CHAPTER 1

Getting Started

1.1 Starting Mathematica

When we start Mathematica a fresh window, or notebook, will open. This is where we will do all of our mathematical calculations and graphics. At the top of the window we see the title of the window, which initially is “untitled-1.” Later, when we see how to save our work as a file, we can give a name to the file, and that name will appear in the title bar of the window.

1.2 Entering Expressions

Let’s do our first calculation! If we type 1+1 and then press Shift+Return (i.e., hold down the Shift key and then the Return key)

1 Mathematica computes the sum and places the answer on the next line in the window. This is called evaluating or entering the expression. The window now contains

1On Mac keyboards, Shift+Return is the same as Enter. With either Windows or Mac OS, using the Return key will simply move the cursor to the next line, allowing us to type more.
Example 1.2.1

\texttt{In[1]= 1 + 1}
\texttt{Out[1]= 2}

Notice that \textit{Mathematica} has placed “\texttt{In[1]:=}” and “\texttt{Out[1]=}” labels to the left of 1+1 and 2, respectively. To the right of the input and output, \textit{Mathematica} has placed a set of \textit{brackets}. The two innermost brackets enclose the input and output, respectively, and the larger bracket groups the input and output together. Each bracket contains what is known as a \textit{cell}. All of the calculations that we do in this notebook will be organized into cells and the brackets that surround the cells will come in handy for organizing our work. We'll have a lot more to say about this in Chap. 11, so don’t worry too much about the brackets now. In fact, \textit{until we get to Chap. 11 we will be omitting the brackets most of the time when we display Mathematica input and output.}

1.3 Editing Cells

Let’s change 1+1 to 1+2. \textit{Mathematica} supports all the usual mouse-driven text-editing features of word processors. We can simply use the mouse to place the cursor in the input cell and edit the entry so that it reads 1+2. To redo the calculation, we now reenter the cell by once again pressing Shift+Return. The result is

Example 1.3.1

\texttt{In[2]= 1 + 2}
\texttt{Out[2]= 3}

Notice that the the \texttt{In} and \texttt{Out} labels have changed to “\texttt{In[2]:=}” and “\texttt{Out[2]=}”. Each time we reevaluate a cell, the numbers in the \texttt{In} and \texttt{Out} labels will change.

To create a new cell with a new calculation, simply start typing. \textit{Mathematica} will place the input in a new cell. When many cells are present we can use the mouse to place the cursor between existing cells and click the mouse button to insert a new cell at that location. Notice how the cursor changes from a vertical bar when located inside a cell, to a horizontal bar when located between cells. With the cursor between cells, click the mouse button and then start typing. \textit{Mathematica} will create a new cell at the desired location.
CHAPTER 1  Getting Started

Finally, we can also click on the bracket which encloses a cell to select it. After selecting a cell we can reevaluate it by pressing Shift-Return or treat it just like any selected item in a text document and cut, copy, or paste as usual. Try deleting an entire cell by clicking on its bracket and then choosing Edit \> Cut from the menu bar, (or using the equivalent keyboard shortcut).

There are lots of ways that we can change the appearance of cells, changing the font, fontsize, color, and the like. We’ll explore these topics in Chap. 11.

1.4 Basic Arithmetic

*Mathematica* can do all the basic operations of addition, subtraction, multiplication, division, and exponentiation (raising one number to another) which are denoted by the symbols \(+\), \(-\), \(*\), \(/\), and \(^\).\(^2\) We can also use parenthesis for grouping as usual. Here is an example involving the arithmetic operations.

<table>
<thead>
<tr>
<th>Example 1.4.1</th>
</tr>
</thead>
<tbody>
<tr>
<td>\text{In}[3]= 2*3 + 4^2</td>
</tr>
<tr>
<td>\text{Out}[3]= 22</td>
</tr>
</tbody>
</table>

Here the exponentiation was done first, giving \(2 \times 3 + 16\), then the multiplication, which leads to \(6+16\), and finally the addition. *Mathematica* follows the standard order of operations, first performing all exponentiation (from left to right), then all multiplications and divisions (again from left to right), and finally, all additions and subtractions (from left to right). If we want to override these conventions we need to use parenthesis to group terms.

