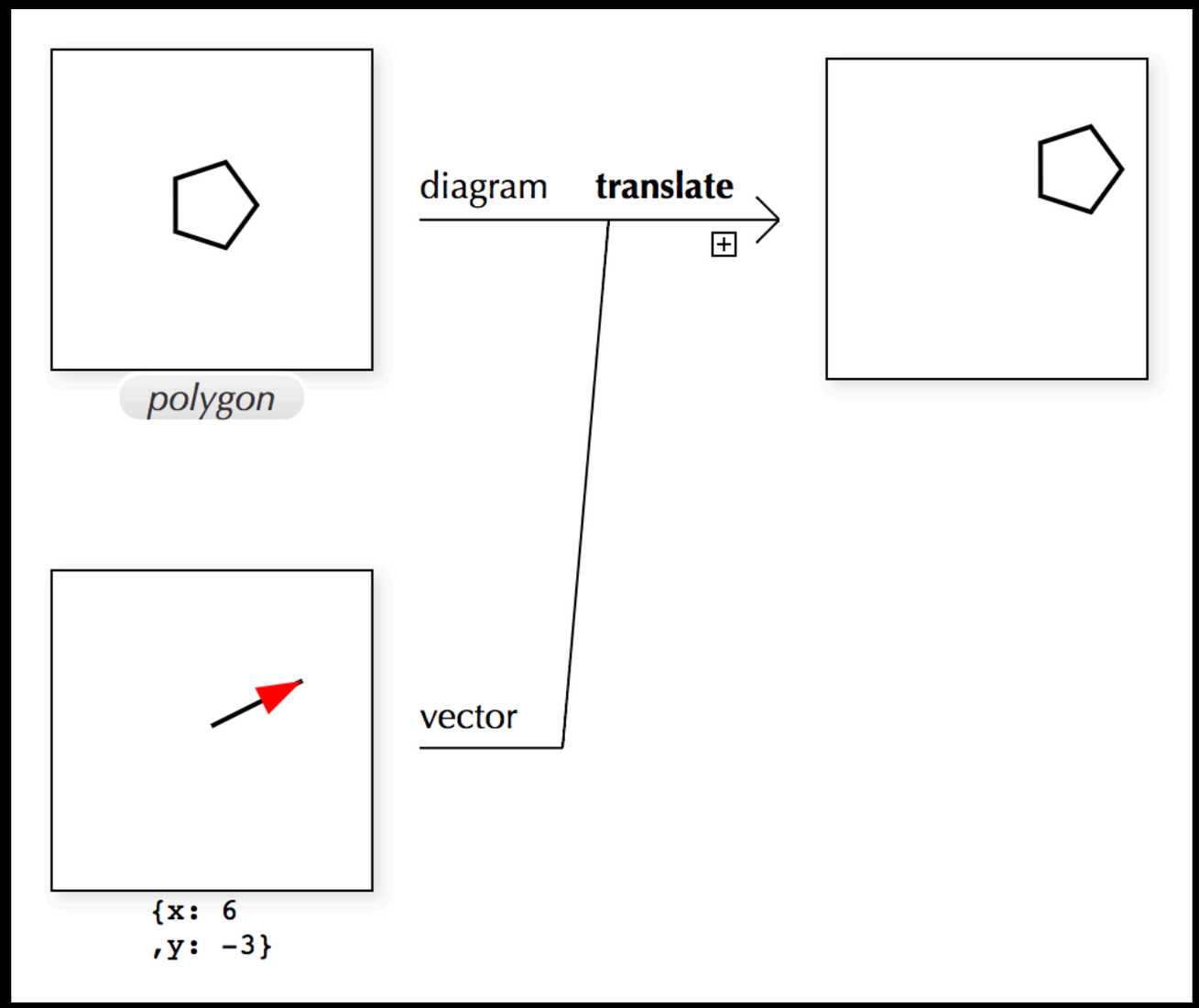
1. PANE foregrounds the **data**

2. PANE uses hybrid **metaphors**

Future work

