

# Mathematica Notebooks

**Paul Abbott**

Department of Physics  
University of Western Australia  
Nedlands WA 6907, Australia  
paul@physics.uwa.edu.au  
physics.uwa.edu.au/~paul

*The Notebook front end is the standard graphical user interface (GUI) to Mathematica. It is very powerful but, although the documentation for the front end is good, a new user can easily find the range of capabilities overwhelming. This tutorial, following on from the “Overview of Mathematica” tutorial, is designed to help you make more effective use of the Notebook front end by highlighting its most important features.*

---

## ■ 1. Introduction

*Mathematica* stores information in a kind of document called a *Notebook* which can contain text and graphics. The document you are now reading is itself a *Mathematica* Notebook. Notebooks are usually divided into sections consisting of groups of cells. When a group is closed, only the first cell in the group is displayed, and the group bracket is shown with a triangular flag at the bottom:



To open a group of cells, double-click on a closed group’s cell bracket. For more information on the structure of Notebooks have a look at Notebooks and Cells.

---

## ■ 2. Notebook Design: Defining and Using Styles

Building a Notebook is straightforward because the Notebook front end is similar in many respects to a word processor. The GUI includes a number of features to make it easy to build structured Notebooks. For example, the *automatic grouping* of cells is something that the user does not have to worry about.

Many *Mathematica* users just use the program to do mathematics. However, for teaching and research, it is helpful to document what a series of commands is doing. The preparation of a Notebook is greatly assisted by the use of *styles*. For documentation, one can use Text cells. And it makes sense to structure Notebooks by breaking it into a number of sections using Section cells. *Mathematica* is capable of producing complex documents, such as scientific papers or books, and provides precise control over the appearance of the text and mathematics using options, styles, and style sheets.

### □ 2.1. Options

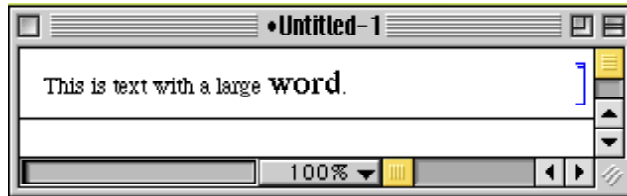
Options are used to control the appearance and behavior of everything in the front end.

---

1. Open a new Notebook. Use the menu command **Format**  $\triangleright$  **Style**  $\triangleright$  **Text** or, alternatively, use **Format**  $\triangleright$  **Show Toolbar**, and select **Text**, to create a new text cell. Type in some text.

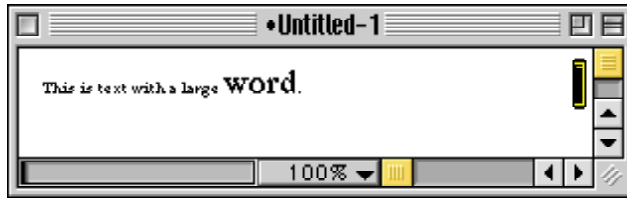
---

2. Select a single word in the text cell and change its FontSize to 18 point using the menu command **Format**  $\triangleright$  **Size**. You should get something like:




---

3. Now, selecting the cell bracket, and changing the size of the text in the entire cell to 9 point. You should get something like:



$\Delta$  Options can be set at different *levels*. In the example above, options were set at two levels: character and cell.

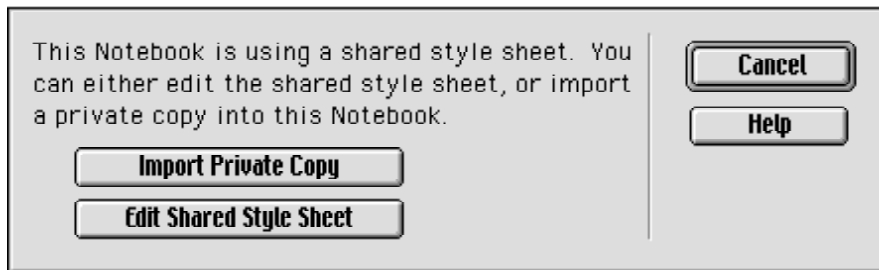
## □ 2.2. Styles and Style Sheets

To change the size of *all* text in a document it is *not* a good idea to select all text cells and change their size. Instead, one should edit the document's *style sheet*. Style sheets are actually notebook documents containing special style definition cells. Style sheets can be shared among notebooks, or a notebook can have a private collection of style definitions stored within the same file as the rest of the notebook.

To redefine the "Text" style:

---

4. Go to the **Format** menu and choose **Edit Style Sheet...**, resulting in this dialog box:



To make a global style change for our Notebook, select **Import Private Copy**.

