

# Chapter 7

## Operators & Functions

In this chapter, we will build a more complex mathematical expression:

paste\_12.eps ↵

To introduce you to the **Functions** palette, we will then modify our equation to read:

paste\_13.eps ↵

### Large Operators & Spaces

This example uses another construction we have not seen before: the integral. To create an

integral, bring up the Operators & Calculus palette and click on the integral button.

OperatorsCalculusButton.tiff ↵

This creates an integral sign with question marks for its limits.

Select the lower limit either with the mouse, or by typing  $^a_?$ . Keep in mind that, as far as selection is concerned, limits on operators are treated just the same as they are for superscripts and subscripts.

Type  $^a_?$  to replace the lower limit, then press  $\leftarrow$  twice (or equivalently, **Control-p** twice). The first press of  $\leftarrow$  selects the  $^a_?$ ; the second press selects the upper limit on the integral. To fill out the upper limit, click the  $^a_\infty$  button on the Misc. Symbols palette, followed by return to place the cursor after the integral sign. Go ahead and fill out the rest of the equation up to the  $^a_?$ .

You now need to put some spacing between the  $^a_?$  and the  $^a_\Gamma$ . Since you need a large space, you could either press the space key several times, or you could press the space key just once and modify its attributes. To go with the second (better) option, select the single space (remember that you may need to go to composition mode to do this) and bring up the Attributes inspector for that **Space** element. The **Space Inspector** should become visible:

SpaceInspector.tiff ↵

EquationBuilder uses four kinds of spaces internally. By default, the space bar creates the smallest space. This is known as a thin space. In our case, we wish to use the largest space. This is known as a quad space. Click on the quad space button at the bottom of the column, and then go ahead and fill out the rest of the equation.

One thing that you may notice is that since this is being typeset as a display equation, the  $a^{1/2}$  is a bit too tall. One way to get around this problem is to typeset the entire expression as an in-line equation (see Chapter 3). This reduces the size of the  $a^{1/2}$  to what we want, but unhappily, it also reduces the size of the integral sign.

Fortunately, the size of an **Operator** element like our integral sign is inspectable via the **Operator Inspector**. To bring up this inspector, select the integral and then bring up its Attributes inspector. The Operator Inspector should now be visible:

## OperatorInspector.tiff ↵

Click on the large summation symbol under **Size** to override the integral's default size. This will increase the size of the integral to what it would have been had the equation been typeset as a display equation. Go ahead and fill out the rest of the equation.

## Modifications

There is one remaining Element Creator palette we should discuss: the Functions palette.

FunctionsButton.tiff ↪

This palette is largely a convenience over manual entry and contains many common mathematical functions that may be of use to you. For example, if we had wished to use  $e^x$  rather than  $e^o$  to signify the exponential in our example equation, this function is available here. Note that in mathematical typesetting, functions are commonly typeset in a roman font, whereas variables are italicized. By separating out the string of characters that makes up a **String** element, EquationBuilder is able to automatically and intelligently keep track of typesetting mathematical functions in the proper font.

The text that a String element contains can be inspected using the **String Inspector**. You can bring up this inspector by selecting a string element and bringing up its Attributes inspector.

## StringInspector.tiff ↪

Here you may enter whatever string you wish into the **Contents** field.

You could have also accomplished the same thing by typing out  $e^x$ , selecting those

characters, and then switching their font type from Default to Roman using the Element Inspector. Remember, though, that EquationBuilder would then treat all the characters as different variables.

Assuming you were willing to keep track of the fonts, it should not make too big a difference. However, you should know that for special functions like  $\lim^{\circ}$  and  $\max^{\circ}$ , EquationBuilder, by default, positions superscripts and subscripts above and below, rather than to the side. For special functions like these, it is generally advantageous to create them as String element functions.

Also, remember that because String elements maintain their own font types, they are somewhat immune to local font changes, which is sometimes useful.