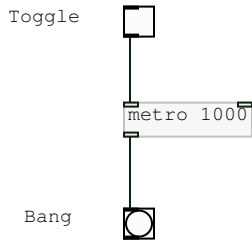


MAT120 Pd Examples

Example 1: metro with toggle and bang

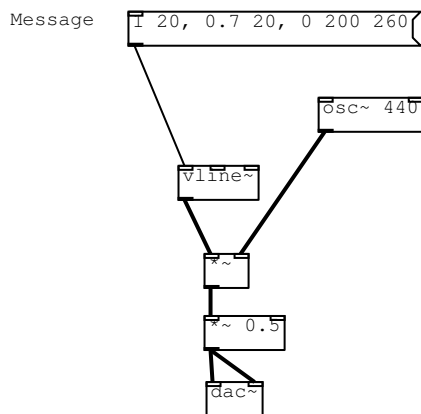
The toggle box is like an ON/OFF switch. It is connected to the metro 1000 box on the left, or hot inlet. Note: It doesn't make a sound at this point, just outputs a bang on one second intervals.



Use this patch to see how toggle can turn metro on and off, and how metro sends out a bang every 1000 milliseconds (ms). Edit 1000 to change the rate of the metronome.

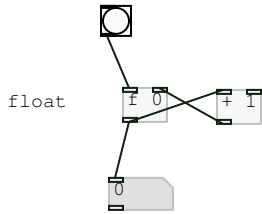
Example 2: vline with osc, signal multiplication

Vline is a signal output box, like osc. The output in either case is a function which you can think of as a bunch of dots on a graph. The dots are coming out at the rate (called the sample rate) of 44100 per second. That's a lot of dots! These are all numbers which represent a graph such as an envelope function (vline) or a sine function (osc).



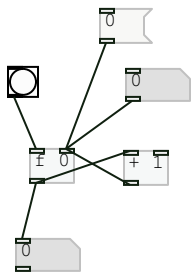
Use this one to play A440 with an envelope. The envelope ramps up to amplitude 1 in 20 ms, then down to 0.7 in another 20 ms, then sustains for 260 ms and then ramps down to 0 in 200 ms. So the total envelope time is 500 ms. Edit the times to change the envelope or edit the frequency 440 to change the pitch.

Example 3: float box



This patch uses a float box to store a variable number initialized to zero. The bang in the upper left or "hot" inlet pushes that number out through the lower outlet, where it is passed on to the number box and also to the plus one box's left or "hot" inlet.

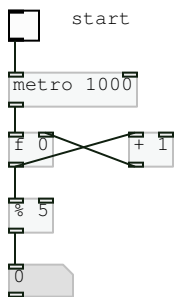
The plus one box receives the number from the float box and adds one to it. Then it outputs the new value to the right inlet of the float box. The right inlet is called "cold" since it does not prompt an outlet. It simply stores that value as its current value without outputting it. When the next bang occurs, that value is pushed out.



In this patch we have added a message box with zero and a number box with zero to the cold inlet of the float. These can be used to reset the stored value of the float to zero by clicking on the message, or to any value by changing the number box value by clicking and holding then sliding up or down with the mouse. Note: the output in the lower number box can also be changed but this is overridden by any values coming from above.

Example 4: mod function: %

Modular arithmetic uses remainders.



The box `% 5` takes input from the

f box and gives the remainder after division by 5. This means that the number box at the bottom will read 0 after it reaches 4, and then it will continue to cycle through 0 to 4.

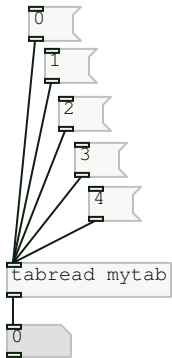
Example 5: indexed tables with Pd

```
table mytab 3
```

creates a table called mytab that can store 5 things note: table is a reserved word in Pd, but mytab is a name chosen for the table by the user.

```
;
mytab 0 7 -2 3
```

when this message is clicked it puts the five numbers 7 -2 3 8 13 into mytab, with indices: 0 1 2 3 4

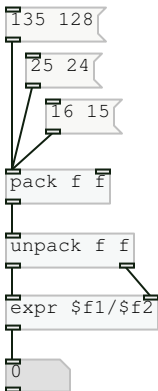


click on a message to send an index value to tabread and then watch the value with that index in mytab come out the bottom.

note: tabread is also a reserved word in Pd

Example 6: pack and unpack

This example is used to convert from the fractional form of a frequency ratio to the decimal approximation form. Note: the number box does not display all of the digits that are actually saved in this number. There are about 10 digits of accuracy in addition to the 4 that are displayed.



click on a message to send the two values to the pack function box.

pack receives two values from the previous message and puts them into a list format.

unpack takes the two values and prepares to send them on to another function.

expr \$f1/\$f2 takes the two values in order from left to right and forms the quotient.