One nice feature of *Mathematica* is that of *implied multiplication*. We do not need to use the multiplication sign \(*\) in order to multiply. Instead, a blank space between things that can be multiplied (numbers, variables, expressions) will be treated as multiplication. The blank space can even be omitted if parenthesis are used to indicate multiplication. If we do leave a blank space for multiplication, sometimes *Mathematica* will fill in the space with the multiplication symbol \(\times\).

Basically, we can type calculations pretty much the way we would write them. Here are several examples, all contained in a single input cell.

\(^2\)When computers were first introduced, exponentiation was denoted by the "up-arrow" \(^\dagger\). The shaft of the arrow was eventually lost and we were left with only the arrowhead.
Example 1.4.2

\[ \begin{align*}
\text{In}[4] & = 5 \times 6 \\
& = 2 \times (3 + 4) \\
& = (2 - 3 + 1) \times (1 + 2/3) - 5^5 (-1) \\
& = 6 ! \\
\text{Out}[4] & = 30 \\
\text{Out}[5] & = 14 \\
\text{Out}[6] & = \frac{1}{5} \\
\text{Out}[7] & = 720
\end{align*} \]

Here we entered four separate calculations in a single input cell. (This is when you use the Return key—to type a new line in the input cell.) Notice that each result is placed in its own output cell. We didn’t use the multiplication sign for \( 5 \times 6 \) in the first calculation and instead left a blank space. After entering the blank space and the 6, Mathematica inserted the \( \times \). In the second and third calculation, because of the parenthesis, there is no confusion caused by leaving out the multiplication sign, so it is easier not to use it. The fourth calculation illustrates the factorial symbol \( ! \). We read \( 6 ! \) as “six factorial” rather than shouting SIX. By definition, \( n ! \) is the product of all integers from 1 to \( n \). Thus \( 6 ! = 6 \times 5 \times 4 \times 3 \times 2 \times 1 = 720 \).

1.5 Using Previous Results

Quite often we will perform a calculation and then want to use the output of this calculation for our next calculation. We can use the percent symbol %, to refer to the output of the previous cell. Here is an example.

Example 1.5.1

\[ \begin{align*}
\text{In}[11] & = 2^5 \\
\text{Out}[11] & = 32 \\
\text{In}[12] & = \% + 100 \\
\text{Out}[12] & = 132
\end{align*} \]

Notice that the first cell gave output of 32 and that the next cell added 100 to this to give 132. In this case the % symbol referred to the previous output. We
can even use \% to refer to the result before the last result, or even \%\% for the result before that. Sometimes using the \% symbol can be quite handy. However, it is important to remember that \% always refers to the last output. This can sometimes lead to unexpected results! In this book, we will rarely use the \% symbol.

### 1.6 Exact versus Approximate

One of the truly amazing features of Mathematica is that it will work things out exactly whenever possible. Sometimes this is just what we need, but sometimes it would be nicer to get an approximate answer. Consider the following example.

**Example 1.6.1**

\[
\text{In[1]:= } 3^{20} / 2^{21}
\]

\[
\text{Out[1]= } 3486784401 / 2097152
\]

It's pretty hard to get a feel for the fraction $\frac{3486784401}{2097152}$ and it might be nicer to approximate it with a decimal representation. We can force Mathematica to do this in two important ways. The first is to use decimal representations from the very beginning. If we replace $3^{20}$ with $3.0^{20}$ (or $3^{20.0}$, or even $3.0^{20.0}$) look what happens.

**Example 1.6.2**

\[
\text{In[1]:= } 3.0^{20} / 2^{21}
\]

\[
\text{Out[1]= } 1662.63
\]

Mathematica always views decimal representations as approximations. Thus Mathematica considers 3.0 to be an approximate number rather than an exact number. If we ever do a calculation that involves approximate numbers, Mathematica will give an approximate answer. On the other hand, if we use exact numbers in the input, Mathematica will do its best to provide exact numbers in the output. Here are several more examples that illustrate this point.
Example 1.6.3

\[
\begin{align*}
\ln(12) & = 3 / 4 \\
3.0 / 4.0 \\
12^{1/2} \\
12^{0.5}
\end{align*}
\]

Notice that \(12^{1/2}\) is the square root of 12 and that this is exactly equal to \(2\sqrt{3}\). So *Mathematica* has not only given us an exact answer, it has also simplified the input. On the other hand, by replacing the exponent of 1/2 by the “approximation” of .5 we have forced *Mathematica* to give us an approximate answer in decimal form.

