

# Mathematica para a LEFT–2006

<b>O Módulo de Interface Gráfico (FrontEnd)</b>	<b>4</b>
Introdução	4
Interagindo com o FrontEnd	8
Ajuda e Atalhos de Input	10
Características gerais duma sessão de trabalho	11
Notebooks e Células	13
Tipos de Células	15
Seleção de células	16
Estilos e Opções para Células	17
Formatação de Células	20
Formatação de Texto	23
Estilos de texto	23
Ambientes de Trabalho e Impressão de Notebooks	24
Etiquetas e Numeração Automática de Células	25
Hyperlinks	25
Etiquetas	25
Numeração de Células e Listas de Items	26
Expressões Bi-Dimensionais	27
Caracteres Especiais	27
Códigos UNICODE de Caracteres Especiais	30
Teclas virtuais e Paletas	42
Utilização de Marcadores (ing. Placeholder)	42
Utilização de Paletas	43
Como criar uma tecla virtual:	45
Como criar paletas de teclas virtuais	47
Importação e exportação de Definições	47
Importação de Módulos Adicionais (Packages e Contextos)	48
Get (<<), Needs	49
Remove[symbol]	51
Contexts	51
Formatação de Expressões: Input e Output	52
Formatos de Exportação: TeX, HTML, EPS	55
Exportação de Texto Formatado	55
Exportação de dados	56
Exportação de Gráficos	56
<b>O Módulo de Cálculo (Kernel)</b>	<b>58</b>
Avaliação de Expressões	58
Primeira avaliação	59
Ciclo de Avaliação	59
Paleta – Cálculos e Funções Básicas do Mathematica	60

Uso de Parêntesis em Mathematica .....	63
<b>Computação Simbólica .....</b>	<b>68</b>
Atribuição e Substituição .....	68
Atribuição Imediata de uma expressão a um símbolo .....	69
Atribuição Retardada de uma expressão a um símbolo .....	70
Substituição Imediata de um símbolo ou modelo de expressão .....	71
Substituição Retardada de um símbolo ou modelo de expressão .....	73
Substituição Iterativa de um símbolo ou modelo de expressão .....	74
Conjuntos, Listas e Matrizes .....	74
List .....	74
Range .....	76
Table e Array .....	77
Variáveis–Modelo e Expressões Conformes a um modelo. ....	78
Estruturas Condicionais e Testes .....	81
Ciclos .....	84
<b>Funções em Mathematica .....</b>	<b>87</b>
Definição de Função .....	87
Operadores e Funções Simbólicas .....	90
Funções matemáticas comuns .....	91
Formas alternativas de escrever funções .....	93
Funções puras .....	95
Atributos e Propriedades de Funções .....	97
Argumentos facultativos e Opções em funções .....	100
Funções definidas por Módulos .....	102
<b>Gráficos, Animação e Som .....</b>	<b>104</b>
Graphics e Show .....	104
Epilog .....	107
A Opção DisplayFunction .....	108
Manuseamento de Célula Gráficas .....	109
Show[graphics] e Show[GraphicsArray] .....	109
Funções que geram Gráficos 2D .....	113
Opções de Plot .....	116
Plot3D .....	120
ParametricPlot e ParametricPlot3D .....	121
ListPlot e ListPlot3D .....	122
Animação de Gráficos .....	124
Escrever células criando um grupo fechado .....	124
Gráficos e Som .....	126
<b>Cálculo Diferencial .....</b>	<b>126</b>
Derivação Simbólica .....	126
Integração .....	130
Solução de Equações Diferenciais .....	132

Inicial. ....	132
Exemplo I ....	132
Exemplo II ....	133
Exemplo III ....	134
Posição inicial ....	135
Amortecimento ....	135
Exemplo IV ....	136

## **Exemplos de Cálculo Numérico .....140**

### **Equações algébricas, Raízes. ....140**

Solução de equações Algébricas ....	141
Zeros de Funções ....	141
Exercício ....	142
Aplicação ....	142
Notas : ....	143
Método Genérico ....	145
Métodos iterativos: convergência de ordem quadrática, cúbica, etc ....	146
Cálculo de zeros: Significado Gráfico ....	148
Exemplo I ....	149
Exemplo II ....	156
Polinómios Quadráticos ....	157
Exemplo: $x^2 + 4x + \frac{1}{9} = 0$ ....	158
Polinómio de grau 3 ....	159

