

Blatt 09 - Basics 2

Kreatives Programmieren 1

Luís Antunes Pena

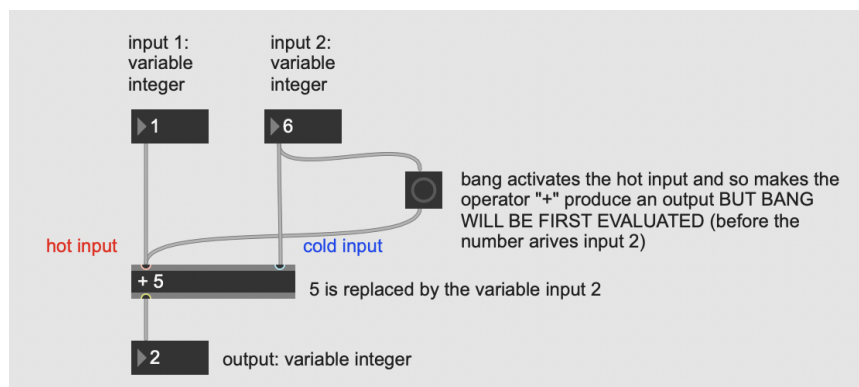
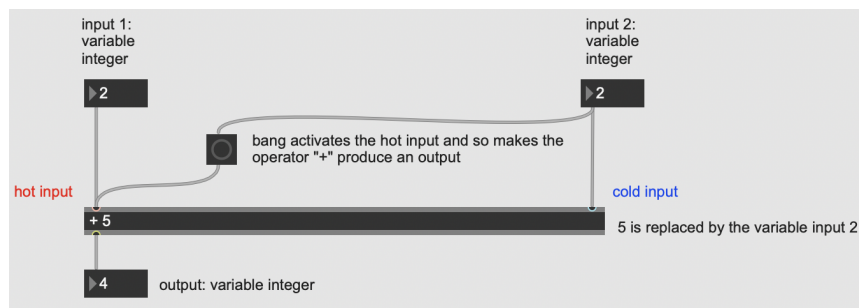
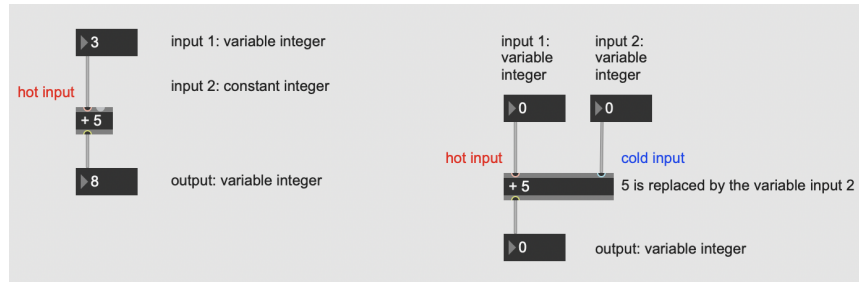
Februar 2024

Contents

1	Basic Functions, hot & cold input	2
1.1	Binary Operators	2
1.2	Inverse the order with !	3
2	Working with Lists	3
2.1	pack & unpack	3
2.2	advanced list processing ZL	4
3	Random Numbers	5
4	The coll object	5
5	Exercises	5
5.1	Make a patch with five voices, each one of them can play one or two per random selected tones. Each voice has different probabilities to be activated.	5
5.2	Make a similar patch but now the notes shouldn't be randomized but the should be sequentially played in a loop containing 3, 4 or 5 notes previously selected.	5