The second important way to force *Mathematica* to give approximate answers is to use the *numeric evaluation* function N. We describe this function in the next section.

1.7 Using Functions

*Mathematica* has thousands of built-in functions. Fortunately, we only have to know a few dozen\(^3\) of the more important ones to do lots of neat calculations. We will be introducing the most important and useful functions in this book as we go. The next example uses the square root function *Sqrt* and the numeric evaluation function N.

Example 1.7.1

\[
\begin{align*}
\ln(10) & = \text{Sqrt}[27] \\
N[\text{Sqrt}[27]]
\end{align*}
\]

\[
\begin{align*}
\text{Out}[10] & = 3 \sqrt{3} \\
\text{Out}[17] & = 5.19615
\end{align*}
\]

---

\(^3\)OK, I lied. Knowing a hundred functions would be nice. Actually, memorizing the names of most functions is not so hard. How hard can it be to remember *Cos* for cosine, *Abs* for absolute value, and *Total* for, well, total? The real work is going to be remembering the *syntax* needed to use these functions. A good strategy will be to get good at using the built-in documentation.
Here we see the use of the built-in square root function \texttt{Sqrt}. Entering \texttt{Sqrt[27]} is the same as entering 27^(1/2). In the first computation we entered the number 27 exactly, so obtained an exact answer. In the second calculation we still entered 27 exactly but \textit{Mathematica} provided an approximate answer. We forced this to happen by using the numerical evaluation function \texttt{N}. This function will convert any number into a decimal representation. Many functions in \textit{Mathematica} have \textit{optional arguments} and the numerical evaluation function \texttt{N} is one of them. By adding the optional argument \texttt{n}, \texttt{N[x, n]} will estimate \texttt{x} with \texttt{n}-digit precision. If we want to know a certain number of digits in the decimal representation of \texttt{\pi}, for example, we can use \texttt{N} as in the first line of Example 1.7.2.\footnote{You may remember that \texttt{\pi} is an \textit{irrational} number, one whose decimal representation never ends and never repeats. Amazingly, in November of 2005, Chao Lu recited the first 67890 digits in the decimal expansion from memory! Check out \url{http://www.pi-world-ranking-list.com/index.html}.} Here we use \texttt{Pi} to stand for \texttt{\pi}, the ratio of the circumference to the diameter of any circle. \textit{Mathematica} has special symbols for a number of important mathematical constants, including \texttt{E}, the base of the natural logarithm, and \texttt{I}, the imaginary number whose square is \texttt{-1}.

\begin{example}
\texttt{Example 1.7.2}
\texttt{ln[33]= N[Pi, 100]}
\texttt{N[Pi/10, 30]}
\texttt{N[Pi/10.0, 30]}
\texttt{Out[33]= 3.141592653589793238462643383279502884197169\}
\texttt{399375105820974945923078164062862089986280\}
\texttt{34825342117068}
\texttt{Out[34]= 0.314159265358979323846264338328}
\texttt{Out[35]= 0.314159}
\end{example}

The second and third lines of Example 1.7.2 illustrate an important feature of \texttt{N}. In both cases we have asked \textit{Mathematica} for 30 digits in the expansion of \texttt{\pi}/10. In the first case we get it, but in the second case we do not. This is because in the second case we have already moved to an approximation by using 10.0 instead of 10.

Here are some sample calculations involving the constants \texttt{E} and \texttt{I}. 
Example 1.7.3

\[
\text{In}[21]:= \text{Sqrt}[-16] \\
N[E, 10] \\
E^{(I \text{Pi})}
\]

\[
\text{Out}[21]= 4i \\
\text{Out}[22]= 2.718281828 \\
\text{Out}[23]= 1
\]

The last calculation is one of the more amazing identities in all of mathematics! It follows from Euler's formula\(^5\)

\[ e^{i\theta} = \cos \theta + i \sin \theta. \]

If we substitute \( \theta = \pi \), we obtain \( e^{i\pi} = -1 \).

In the last computation we tried to take the square root of \(-16\) and \textit{Mathematica} responded with the imaginary number \(4i\). \textit{Mathematica} is perfectly happy using complex numbers \(a + bi\) where \(a\) and \(b\) are real numbers and \(i\) is the imaginary number \(\sqrt{-1}\). The numbers \(a\) and \(b\) are called the real and imaginary parts of \(a + bi\), respectively. \textit{Mathematica} has several built-in functions that deal especially with complex numbers. Two of the more important ones are the functions \texttt{Re} and \texttt{Im} which return the real and imaginary parts, respectively, of a complex number. Another important function is the absolute value function \texttt{Abs} which works not only for real numbers but for complex numbers too. In the case of a complex number \(a + bi\), its absolute value is defined as \(\sqrt{a^2 + b^2}\). Example 10.3.10 gives a few more calculations.

