Chapter 20 -- Finding your way on your own with on-line documentation

Section 20.1 Using on-line documentation

The appeal of systems for technical work such as Maple is that once you know the basics, you can rely on an extensive built-in libraries of procedures. However, the size and volume of the libraries easily exceeds what can be taught in a course or read about in a tutorial. Using a system to get work done will inevitably lead to the need to learn use additional features on your own that you did not originally learn or were taught in a course.

Fortunately, most systems have extensive explanations.

This course has introduced many of the key concepts needed to understand the documentation at the level it is written. The advice about learning new features is:

a) Become familiar with the general nature of features.
b) Become adept at looking things up.
c) Look at the initial explanation and the examples for the fastest route to learning how to invoke a feature.
d) Experiment with the examples in the documentation to see if you can reproduce the effects described. Once you have that figured out, proceed to modifying the examples so that they are doing what you are seeking.

Section 20.2 Case study -- learning another way of plotting

Looking at the help pages for commands we have already been introduced to sometimes leads to the discovery that there are alternative ways of invoking the commands that do something similar. These extensions are usually straightforward to understand because of the prior familiarity with other forms of the command.

Looking at the plot command, we discover that there is a form where it will plot a smooth curve using two lists as arguments.

**Figure 20.2.1 Plot command help**


We scroll to the bottom of the page and find an example of this. We are looking for a version of plots where v1 and v2 are lists. We don't see something exactly like that but we do see something with Vectors which are similar.
Evidently, the first list is the values of the $x$ (horizontal) coordinates, and the second list the values of the $y$ (vertical) coordinate.

We copy and paste the example into a Maple worksheet and then see if we can get it to work.

**Figure 20.1.3  Copying the example**
According to the documentation, we should be able to get this to work if the first two arguments are lists or vectors. So we edit the example to do lists instead and re-execute the line to see if it works in the same way.

Figure 20.1.3 Modifying the example
The other thing that we want is to have a line drawing, not just points. Looking at the example, we decide to experiment with removing the "style=point" option and seeing what we get. Experimentation in an empty worksheet would probably be a lot faster than looking up things up to verify that this is the right thing to do.

**Figure 20.1.4 Experimental modification**
Section 20.2 Case study -- plot options

In previous labs, we have explored a number of optional arguments to the `plot` function. Suppose we know (because we’ve seen an example) that `plot` can add captions and axis labels. We also are interested in ways of specifying textually particular ranges for the horizontal and vertical axes, having seen this feature at work in the 2d plot wizard (Section 2.5.3).

To do this we activate Maple help and type in "plot". We see a list of topics, and select "plot options".
The page for plot options is quite extensive. We browse through the list and find that the way of getting a caption is through "title". We skip to the end of the page and find an example of this. We copy and paste it into a blank worksheet and see whether we can reproduce what the example says should happen.

Once we succeed, we try to modify the example so that it does something a bit closer to what we want.

In a similar fashion, we find that "view=" is the way to specify limits on the horizontal and vertical- axes.

Section 20.3  Another case study -- sprintf

We have a need to create a plot caption that contains a value of a parameter used to generate the plot. However, we expect to vary this parameter in a script, so we would like to generate the caption dynamically while the script is running.

What this means is that we have to have some way of placing the parameter value in the string we create for the title in the plot invocation. We enter "string" into Maple help and see a list of topics.

We recall that "printf" is a way of taking values and printing them out in the context specified by a string. We take a chance and look up "sprintf".

This is what we want. We invoke an example to see whether we can get the same thing to happen.

Then we create a sprintf statement that tries to create a string that has a parameter value in it.

Finally, we embed this in a plot to see if the string has the desired effect in the title.

Section 20.Z  Chapter Summary

How to acquire information about features from reading documentation

Use topics search to find candidate pages to look at.

Look at documentation for procedures you already know about, to see what additional features they have.

Once you find a feature, look at the examples. Start off by copying an example given in the documentation exactly as is and getting it to work as described. Then modify it to start doing more closer to your final goal. Learn through experimentation under guidance of the examples.
and the explanation provided.