1. Show data
2. Work with data
3. Connect with the world

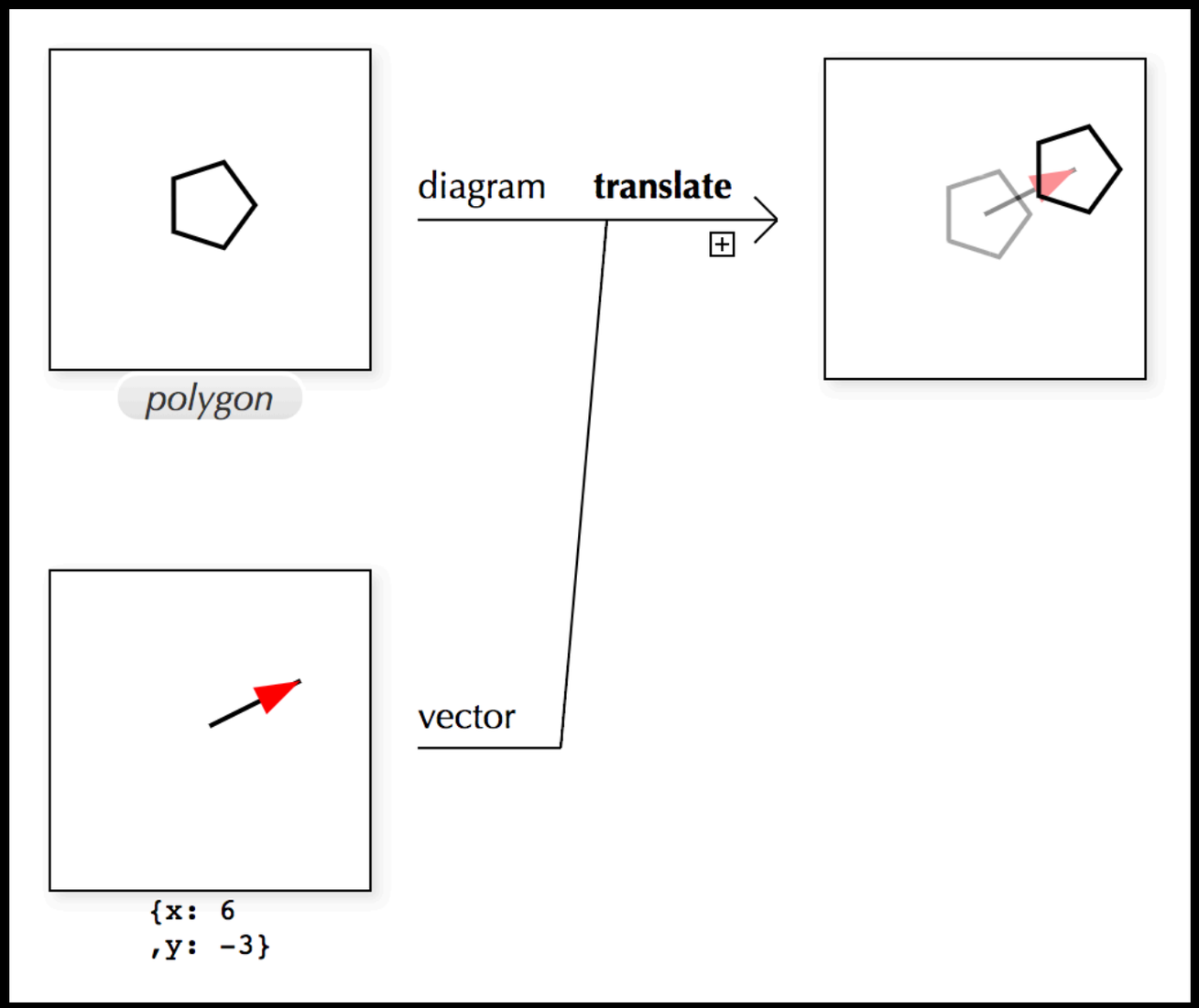
1. Show data



(present)