Example 1.7.4

\[
\text{In}[24]:= (2 + 4 \text{I}) (6 - 3 \text{I}) \\
\text{Re} [2 + 4 \text{I}] \\
\text{Im} [6 - 3 \text{I}] \\
\text{Abs} [-23] \\
\text{Abs} [3 + 4 \text{I}]
\]

\[
\text{Out}[24]= 24 + 18 \text{I} \\
\text{Out}[25]= 2
\]

\(^5\)Leonhard Euler (1707–1783) was one of the greatest mathematicians of all time. He published his famous formula in 1748.
Example 1.7.4 (Continued)

Mathematica has all the common mathematical functions built-in. These include the trigonometric functions and their inverses, the hyperbolic trigonometric functions and their inverses, and the logarithm and exponential function. Mathematica also has many special, more esoteric functions too. In this book we will be primarily interested in the more common mathematical functions.

There are two very important features of all built-in Mathematica functions. First, all built-in functions in Mathematica begin with capital letters. Some, like the inverse cosine, ArcCos, may even have multiple capital letters. Second, square brackets are always used to surround the input, or arguments, of a function. So we type Abs [-12], not Abs (-12), if we want to compute the absolute value of -12. Moreover, this is the only use of square brackets in Mathematica. (Actually, the only use of single square brackets. We’ll see shortly that double square brackets, [], are used with lists.) Parentheses, ( and ), are used to group terms in algebraic expressions. One other set of delimiters that will be extremely important are the “curly braces,” { and }. These are used to delimit lists, something that we will be introducing shortly. The three sets of delimiters, [], (), and {} are used for functions, algebraic expressions, and lists, respectively, and only for these purposes. This can be hard to get used to at first, but leads to a great system.

1.8 Using Variables

We may introduce variables and give them values using the equals sign. Here are some examples.

Example 1.8.1

\begin{verbatim}
In[29]= a = 2
In[30]= b = 3
In[31]= a + b
\end{verbatim}

\texttt{Out[29]= 2}

\texttt{Out[30]= 3}

\texttt{Out[31]= 5}
Now that we have defined \( a \) to be equal to 2, it will remain equal to 2 unless we set it equal to something else, or use the \texttt{Clear} function to clear its value. This is extremely important and can sometimes lead to a great deal of frustration! If we forget that we have given a value to a certain variable and then try to use the variable later as if it had no value, we can run into unexpected results. Example 1.8.2 shows how the \texttt{Clear} function works. Remember that previous to evaluating the cell, \( a = 2 \) and \( b = 3 \).

\begin{verbatim}
Example 1.8.2
In[22]:= Clear[a]
    a + b
Clear[b]
    a + b
Out[23]= 3 + a
Out[24]= a + b
\end{verbatim}

Variables that are given values retain those values until we quit \textit{Mathematica} or use \texttt{Clear}. It is very important to remember this! Also, once we use \texttt{Clear} the variable will continue to \textit{not} have a value until we give it one. Thus, if we reenter the above cell we will not get the same output! The second time we enter it, both \( a \) and \( b \) will have been cleared and we will not get the output of \( 3 + a \). \texttt{Clear} can be used with a number of options. A useful construction is \texttt{Clear["Global\$\"]} which will clear everything!

The real power of \textit{Mathematica} is that it can manipulate abstract expressions rather than just specific numbers. So we will often use variables that are, well, variable! That is, they have not been set equal to any specific value. The following example illustrates this. We'll talk a lot more about the \texttt{Expand} function in Chap. 4, but for now you can probably guess what it does.

\begin{verbatim}
Example 1.8.3
In[28]:= Expand[(x + y)^10]
Out[28]= \( x^{10} + 10 \ x^9 \ y + 45 \ x^8 \ y^2 + 120 \ x^7 \ y^3 + 210 \ x^6 \ y^4 + 252 \ x^5 \ y^5 + 210 \ x^4 \ y^6 + 120 \ x^3 \ y^7 + 45 \ x^2 \ y^8 + 10 \ x \ y^9 + y^{10} \)
\end{verbatim}
Variables are case sensitive. Thus \( s \) and \( S \) are two different variables. Example 1.8.4 illustrates this point.

