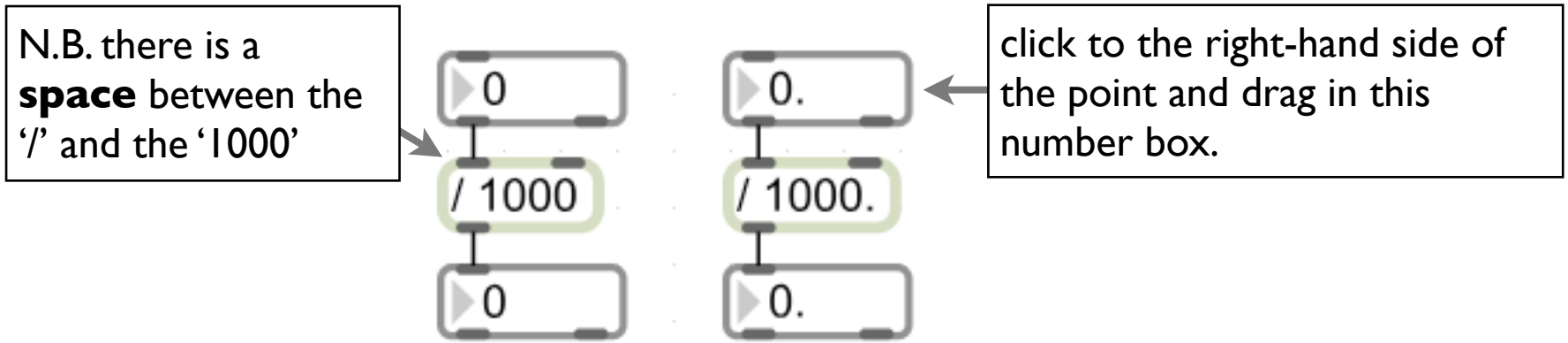
Max/MSP exercises 1b

Ex. 1

1. There is an obvious difference between these routines. But how does it change the outcome?



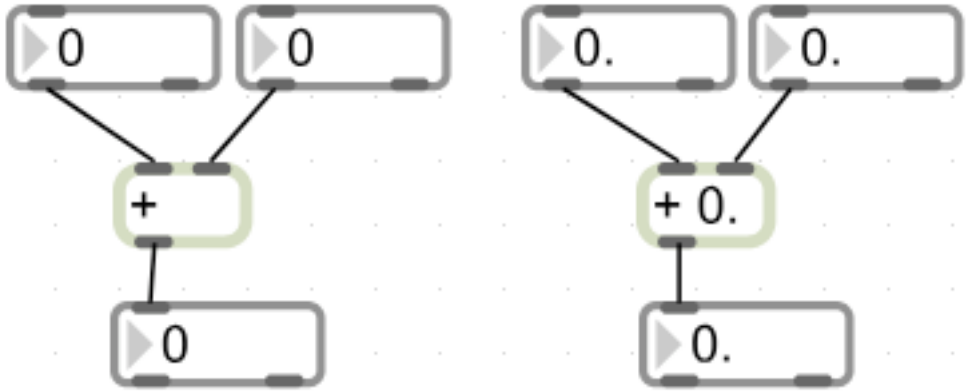
2. Try changing the [/ 1000.] box to [/ 1000]. What happens now?

What might be the reason for these differences?

Ex.2

This introduces a key issue with Max/MSP: **hot & cold** inlets.

1. Build these routines and try changing the various number boxes. Here we come across the same decimal point issue again. Why is this?



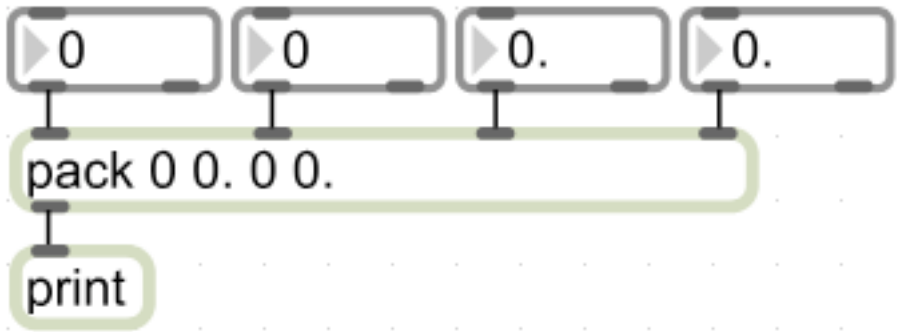
2. When is the bottom [number box] updated?

3. Try replacing the [+] with other mathematical symbols: [-], [*], [/], [%] (N.B. '*' is multiply, '/' is divide, '%' is… well, see if you can figure it out).