### **Solução aproximada de ODEs .....162**

Método de Euler ....	162
Exemplo V ....	163
O método de Euler melhorado ....	165
Método de Runge–Kutta ....	167
Runge–Kutta Adaptativo ....	168

## **Transformações de Fourier .....169**

Integrais de Fourier ....	169
Definição ....	169
Exemplo ....	169

### **Propriedades da Transformada de Fourier .....171**

Linearidade: ....	171
Translação: ....	171
Dilatação: ....	172
Derivação: ....	173
Convolução: ....	174
Exemplo ....	174

### **Séries de Fourier .....177**

Definição ....	177
Exemplos ....	178

Ortogonalidade .....	181
<b>Transformada Rápida de Fourier</b> .....	181
Notas de Implementação .....	183

## O Módulo de Interface Gráfico (FrontEnd)

### ■ INTRODUÇÃO

O programa *Mathematica* é um ambiente gráfico de trabalho para cálculo simbólico, numérico, edição de texto e interacção com comandos nativos (i.e. definidos pelo software do *Mathematica*) e exteriores (definidos por outros programas alheios ao *Mathematica*), quer do próprio sistema operativo (OS) do computador onde é executado, quer de outros programas (C,C++,Java,...) escritos e compilados para interagir com o *Mathematica* por via de mensagens de baixo nível enviadas através de canais (ing. **sockets**) dedicados com um protocolo denominado *mathlink*. Na realidade o *Mathematica* consiste em dois programas distintos:



O **FrontEnd**, que controla e processa o ambiente de trabalho (uma ou mais janelas designadas Notebooks) e o aspecto gráfico (formatação) das janelas em que o utilizador introduz as suas expressões e instruções de programação, e para onde são igualmente devolvidos os resultados dos cálculos, quer na forma de expressões simbólicas ou numéricas, quer na de imagens e representações gráficas. Inicie o FrontEnd escrevendo **mathematica &** numa linha de **shell** em Linux, ou procure o item *Mathematica* no menu de Aplicações instaladas no sistema, ou ainda pressione duplamente um ícone de atalho ou alias no **Desktop** para o executável *Mathematica* (Mac e Windows).



O **Kernel**, que contém a biblioteca de funções nativas do Mathematica e executa avaliações de expressões, comandos de menu seleccionados pelo utilizador ou programas (sequências de expressões) contidos numa ou mais células do tipo Input, devolve o resultado dos cálculos em geral para outras células do tipo Output (ou para programas exteriores através do *mathlink*) e armazena e recupera informação sobre as definições e avaliações feitas durante toda a sessão de trabalho desde a primeira avaliação. Use a combinação **SHIFT RET** ou a tecla **ENTER** do teclado numérico para avaliar todas as expressões na célula em que se encontra o cursor ou todas as células seleccionadas. Use **CMD RET** para avaliar uma expressão seleccionada e substituí-la pelo resultado.

Ao contrário de outros tipos de programação em que é necessário compilar o código completo dum programa para obter alguma ideia se todos os seus componentes estão a funcionar como se pretende (**debugging**), a programação com o Mathematica permite ver o funcionamento de cada componente de um programa mais complexo testando-a mesmo antes de acabar de escrever o resto do programa. Isto é, toda e qualquer expressão introduzida no **FrontEnd** do **Mathematica** pode ser avaliada pelo **Kernel** individualmente e imediatamente. O Mathematica é suficientemente desenvolvido para detectar erros de sintaxe numa expressão matemática (por exemplo falta de um parênteses "}" que fecha um que se abriu antes com "{", ou faltar o denominador de uma fracção, etc.) Se a expressão que se envia para o **Kernel** tiver erros ou não fizer sentido o resultado é em geral uma mensagem explicando o tipo de erro (geralmente sintaxe). Se a expressão depender de outras definições que o utilizador tenha feito, estas devem ter sido avaliadas pelo **Kernel** pelo menos uma vez em cada sessão de trabalho: o **Kernel** mantém uma memória de todas as definições e resultados de avaliações que se passam **durante a mesma sessão de**

**trabalho** (a menos que sejam explicitamente apagadas pelo utilizador), por isso não é necessário re-avaliar todas as definições que entram numa dada expressão que se quer testar ou modificar.