\[
\begin{align*}
\text{Example 1.8.4} \\
\text{In}[29] &= a = 2 \\
& \quad b = 3 \\
& \quad c = 4 \\
& \quad s = a + b + c \\
& \quad S \\
\text{Out}[29] &= 2 \\
\text{Out}[30] &= 3 \\
\text{Out}[31] &= 4 \\
\text{Out}[32] &= 9 \\
\text{Out}[33] &= S \\
\end{align*}
\]

Since \( s \) (for sum) is the sum of \( a \), \( b \), and \( c \), we see 9 for the fourth output line. But because \( S \) is not the same variable as \( s \), the fifth output line contains the name of the variable \( S \). This variable has no value since we have not set it equal to anything.

You can use almost anything as a variable name except that variable names cannot start with a number. Thus \( x2 \) can be used but \( 2x \) cannot. Moreover, words or letters that already have meaning in Mathematica cannot be used. For example, we cannot use \( E \) as a variable name because \( E \) is already being used by Mathematica to stand for the base of the natural logarithm. Other reserved words and letters exist too. If you try to use one, Mathematica will simply tell you that you are not allowed to. Here is an example of this.

\[
\begin{align*}
\text{Example 1.8.5} \\
\text{In}[42] &= C = 12 \\
\text{Set::wrsym} : \text{Symbol C is Protected.} >> \\
\text{Out}[42] &= 1.2 \\
\end{align*}
\]

Since \( C \) is reserved we cannot use it for a variable, and Mathematica warns us that this is the case by typing the rather cryptic “Set::wrsym: Symbol C is
protected.” Moreover, the double arrowhead, \texttt{>>}, is actually a hyperlink to the on-line documentation, or \textit{Help Files}. If we click on this link the window shown in Fig. 1.1 pops up and explains the warning. We’ll have a lot more to say about the Help Files as we go, starting a little later in this chapter.

1.9 Using Comments

After we start to do more complicated calculations, our input cells might start to have dozens of lines. When this happens, it can start to get hard to follow what is going on. Putting \textit{comments} in our input cells, especially the more complicated
illustrate using \texttt{Length}. Also, notice that by following the \texttt{Table} command with a semicolon, \textit{Mathematica} does not print out the list in an output cell.

\begin{example}
\textbf{Example 1.12.5}
\begin{verbatim}
\texttt{In[87]:=} \texttt{cubes = Table[i^3, \{i, 1, 10\}] ;}
(* Length will give the number of elements in a list *)
\texttt{Length[cubes]}
\texttt{Out[87]= 10}
\end{verbatim}
\end{example}

\section*{1.13 Palettes}

So far we have seen how to enter the square root of 12 in two different ways: as \texttt{Sqrt[12]} or as \texttt{12^(1/2)}. A third way is to enter it as $\sqrt{12}$ by using the “Basic Mathematics Input Palette.” If we select \textbf{Palettes} $\rightarrow$ \textbf{BasicMathInput} from the menu bar, a window will open from which we may then select various forms of algebraic expressions, relational symbols, and Greek letters. Figure 1.2 shows what the Palette looks like. If we click on the square root expression, which is the first expression in the second row, the square root symbol, $\sqrt{}$, will be placed into the input cell and the cursor will be placed under the square root symbol so that we can begin typing there. After typing 12 we may press Shift+Return and obtain the answer. Example 1.13.1 shows how it looks.

\begin{example}
\textbf{Example 1.13.1}
\begin{verbatim}
\texttt{In[37]= (* using the square root symbol *)}
\texttt{\sqrt{12}}
\texttt{Out[37]= 2 \sqrt{3}}
\end{verbatim}
\end{example}

The BasicMathInput Palette contains a number of templates easily recognized by the little squares that are present, some of which are black. If you click on a Palette template it will be inserted in your notebook and whatever you type next will be inserted in the template at the location of the black square. Pressing the Tab key will take you to the next square in the template. If you select some expression and then click on a template in the Palette, whatever you selected will be pasted into the template at the location of the black square. The black squares are called Selection Placeholders while the white squares are simply Placeholders. After filling in all the place holders, type Ctrl+space (the Control key and the spacebar at the same
and with a

it ways: as

recognized

click on a

next will
he Tab key
ession and
ed into the

Selection
illing in all
at the same

Figure 1.2  The Basic Math Input Palette.
time), or use the right arrow key, to move the cursor to the right of the template if you wish to keep typing.