Ex.3

Reiterating the above 'hot&cold' issue…

1. Change all four number boxes in this patch and check the Max window.



2. When is the Max window updated?

3. What do you notice about the second and fourth items in each list?

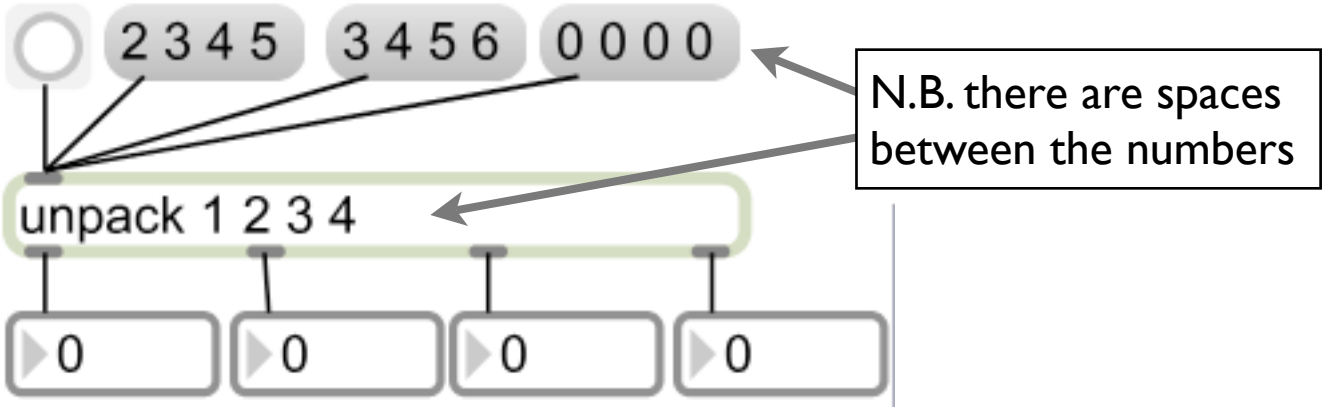
We have now come across four types of messages in Max:

- bang
- integer or int (whole number)
- floating point or float (number with decimal point)
- list.

These will be covered in more detail later on; for now, just know that they exist!

Ex.4

1. Copy this routine, then hit the [button] object. What happens?



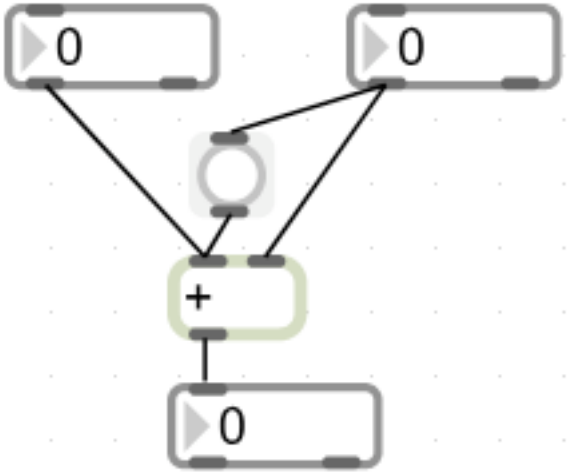
2. What happens when you hit the other [message box]es?

Some object will therefore store data and send it out when they receive a bang. This may not seem useful now, but it is...!

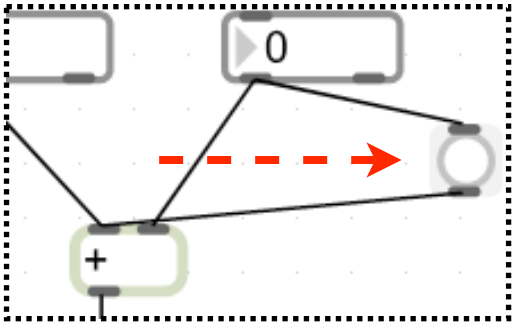
Ex.5

Here's another of the key issues to get your head around in Max.