Toda a interacção com o **Mathematica** se processa através de instruções escritas numa ou mais janelas (que são objectos designados de **Notebook**), cujo conteúdo pode em qualquer altura ser guardado em documentos **NomeDoc.nb** com nome **NomeDoc** arbitrário e terminação **.nb**. Estes documentos estão em ASCII, mas em geral devem apenas ser editados pelo **FrontEnd** porque num processador de texto aparecem desformatados e de difícil leitura (e facilmente corrompíveis se editados).

Durante uma sessão do Mathematica podem-se criar novos **Notebooks** com o menu **File** ▸ **New** ou abrir **Notebooks** já guardados com o menu **File** ▸ **Open...**. Podem-se ter simultaneamente vários Notebooks abertos para trabalhar na mesma sessão. Podem-se copiar conteúdos de uns para os outros e fazer avaliações de expressões em várias janelas. Note-se contudo que, a menos que se especifique explicitamente um Kernel diferente para cada Notebook, as avaliações são processadas pelo Kernel comum de vários Notebooks pela ordem em que chegam, e afectam as definições de símbolos idênticos em Notebooks diferentes. Para o Kernel não interessa de onde veio o pedido de avaliação de uma expressão (de facto interessa só na medida em que tem que saber a que janela devolver um resultado) apenas o seu número de Input (**\$Line**) e as associações que requer para os símbolos de que se compõe a expressão a avaliar. Quer isto dizer que as variáveis definidas pelo utilizador no Mathematica em geral são **Globais** durante uma sessão do Kernel para vários Notebooks.

Quando se inicia uma sessão de *Mathematica* em geral apenas se inicia o FrontEnd com um novo Notebook que aparece numa janela designada **Untitled – 1.nb**. Esta janela serve para receber todo o tipo de comandos e seus resultados de avaliação, texto, comentários, gráficos e formatações durante o que se designa de uma sessão do FrontEnd. Todo este trabalho pode ser guardado num documento com nome arbitrário e terminação **.nb** que lhe é acrescentado automaticamente. Posteriormente poderá abrir este Notebook noutra sessão e re-avaliar e/ou alterar os comandos e resultados aí registados.

A Figura 1 mostra o aspecto geral de uma sessão de trabalho onde se definiram valores para os símbolos  $\omega$ ,  $\kappa$ ,  $T$  e  $\lambda$  para serem usados posteriormente na representação gráfica 3D da função  $2 e^{-0.1 x^2} \text{Sin}[\kappa x - \omega t]$  entre os limites  $x \in [-2\lambda, 2\lambda]$  e  $t \in [-2T, 2T]$ . Por último aparece informação sobre a utilização da função nativa **Plot3D** do *Mathematica*.

Notebook.nb

File Edit Cell Format Input Kernel Find Window Help

100%
Graphics
<< Go Back

2...3...4...5...6...7...8...9...10...11...12...13...14...15...16...17...18...19

### Definições

Frequência  $\omega$  e Número de Onda  $\kappa$

```
In[1]:=  $\omega = 5; \kappa = 3;$ 
```

Período  $T$  e Comprimento de Onda  $\lambda$

```
In[2]:=  $T = \frac{2\pi}{\omega}; \lambda = \frac{2\pi}{\kappa};$ 
```

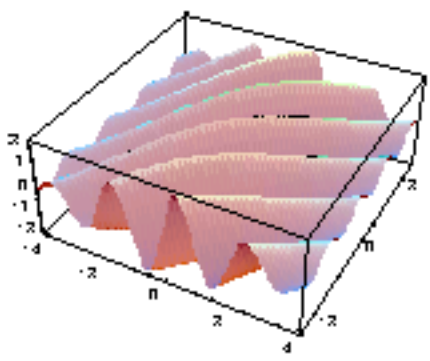
```
In[4]:=  $\{\omega, \kappa, T, \lambda\}$ 
```

```
Out[4]=  $\{5, 3, \frac{2\pi}{5}, \frac{2\pi}{3}\}$ 
```