If we want to enter $\sqrt{12} + \sqrt{27}$ we need to perform the following steps.

1. Select the square root symbol from the Palette.
2. Type 12.
3. Type Ctrl+spacebar.
4. Type +.
5. Select the square root symbol from the Palette again.
6. Type 27 and press Shift+Return.

Using the BasicMathInput Palette can make your Mathematica input cells look pretty, but this is primarily a typesetting feature. Almost anything that can be done by using the Palette can also be done without using it! Whether you end up using the Palette a lot or not is largely a matter of taste. On the other hand, really complicated expressions can be easier to “see” if they are typeset. So using the Palette to enter complicated expressions can be quite useful. A few of the templates available in the Palette can also be inserted by selecting Insert $\triangleright$ Typesetting from the menu bar. Furthermore, these have keyboard shortcuts that make using them a lot faster than selecting from the Palette. For example, to typeset $\sqrt{12} + \sqrt{27}$ using keyboard shortcuts we would type

\[
\text{Ctrl+2, 1, 2, Ctrl + spacebar}^9, \text{Ctrl+2, 2, 7}
\]

(Here the commas separate the keystrokes—don’t type the commas!)

In addition to the keyboard shortcuts available on the Insert menu, the Greek letters all have shortcuts too. Typing esc, letter, esc (esc is the Escape key) will insert the Greek equivalent of letter. Thus esc, a, esc will insert α. Typing esc, e, esc will insert ε but typing esc, ee, esc will insert Ε, the base of the natural logarithm. Similarly, typing esc, ii, esc will give the imaginary number i. (The special constant π is typeset with esc, p, esc, not esc, pp, esc. You can’t use π as a variable name. It has to represent the ratio of the circumference of a circle to its diameter.)

You’ll need to experiment to see how much use you want to make of the Palette. Certainly the keyboard shortcuts make using it more palatable! We’ll have more to say about using Palettes, including making your own custom Palettes, in Chap. 11.

---

9 You may also be able to use the right arrow key.
1.14 Saving and Printing Our Work

To save our work, we choose File » Save from the menu bar and give the name and location where the file should be saved. Mathematica files are known as notebooks and are stored with a “.nb” extension. Notice that after saving the file, the title appears in the title bar of the window where it used to say “untitled-1.” It is a good habit to save our work often just in case the computer should crash for some mysterious reason. This may be a rare occurrence, but when it does happen we don’t want to lose all our work!

It is possible to have many notebooks open at the same time and switch between them by using the Window menu. We’ll have a lot more to say about notebooks in Chap. 11 where we’ll see how to insert text, photos, and all sorts of other materials into a notebook and organize the whole work into chapters and sections just like a book. Once we have created several notebooks, or downloaded notebooks from various Web sites, we may open them by selecting File » Open from the menu bar. Most of the commands in the File or Edit menus will be familiar to anyone who has worked with other programs such as word processors. Still, some of the commands are not so obvious and will be covered in this book.

To print a notebook simply choose File » Print. Notice that it is also possible to print a single cell, or selection of cells. First click on the cell bracket to select a cell and then choose File » Print Selection.

1.15 Getting Help!

Mathematica is equipped with a huge collection of files and tutorials that explain how to use the program. There must be thousands of pages, if not tens of thousands of pages, of documentation. These files can be accessed by choosing Help » Documentation Center or Help » Virtual Book from the menu bar. We will generally refer to this reference as the “Help Files” and it is very important for the Mathematica user to learn how to navigate and use the Help Files.

We have already seen one case of using them. Namely, when we tried to use the letter C as a variable and Mathematica gave us a warning. In that case, a hyperlink appeared in our notebook and if we clicked it we were taken to a page in the Help Files which explained the problem.

Alternatively, if we select Help » Documentation Center, a window will open with lots of links for us to choose from. At the top of this window is a search field where we can type a word or phrase we want to find out about. If we type in a

---

10 The people at Wolfram encouraged me to write this book as a Mathematica notebook!
Table

Table[expr, {i, imax}]
generates a list of imax copies of expr.

Table[expr, {i, l, imax}]
generates a list of the values of expr when i runs from l to imax.