---

Here is the section of the style sheet containing the text style:

## ■ Styles for Body Text

---

Prototype for style: "Text":  
Text

---

Prototype for style: "SmallText":  
SmallText

5. Select the innermost cell bracket for "Text" style:

---

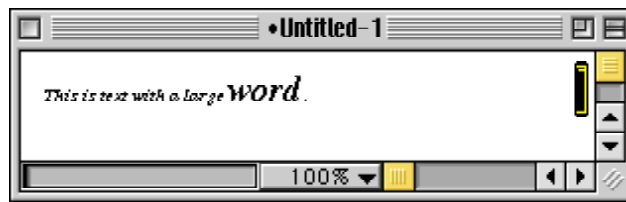
Prototype for style: "Text":  
Text

6. Go to the **Format** > **Size** menu, choose 14 point and then go to the **Format** > **Face** menu and select **Italic**. The entry in the style sheet should now look like:

---

Prototype for style: "Text":  
*Text*

and your Notebook should now look like:



△ It is clear that the global style change has made the text italic but our choice of font size does *not* override the local options.

Alternatively, selecting the innermost cell bracket for "Text" style, you can use **Format** > **Show Expression** to view the underlying Cell expression:

## ■ Styles for Body Text

```
Cell[StyleData["Text"],
  CellMargins->{{12, 10}, {7, 7}},
  Hyphenation->True,
  LineSpacing->{1, 3},
  CounterIncrements->"Text",
  FontSize->14,
  FontSlant->"Italic"]
```

This cell expression can be edited directly — and this is often the quickest way of editing a style.

## □ 2.3. Style Environments

Usually, one prepares documents for printing. However, it would be nice if you could simply transform the format of a document so that it is easy to edit (a “Working” environment) and also for the use in lecture presentations (a “Presentation” environment). Such a capability is provided through the use of *style environments*:



---

10. Select the expression on the left-hand side of the = sign and use **Cell** ▷ **Convert To** ▷ **TraditionalForm**.

---



---

11. Select the expression on the right-hand side of the = sign and use **Kernel** ▷ **Evaluation** ▷ **Evaluate In Place**.

---

### ■ Showing the steps in computing an integral

---

12. Copy the expression  $\int x^2 (x + 1)^3 dx$  and paste it twice into a new cell (say in **DisplayFormula** style), separated by an = sign.

---



---

13. Use the Algebraic Manipulation palette to do an expand in-place of  $(x + 1)^3$  on the right-hand side;

---



---

14. Copy and paste the whole right-hand side and do an expand in-place of the whole integrand;

---



---

15. Next, copy and paste the integral and do an in-place evaluation of the integral (using **Kernel** ▷ **Evaluation** ▷ **Evaluate In Place**);

---



---

16. Align on the = sign using the option inspector (**Format** ▷ **Option Inspector...**).

---

### □ 3.3 Palettes

---

17. Add definitions for vector operators by entering the following commands:

---

$$\nabla /: \nabla \times \{f_x_, f_y_, f_z_ \} := \left\{ \frac{\partial f_z}{\partial y} - \frac{\partial f_y}{\partial z}, \frac{\partial f_x}{\partial z} - \frac{\partial f_z}{\partial x}, \frac{\partial f_y}{\partial x} - \frac{\partial f_x}{\partial y} \right\}$$

$$\nabla /: \nabla \cdot \{f_x_, f_y_, f_z_ \} := \text{Simplify} \left[ \frac{\partial f_x}{\partial x} + \frac{\partial f_y}{\partial y} + \frac{\partial f_z}{\partial z} \right]$$

$$\nabla /: \nabla(f_) := \left\{ \frac{\partial f}{\partial x}, \frac{\partial f}{\partial y}, \frac{\partial f}{\partial z} \right\}$$


---

18. Use the palette,



to prove that  $\nabla \times (\nabla \times \mathbb{A}) = \nabla(\nabla \cdot \mathbb{A}) - \nabla \cdot \nabla(\mathbb{A})$  for an *arbitrary* (differentiable) vector field,  $\mathbb{A} = \{f(x, y, z), g(x, y, z), h(x, y, z)\}$ .

---

### □ 3.4 Notation

19. Define a special notation using the *Notation* package:

```
<< Utilities`Notation`;
SetOptions[Notation, WorkingForm -> TraditionalForm];
Notation[f_ : x_ -> y_ ==> f_ := Function[x_, Evaluate[y_]]]
```

Enter the mapping  $h : x \mapsto x \cos(x^2)$  in a new input cell, and evaluate  $h(y)$ .

---

## ■ 4. GridBox, StyleBox, and DisplayForm

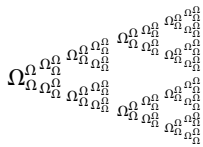
GridBox, StyleBox, and DisplayForm are powerful tools for automatically generating and formatting quite general typeset structures.