relationally

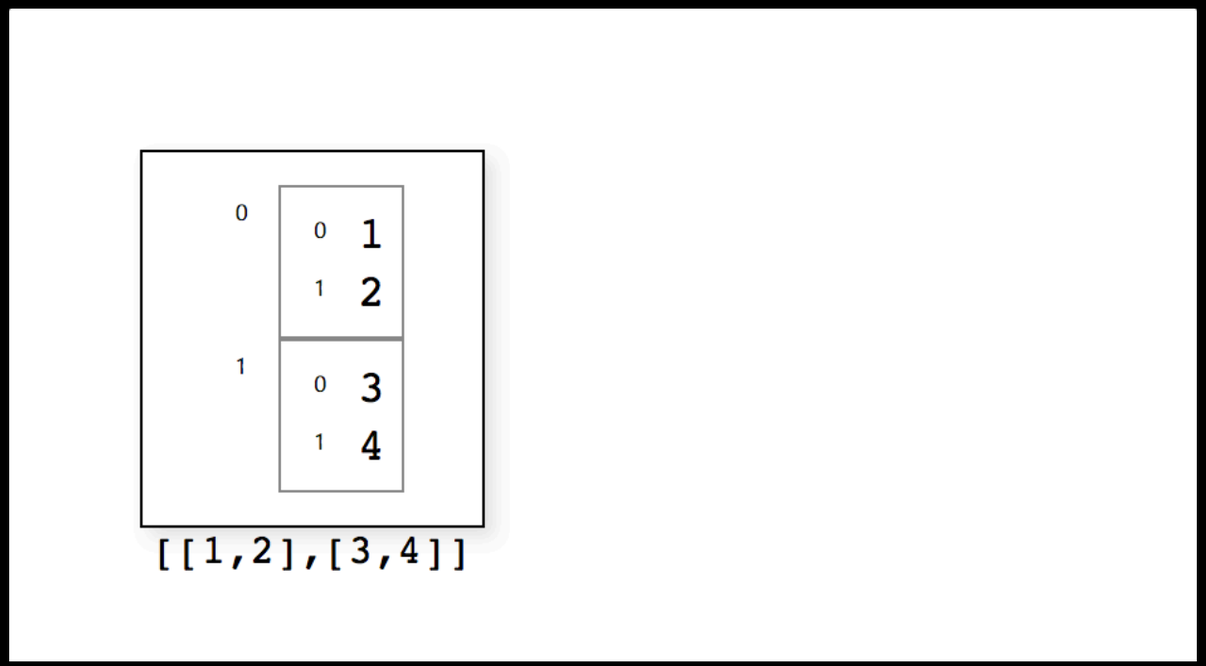
1. Show data



(future)