Table[expr, {i, l, imax, di}]
starts with i = l.

Table[expr, {i, l, imax, di}]
uses steps di.

Table[expr, {i, {l1, l2, ...}}]
uses the successive values l1, l2, ...

Table[expr, {i, {l1, imax}}, {j, {j1, j2, ...}}]
gives a nested list. The list associated with i is outermost.

Figure 1.3 The help page on the Table command.

Mathematica command or function there we can get the documentation for that function. For example, suppose we type Table into the search field.

Figure 1.3 shows the Help Files page about the Table command. The page begins by explaining the syntax of the Table command, that is, the different ways that we can use the command. Notice that six different forms are listed and so far we have discussed the third and fourth forms. The second form shows that we can not only omit the steps, but also the starting value of the counter. If we do this Mathematica will assume that the counter should start at 1. The first form of the command shows that we can even omit the name of the counter! This would be a fairly uncommon usage of Table, but sometimes it's just what we need. After showing the various forms of the Table command, the help page then gives a lot more information including examples and links to tutorials.

Sometimes we might remember the command we want but not quite remember how to use it. We can get a quick answer by typing a question mark followed by
the name of the command directly into a *Mathematica* cell and evaluating that cell. Example 1.15.1 shows what happens. We get a description of the syntax and a hyperlink that will take us to the documentation shown in Fig. 1.3.

**Example 1.15.1**

```mathematica
In[28]:= (* getting information on a function *)
? Table

Table[expr, {i, imin, imax}] generates a list of imax copies of expr.
Table[expr, {i, imin, imax}] generates
a list of the values of expr when i runs from 1 to imax.
Table[expr, {i, imin, imax}] starts with i = imin.
Table[expr, {i, imin, imax, di}] uses steps di.
Table[expr, {i, {i1, i2, ...}}] uses the successive values i1, i2, ...
Table[expr, {i, imin, imax, {i1, i2, ...}}]
gives a nested list. The list associated with i is outermost. »
```

Another way to reach the documentation page on a specific function, if the function name appears in our notebook, is to select the function name and then choose **Help ▶ Find Selected Function** from the menu bar.

If we can’t remember, or don’t know what function we need, select **Help ▶ Function Navigator** from the menu bar. This will bring up a catalog of functions organized in various categories. For example, under **Core Language ▶ List Manipulation ▶ Constructing Lists** we’ll find **Table** (as well as 11 other functions). The **Function Navigator** is a great place to learn about new functions. Suppose we are working with lists and need to pick out a certain element from a list. By browsing the **Function Navigator** we can see what functions are available and we might find just what we need.

Finally, the **Virtual Book** is an excellent resource. It groups together all the guides and tutorials that are in the Help Files in an organized way. The Help Files are an indispensable source of information and the **Mathematica** user needs to learn how to use this valuable resource. We’ll be offering a guide to the Help Files as we go.

**1.16 Find Out More**

In this chapter we have learned how to start *Mathematica*, type some basic commands into a notebook, and save our work. We have seen some of the syntactic features of *Mathematica* such as the fact that all built-in functions start with a capital
letter, all functions use square brackets to enclose their arguments, and so on. We
have also learned how to access the Help Files, an important source of information.
To find out more about getting started, we recommend you go through a couple
of the Mathematica tutorials that can be found in the Help Files. The following
tutorials and other information should be helpful.

- First Five Minutes with Mathematica—This is a very quick introduction that
  shows off a few functions that we’ll be seeing in later chapters. Choose Help
  Documentation Center from the menu bar and look for the link to this
  tutorial in the lower right column of the page.
- The Virtual Book—Open the Virtual Book and start perusing it! Take a look
  at the Introduction and start reading the entries under Getting Started.
- The Function Navigator—Mathematica has over 2200 built-in functions.
  Open the Function Navigator and start looking around in it. You can also
  see an alphabetical list of all functions by choosing Help Documentation
  Center and then “Index of Functions” from the lower right column of the
  page. Try clicking on one of the functions. It will take you to the Help Files
  page for that function.
- Entering Expressions—Start reading Notebooks and Documents Input
  and Output in Documents in the Virtual Book. Don’t skip this one!
- Building Up Calculations Overview—Read the Building Up Calculations
  section under Core Language in the Virtual Book.

As you browse through the documentation you will find other links that might
be helpful. Lots of the Help File pages are not going to be of interest now, but will
become useful as you learn more.

