

## Randomness

random function: outputs a random number. Input can be by bang (left), int (right) or seed (left).

**example 1: create a random box, set its argument to 10, and hook it up to a bang and a number output box. test.**

Note that the random generator outputs anything up to 1 less than the argument. This is because 10 means "10 numbers" not "a max of 10", hence 0-9. The number box reports the output.

**example 2: add a message box to the right inlet of the random function, and set it to a higher number than 10. test.**

Note the difference; the inlet resets the argument (although not visibly).

Another way of inputting numbers is with a number box. You cannot change a number box's state in the edit window, but you can change its state in the run window, by dragging over it.

**example 3: add a number box to the right inlet of the random function, and test.**

Now add a slider to the right inlet of the random function. You set the slider by clicking on it and selecting Object>Get Info from the menu, or command-I. The number set in the top box is the argument, that is, the maximum setting of the slider (with 0 as the lowest).

**example 4: add a slider box to the right inlet of the random function, and test.**

**example 5: hook up a metro button between the bang and the random to get auto-testing as you change the slider and the number box.**

Now we are going to create a select box, which will take numbered input and then send a bang out the correspondingly numbered outlet. This allows you to have forking choices.

**example 6: create a select box with arguments 1 2 3; attach 3 message boxes (1 2 3) to a number box, and attach the number box to the left inlet of the select box. Attach message/print pairs to the left two outlets of the select box. Attach a number box to the remaining outlet of the select box. test.**

Note how useful the number boxes are for telling you the state of your program at various points in the information flow.

**example 7: delete the 1 2 3 boxes, leaving just the number box attached to the inlet of the select box. test by typing numbers in the number box.**

## Using Randomness in Max/MSP/Jitter

The yellow triangle tells you when a number box is ready to receive input through typing or dragging.

Now let's put the different pieces together-- randomness plus selection.

### **In-class Exercise 8**

**Hook together the following elements, in this order:**

- stop and bang buttons**
- to metro object**
- to random object**
- to select object**
- to multiple bang/message box/print objects.**

### **Variations**

- **What happens if you link in a second random subtree?**
- **What happens if you set the random argument to a much higher number than the select argument?**
- **What happens if you link in a second metro subtree?**
- **Try using one of these alternate objects instead of random:**
  - decide:** Choose randomly between on and off (1 and 0)
  - urn:** Generate random numbers without duplicates