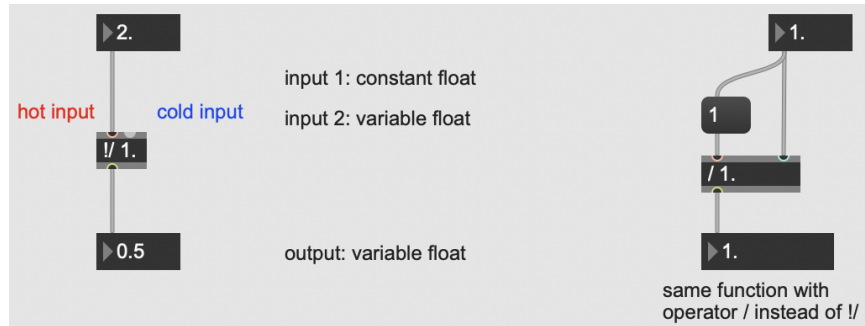
1 Basic Functions, hot & cold input

1.1 Binary Operators



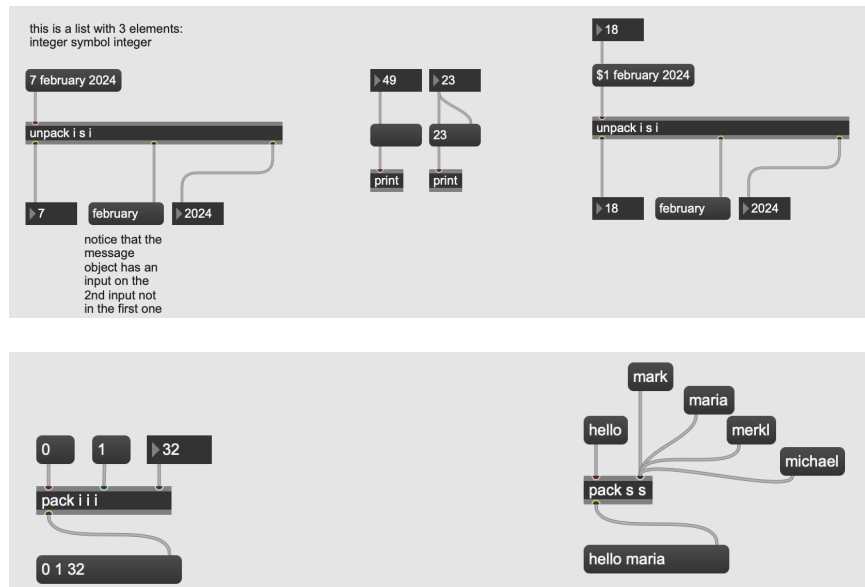
1.2 Inverse the order with !

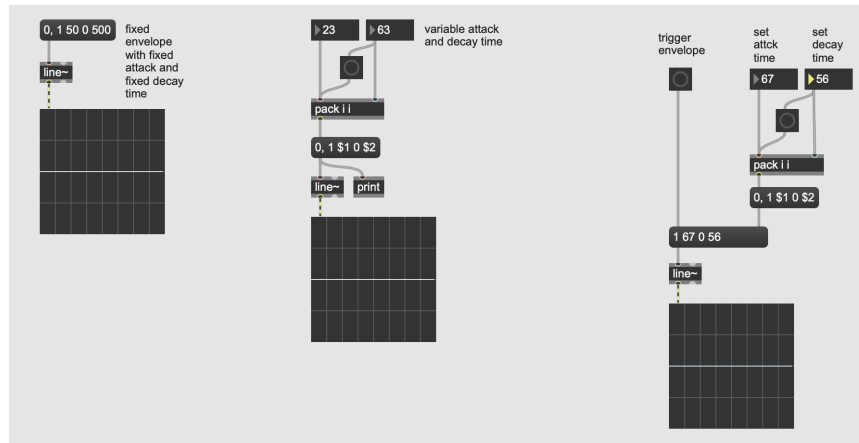
Arguments of the function `!` are reversed:



2 Working with Lists

2.1 pack & unpack





2.2 advanced list processing ZL

zl

Process lists in many ways
Performs several kinds of list processing functions. You set the function with an argument, and can change the function performed with the mode message. The behavior of the zl object's inlets and outlets vary according to the selected mode.

The zl family of objects can also be instantiated like this: `zl.change` `list.change`

Alternatively, the object can be instantiated as 'zl' and the first argument sets its mode of operation: `zl change`

zl modes that accept right inlet messages can be initialized with an argument:

An optional integer arg sets the maximum list length for all modes of operation for the zl object. The default list length is 256, and the maximum is 32,767:

Arguments:
length (int) (optional)
mode (symbol) (optional)
function-int (int) (optional)
function-list (list) (optional)

zl.change filter out repetitions

zl.compare compare two lists

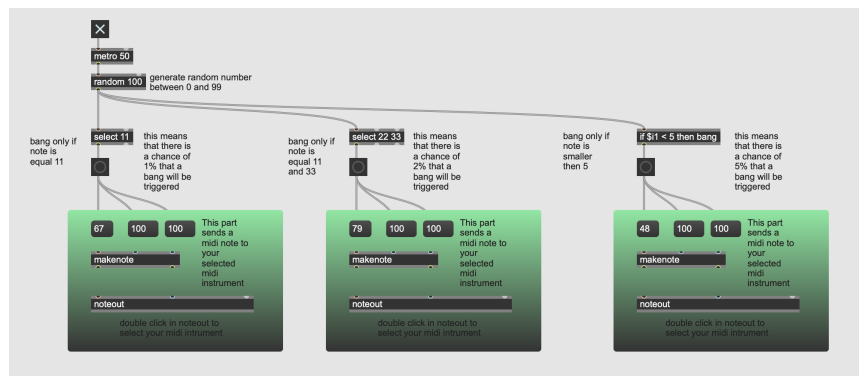
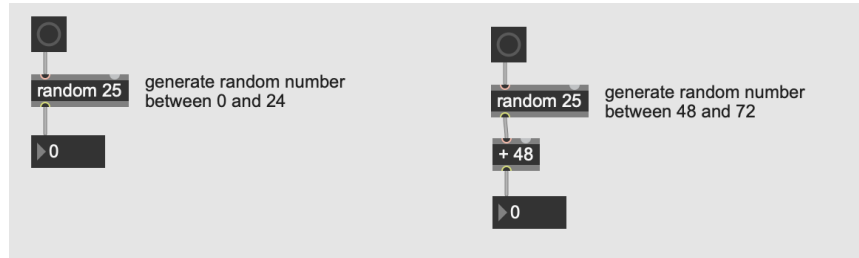
zl.delace de-interleave a list

zl.ecils slice a list in reverse order

zl.filter remove items in a list

zl.group group items into a certain sized list

3 Random Numbers



4 The coll object

5 Exercises

- 5.1 Make a patch with five voices, each one of them can play one or two per random selected tones. Each voice has different probabilities to be activated.
- 5.2 Make a similar patch but now the notes shouldn't be randomized but the should be sequentially played in a loop containing 3, 4 or 5 notes previously selected.