Quiz

1. Use Mathematica to compute $(\frac{1}{2} + \frac{1}{3})^3$ exactly.

2. Use Mathematica to compute $(\frac{1}{2} + \frac{1}{3})^3$ and represent the answer in decimal
   form.

3. It turns out that the numbers $e^x$ and $\pi^x$ are pretty close to each other. (Here
   $e$ is the base of the natural logarithm and $\pi$ is the ratio of the circumference
   of any circle to its diameter.) Without computing them it is not easy to decide
   which is bigger. Use Mathematica to find out which number is bigger.
Two-Dimensional Graphics

*Mathematica* can be used to draw beautiful pictures that make it easy to visualize complicated curves, surfaces, data sets, or other shapes. In this chapter we'll focus on two-dimensional graphics. In Chap. 6 we'll introduce tools to display three-dimensional objects.

### 2.1 The Plot Function

One of the most fundamental and useful graphics tools is the *Plot* function which can be used to draw the graph of a function. Here is a simple example.

```math
Example 2.1.1

In[74]:= (* using Plot to graph a function *)

    Plot[Sin[x], {x, 0, 2 Pi}]
```
The Plot function takes two arguments. The first is the function that we want to plot which, in this case, is the sine function Sin[x]. The second argument is the list \{x, 0, 2 Pi\} which tells Mathematica to graph sin(x) from x = 0 to x = 2\pi. In other words, this list indicates the domain of the function. The domain list \{x, 0, 2 Pi\} is similar to the counter list \{k, 1, 10\} that we might include as an argument to the Table function. In both cases we are naming a variable, x or k, and giving the minimum and maximum values that we want it to vary between. With the Plot function it is important that the variable we use in the function matches the one we use in the domain list. If we had entered Plot[Sin[y], \{x, 0, 2 Pi\}] it would not work because the two variables (x and y) do not match.

The first argument to the Plot does not need to be a single function. In fact it can be a list of functions, in which case Mathematica will superimpose the graphs of all the functions in the list. Let’s superimpose the graphs of sin(x) and x^2/10. The next example will do this.

Example 2.1.2

In[79]= (* plotting more than one graph *)

Plot[{Sin[x], x^2/10}, \{x, 0, 2 Pi\}]
2.2 Resizing Graphics

After using Plot we can resize the graphic using the mouse. First click anywhere in the graphic. This will display a bounding box surrounding the graphic as shown in Example 2.2.1. At the corners of the bounding box, and at the midpoints of its sides are small squares known as handles. If we drag one of the handles with the mouse the figure will change size. You have to try this yourself to see how it works!

Notice, however, that as you drag the bounding box, the shape of the figure will remain the same, that is, the ratio of the height to the width of the bounding box will remain the same. This ratio is called the aspect ratio of the figure. The aspect ratio remains constant as we drag the handles.

Example 2.2.1

In[1]:= (* click graph to see the bounding box *)

\begin{verbatim}
Plot[Sin[x], {x, 0, 2π}]
\end{verbatim}

Out[1]=

\begin{figure}
\centering
\includegraphics{example221}
\end{figure}
If instead, we hold down the Shift key as we drag one of the handles then we can change the aspect ratio. Doing this to Example 2.2.1 allows us to make the figure wider and shorter as shown in Example 2.2.2.

Example 2.2.2

In[8]:= (* shift-click the bounding box to
stretch it *)

Plot[Sin[x], {x, 0, 2 Pi}]

Out[8]=

What happens if we drag the edge of the bounding box and not one of the handles? This will introduce *margins* around the figure with a new bounding box surrounding the original bounding box. By dragging the edge of the smaller box (not one of its handles), we may drag the smaller bounding box around inside the larger one to place it anywhere inside the larger box. If we hold down the Shift key as we do this, the inner box will automatically be centered in the outer box. To get rid of margins first drag the inner box to one corner of the outer box. Next resize the outer box to be as small as the inner one. Finally, we can crop a figure by holding down the Command key as we drag one of the handles.¹ You really need to try all of this yourself. Example 2.2.3 shows the outcome after first introducing margins and then cropping the inner box.

Example 2.2.3

In[9]:= (* cropping the graph *)

Plot[Sin[x], {x, 0, 2 Pi}]

Out[9]=

¹The Command key is a Mac feature—it does not exist on Windows or Linux machines.