Loops: programming structure for repeating things

Use Table[], Map[], or Do[] to carry out repeated tasks

Table[  \textit{lines to be repeated}  , \{\textit{iterator}\}]

where the lines to be repeated consist of other Mathematica functions or lists of functions separated by semicolons

\textit{iterator} is a special construction that creates a temporary counting variable and specifies number of times to repeat

Simple: \{10\}  (repeats 10 times)
With variable: \{x,10\}  (repeats while incrementing x from 1 to 10 in steps of 1)
Full: \{x,1,10,1\}  (repeats while incrementing x from 1 to 10 in steps of 1)
Full: \{x,10,2,-2\}  (repeats while incrementing x backward from 10 to 2 in steps of 2)
Conditional statements

Is equal?   ==
Is unequal? !=
Greater than? >
Less than? <
And &&
Or ||

If[ statement is true, then this, or else this ]

myage = 65.5;
If[ myage > 50, Print[“my age is older”], Print[“my age is not older”]

If[ myage > 55 && myage < 65, Print[“my age is in the bin”], Print[“my age is outside the bin”]
Working with Strings

Strings are entities of characters, as opposed to numbers. You can manipulate strings in Mathematica as well as numbers. For example:

mytext = "Species";

You can combine strings by joining them with the StringJoin[] function or <> (which do the same thing):

In[16]:= StringJoin[mytext, " Name"]
Out[16]= Species Name

In[17]:= mytext <> " Name"
Out[17]= Species Name

You can create a list of labels using Table[] and ToString[], the latter of which converts numbers to strings so they can be joined to other strings:

In[18]:= Table[mytext <> " " <> ToString[x], {x, 5}]
Out[18]= {Species 1, Species 2, Species 3, Species 4, Species 5}
Random numbers
Mathematica has many functions for generating random numbers.

(* Random real number from 0 to 1 *)

In[4]:= RandomReal[]
Out[4]= 0.9513

(* Random real number from 100 to 1000 *)

In[5]:= RandomReal[{100, 1000}]
Out[5]= 505.785

(* 10 random real numbers from 100 to 1000 *)

In[10]:= RandomReal[{100, 1000}, 10]

(* Random number drawn from a normal distribution with a mean of 10 and standard deviation of 100 *)

In[11]:= Random[NormalDistribution[10, 100]]

(* 10 pairs of random numbers between 0 and 1 *)

In[12]:= Table[RandomReal[{0, 1}, 2], {10}]
Out[12]= {{0.0245927, 0.630284}, {0.260035, 0.591502}, {0.38211, 0.146923}, {0.891077, 0.0315945}, {0.75184, 0.567132}, {0.553506, 0.443656}, {0.614652, 0.300159}, {0.791076, 0.0654448}, {0.19977, 0.272843}, {0.291167, 0.958036}}
Defining your own function

You can create your own customized functions to perform operations that you use a lot.

The syntax uses “:=” to define the operation of the function.

The input parameters are defined as variables with an underscore after them.

The Module function shields the variables used in the custom function from the rest of the notebook (it keeps them from clashing).

Custom functions usually end with Return, which is a function that returns something to the user in response to the input parameters.

This example takes two numbers as input, adds them together and multiplies them by 10, and stores the result in the temporary internal variable \( j \). The value is returned to the user at the end of the function.