### Gráfico

```
In[3]:= Plot3D[ $2e^{-0.1x^2} \sin[\kappa x - \omega t]$ , {x, -2 $\lambda$ , 2 $\lambda$ }, {t, -2T, 2T},
Mesh -> False, PlotPoints -> 70]
```

From In[3]:=



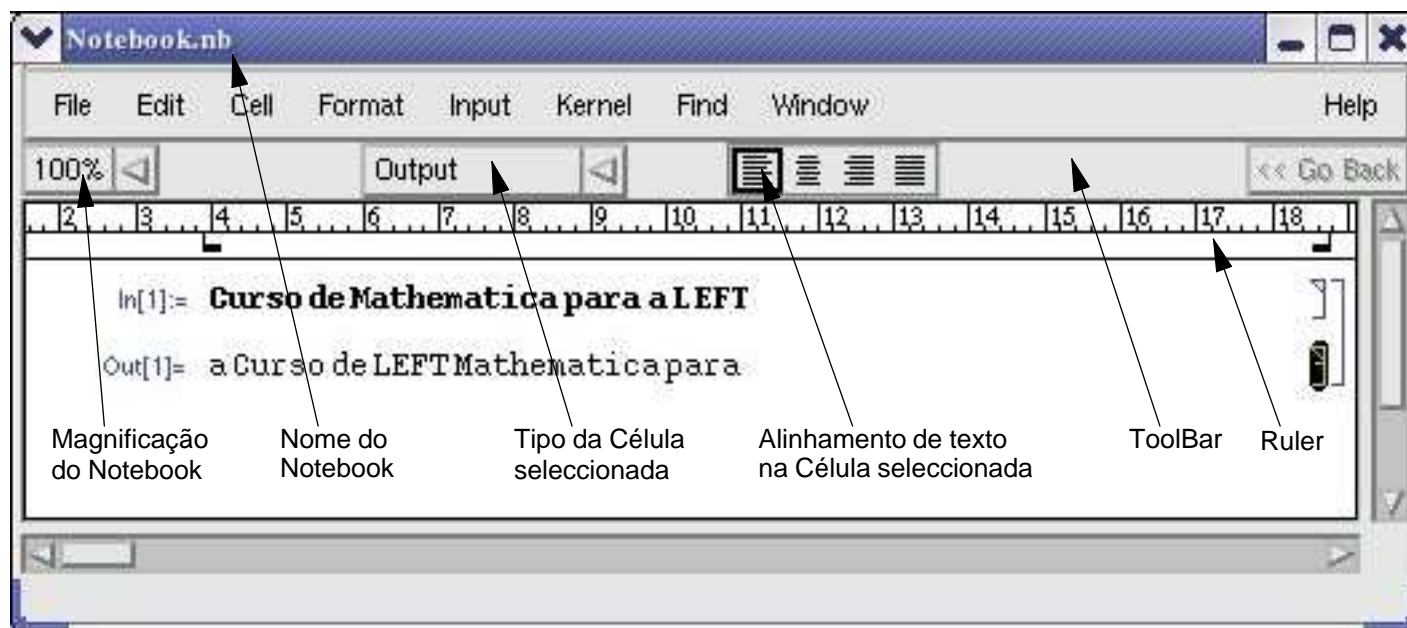
```
Out[3]= - SurfaceGraphics -
```

```
In[5]:= ? Plot3D
```

Plot3D[f, {x, xmin, xmax}, {y, ymin, ymax}] generates a three-dimensional plot of f as a function of x and y. Plot3D[{f, s}, {x, xmin, xmax}, {y, ymin, ymax}] generates a three-dimensional plot in which the height of the surface is specified by f, and the shading is specified by s. [More...](#)

Figura 1

Os elementos que compõem a janela dum Notebook estão exemplificados na Figura 2. Para além da familiar barra de menus, é possível mostrar/esconder uma barra de edição (ing. **ToolBar**) e uma régua (ing. **Ruler**) que mostra os limites da página de impressão e do parágrafo seleccionado.