### □ 4.1 Typeset Objects

Typesetting is completely integrated with *Mathematica's* functional programming capabilities. You can use **Nest** or **NestList** to iterate a typeset box structure and then use **DisplayForm** to display the result.

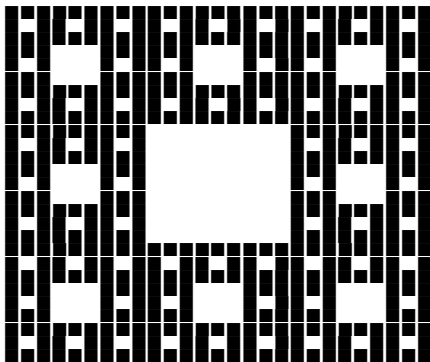
The Sierpinski triangle can be generated using a **SubsuperscriptBox** object,

```
Nest[SubsuperscriptBox[#, #, #] &, Ω, 4] // DisplayForm
```

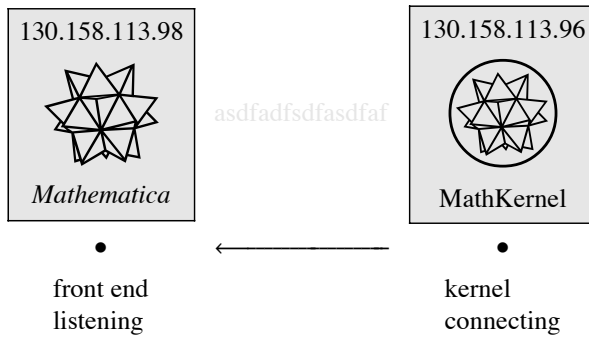


and the Sierpinski gasket by iterating a simple **GridBox** matrix.

```
StyleBox[Nest[GridBox[{{#, #, #}, {#, "", #}, {#, #, #}}] &, ■, 3], GridBoxOptions ->
{RowMinHeight -> 0, RowSpacings -> 0, ColumnSpacings -> -0.12}] // DisplayForm
```



Note that these results are *not* graphics, but are typeset objects. Here is another example of a *typeset figure*:



#### □ 4.2. Formatted Tables

To summarise a set of results, it is useful to display them in a (nicely formatted) table.

**Example:** The Euclidean Lie algebra  $e(3)$  has the Lie subalgebras  $\{R_1, R_2, R_3\}$  and  $\{P_1, P_2, P_3\}$  known as  $so(3)$  and  $e(3)$  which generate rotations and translations respectively. In terms of infinitesimal generators we have

$$R_1 = x_2 \frac{\partial}{\partial x_3} - x_3 \frac{\partial}{\partial x_2}, \quad R_2 = x_3 \frac{\partial}{\partial x_1} - x_1 \frac{\partial}{\partial x_3}, \quad R_3 = x_1 \frac{\partial}{\partial x_2} - x_2 \frac{\partial}{\partial x_1},$$

and

$$P_1 = \frac{\partial}{\partial x_1}, \quad P_2 = \frac{\partial}{\partial x_2}, \quad P_3 = \frac{\partial}{\partial x_3}.$$

*Mathematica* Implementation is straightforward.

$$R /: R_1 := x_2 \frac{\partial \#}{\partial x_3} - x_3 \frac{\partial \#}{\partial x_2} \&$$

$$R /: R_2 := x_3 \frac{\partial \#}{\partial x_1} - x_1 \frac{\partial \#}{\partial x_3} \&$$

$$R /: R_3 := x_1 \frac{\partial \#}{\partial x_2} - x_2 \frac{\partial \#}{\partial x_1} \&$$

$$P /: P_1 := \frac{\partial \#}{\partial x_1} \&$$

$$P /: P_2 := \frac{\partial \#}{\partial x_2} \&$$

$$P /: P_3 := \frac{\partial \#}{\partial x_3} \&$$

Here is the table of commutators.

`CommutatorTable({R1, R2, R3, P1, P2, P3}, {x1, x2, x3})`

	$R_1$	$R_2$	$R_3$	$P_1$	$P_2$	$P_3$
$R_1$	0	$-R_3$	$R_2$	0	$-P_3$	$P_2$
$R_2$	$R_3$	0	$-R_1$	$P_3$	0	$-P_1$
$R_3$	$-R_2$	$R_1$	0	$-P_2$	$P_1$	0
$P_1$	0	$-P_3$	$P_2$	0	0	0
$P_2$	$P_3$	0	$-P_1$	0	0	0
$P_3$	$-P_2$	$P_1$	0	0	0	0

■ **Code**

---

■ **5. Conclusion**

This course attempts to give an overview of the *Mathematica* front end (GUI). A number of Notebook formatting techniques have been presented and these can be used to advantage for courseware development and technical publishing.