1. Copy this routine. What is the button object there for? How is this an improvement on Ex.2?



2. Move the [button] to the right of the [number box] as shown. Try the patch again. What's changed?

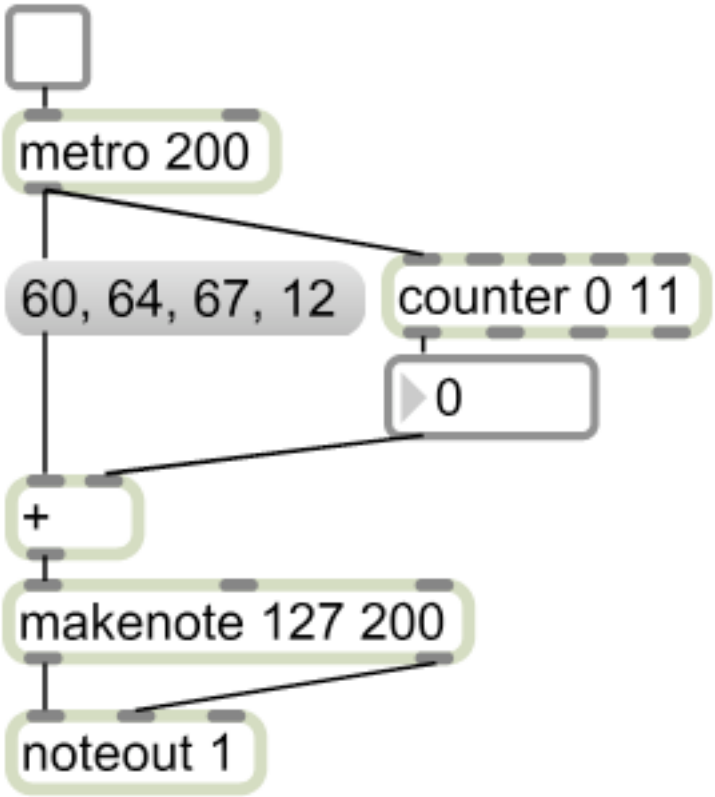


3. Try different positions for the button. Where exactly does this change take effect and what do you deduce from this?

Ex.7

1. Copy this routine. What does it do?

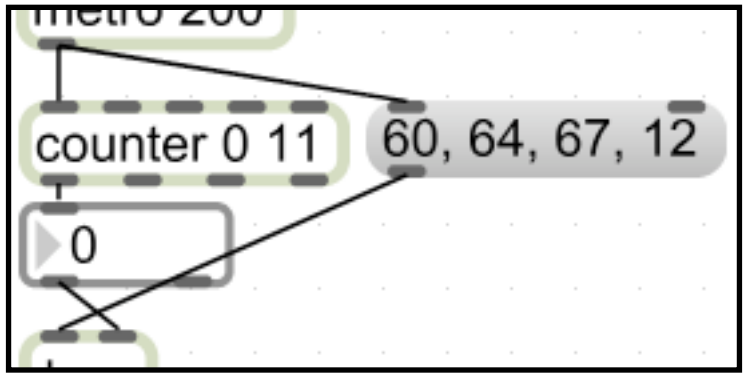
2. You will notice that the numbers in the [message box] are separated by commas. What difference does this make? Connect a [print] object to the outlet of the [message box] to find out. Then try removing the commas and see what happens.



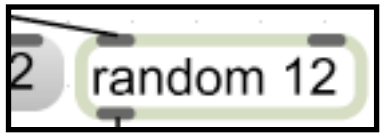
So what's happening? Numbers output from the [message box] are sent (via the [noteout] object) to the Mac's internal synth, which interprets them as MIDI note numbers and therefore plays notes. Four notes are sent at the same time from the [message box], yielding a chord, and adding to each number means that these chords are transposed. You should notice that semitones are added according to the number output from the [count] object.

Ex.7 (cont)

3. What happens (apart from it looking ugly) if you swap the positions of the [message box] and [counter] objects as per the following? (If you don't see anything, try slowing down the metro and watching the number box. Are the calculations correct?)



4. Swap the [counter] and [message box] back. Now replace the [counter] object with the following:

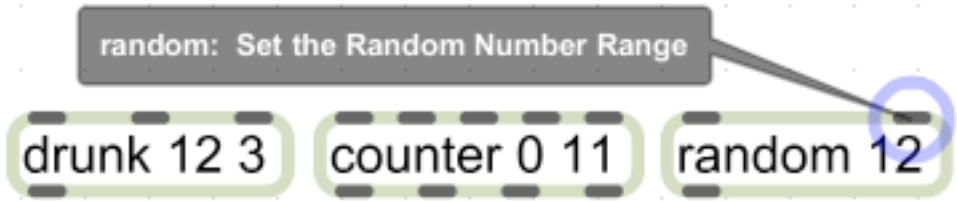


...then:

What do you get in each case?

Ex.7 (cont)

You might already have noticed that when you hover your mouse over the inlets and outlets of any object, they are circled (red indicates a 'hot' inlet; blue indicates a 'cold' inlet) and a speech bubble appears with a hint which tells you what you can connect to that inlet.



5. Use these hints to figure out how to change the range of all three objects in the previous patch (and the step-size of the [drunk] object).

You now have a vocabulary of objects which should (with a bit of thought) allow you to build a few quite interesting routines in Max that could be used for musical purposes, including a simple arpeggiator and tune generator. If you're at a loose end, you might try these. Even if you are unsuccessful, you will be training your brain to think in a Max-like way and becoming more fluent with the interface at the same time.