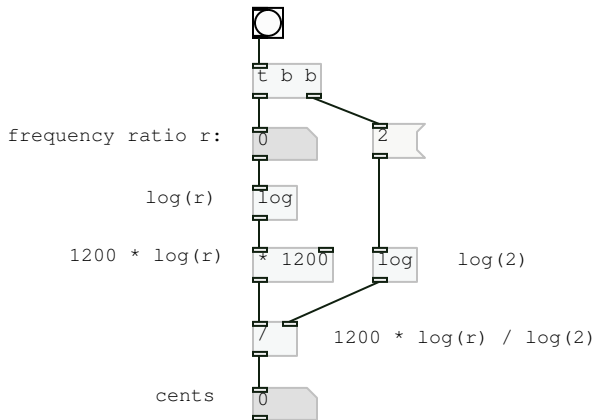
the final value is printed out as a decimal.

note: pack, unpack, and expr are Pd reserved words

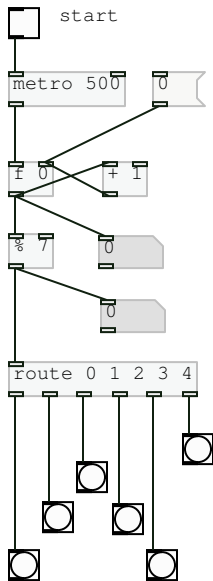
Example 7: trigger box, and frequency ratio to cent conversion

this computes the cent value of a frequency ratio

`t b b` sends out 2 bangs in order from right to left



Example 8: route function



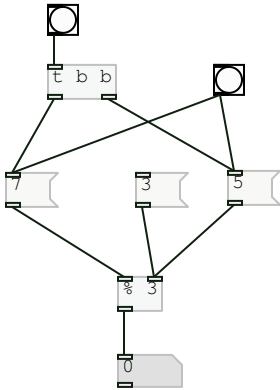
comment

This patch illustrates the route function. Route is a like a branching function which takes the inlet and sends it through one of several different outlets according to what that inlet is.

In this example, the possible inlet values that are recognized by route are 0, 1, 2, 3, 4 Notice that this is 5 values, however there are in fact 6 outlets. The sixth outlet is used if the inlet does not match anything from the list 0, 1, 2, 3, 4

If you push the start button you will see the bang boxes below light up on the half second. The first 5 will respond to the numbers 0, 1, 2, 3, 4 and then the sixth one will respond to the other numbers 5 and 6 Since we are using % 7, the values will cycle through 0 to 6

Example 9: order of bangs with mod



This patch illustrates the order of bangs sent out. The two bangs on the top can be used to send out two more bangs to the messages with numbers 7 and 5. However, they have a different effect on the output at the bottom because they have different orders. In particular: by clicking the number message 3, we are putting the 3 into the cold inlet of the % box. This means that the number 3 is the modulus, or the number which we divide by in order to find the remainder. The remainder after division by 3 is what comes out at the bottom. This is why we see 1 at the bottom. If we push the 5 message, the modulus is changed to 5 and so now when we push 7 the output is 2. On the other hand, we can use the upper right bang to hit both the 5 and the 7. The desired effect would be to get 2 out the bottom again. To test this, push 3 again and then push the upper right bang. The output is 1. Why?

The output is 1 because the bang box is being sent out with the order left then right. This way the left bang first hits the 7 which goes all the way through the hot inlet of % and pushes out the remainder 1 since the modulus is still 3. The other bang is too late with the 5, so it stores the modulus as 5 only after the 7 has gone through. At this point we can again push the 7 and see that we get a remainder of 2 since the modulus is now 5.

Finally, we can run the same test on the trigger box, which should send out the two bangs in the order from right to left. So, first push 3, then test 7 to get 1. Now push the top bang above trigger and get 2. This time the 5 gets loaded first, before the 7 goes through and so the remainder is indeed 2.