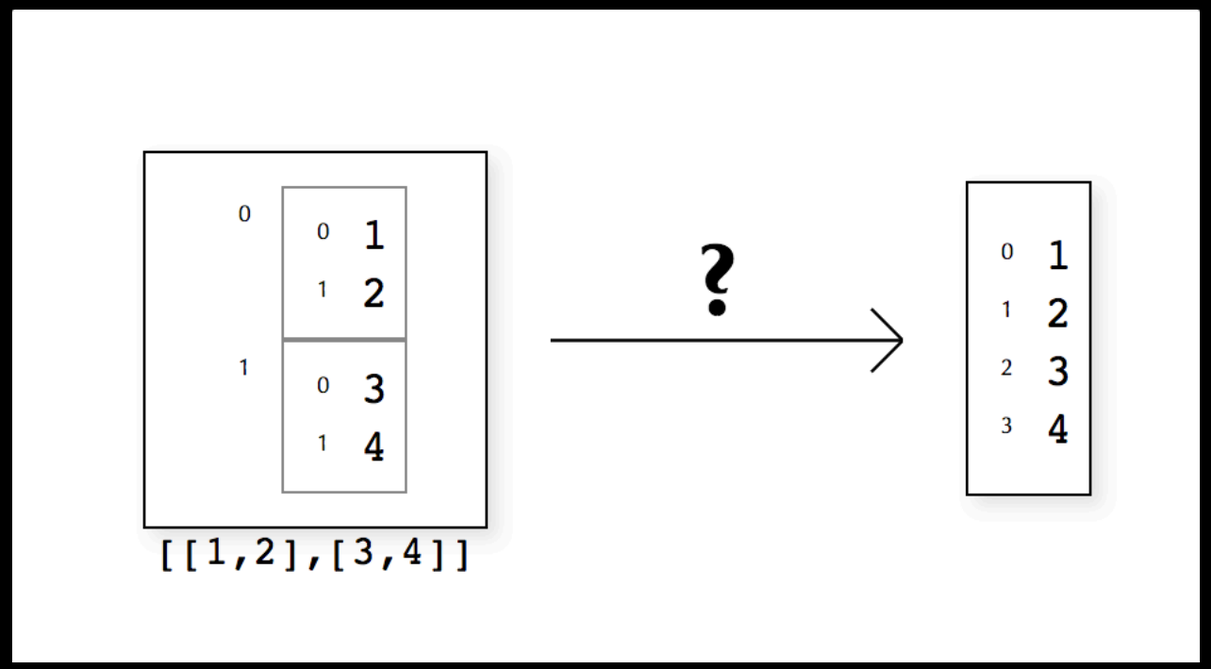
relationally

2. Work with data



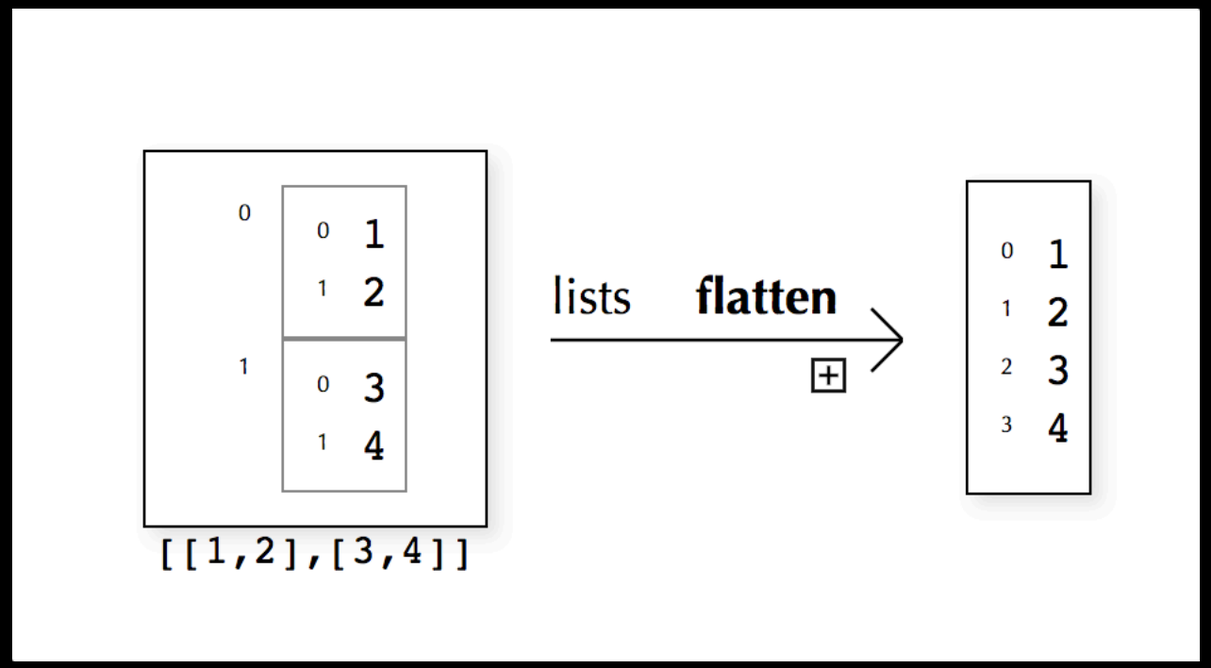
by example

2. Work with data



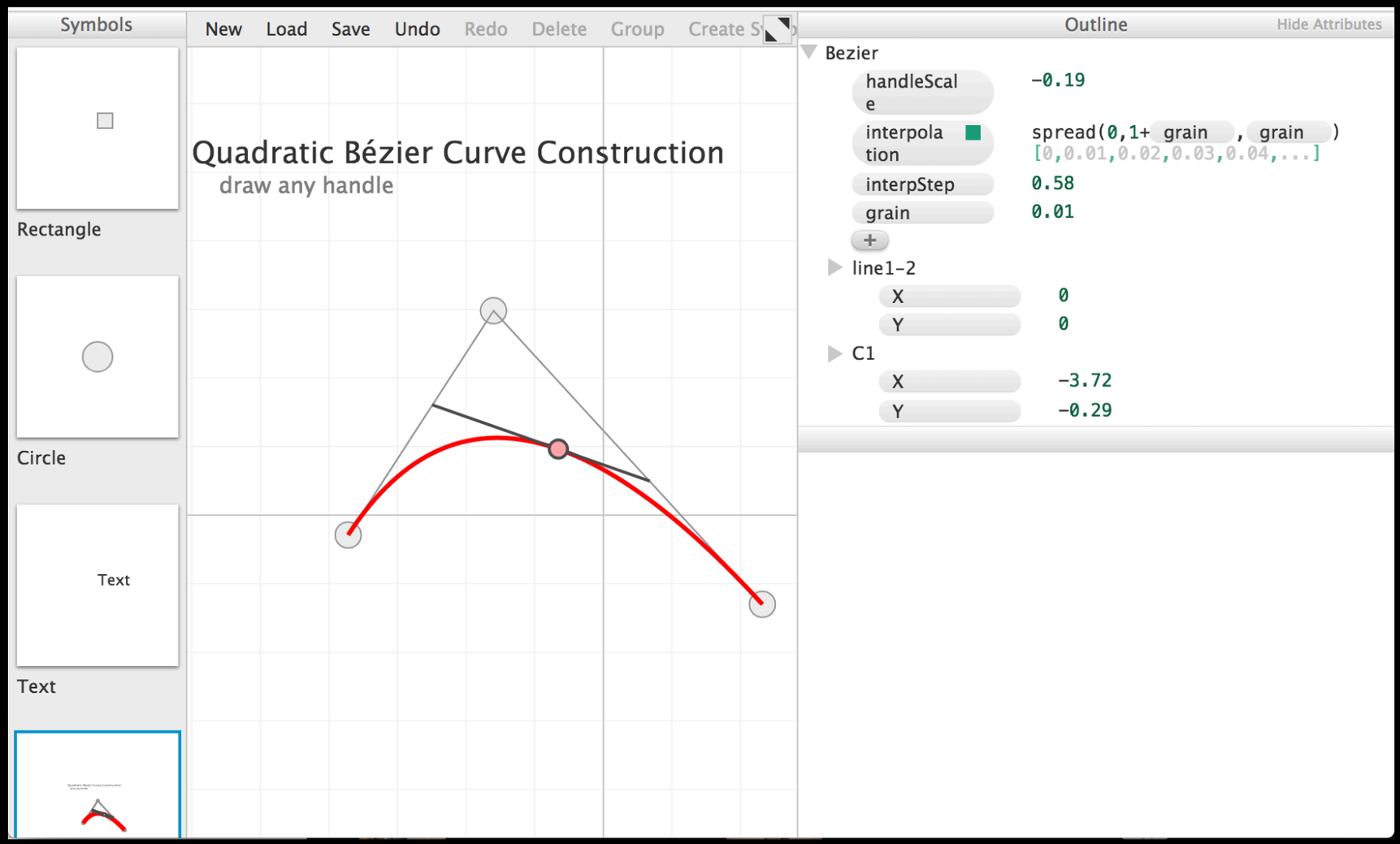
by example

2. Work with data



by example

3. Connect with the world



Apparatus (Schachman, et al.)

an ecosystem of heterogeneous live-programming tools

PANE 

Programming with visible data

LIVE 2018

Joshua Horowitz