**Figura 2**

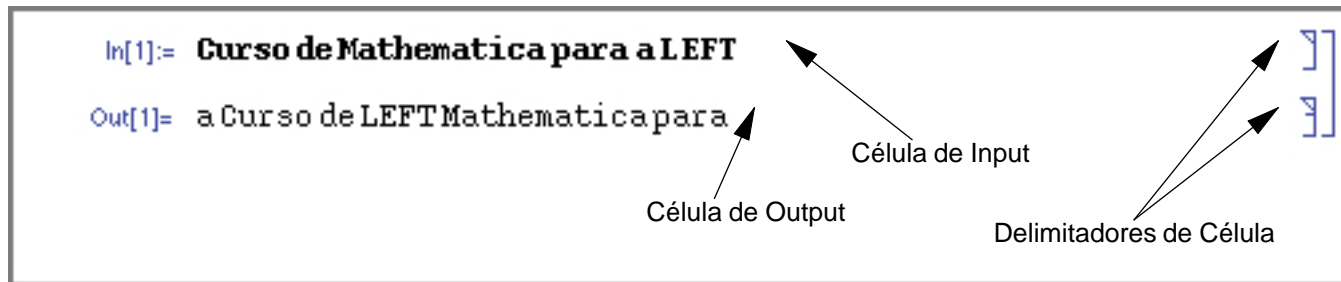
#### □ INTERAGINDO COM O FRONTEND

Embora seja muito parecido com um processador de texto, o FrontEnd trata as expressões introduzidas num Notebook agrupando-as por Células. Um exemplo é um parágrafo de texto como este, que por si só é uma Célula do tipo "**SmallText**", contendo eventualmente outras Células de tipos diferentes como por exemplo este  $x^2$  que é uma célula do tipo "**InlineCell**" com um fundo cinzento.

As Células podem ser entendidas como parágrafos generalizados com propriedades individualizadas. As células que se usam mais frequentemente no *Mathematica* são as de "**Input**" e as de "**Output**":

- (a) – As células de tipo **Input** são as que por norma o **FrontEnd** cria quando se escreve algo num **Notebook**, porque são células com a propriedade de serem avaliáveis e editáveis (ing. **Evaluatable** e **Editable** são opções das células, alteráveis estipulando opção → **True** ou opção → **False**);
- (b) – As células de tipo **Output** são automaticamente geradas pelo **FrontEnd** no **Notebook** para escrever os resultados dos cálculos realizados pelo Mathematica, e não são nem avaliáveis nem editáveis (**Evaluatable** → **False**, **Editable** → **False**), podendo no entanto ser apagadas, copiadas e reproduzidas num formato editável, inclusivamente como células de "**Input**".

Eis um exemplo, em que a primeira é uma célula de "**Input**" e a segunda de "**Output**".



**Figura 3**

(Note que o **Output** gerado não é igual ao **Input** porque os espaços entre símbolos são entendidos como um sinal de produto e o Mathematica põe automaticamente os termos dum produto por ordem alfabética. Para tratar o **Input** como uma expressão textual (ing. **String**) deve-se incluí-la entre '"').

Do lado esquerdo destas células pode-se ler **In[1]:=** e **Out[1]=** o que significa que o Kernel avaliou e memorizou o **Input** e **Output** correspondentes à primeira expressão submetida ao Kernel na sessão (incrementando o contador de expressões avaliadas de **\$Line = 1** para **\$Line = 2**). Durante uma mesma sessão do Kernel os símbolos **In[n]** e **Out[n]** (onde  $1 \leq n \leq \$Line - 1$ ) podem ser invocados como a n-ésima expressão a ser avaliada ou devolvida pelo Kernel, e podem ser usados como outro qualquer símbolo na composição de expressões.

O resultado **Out[\$Line - 1]** da última avaliação enviada ao Kernel é muitas vezes invocado para como parte de um cálculo, por isso usa-se frequentemente a seguinte notação abreviada em qualquer parte duma expressão de **Input**:

- (a) – O símbolo isolado **%** serve para designar **Out[\$Line - 1]**, i.e. o último resultado.
- (b) – O símbolo **%%** invocará **Out[\$Line - 2]** e  $\frac{%% \dots \%}{n}$  equivale a **Out[\$Line - n]**.
- (c) – O símbolo **%1** representa **Out[1]** e **%n** é o n-ésimo resultado da sessão, i.e. **Out[n]**.

A introdução de expressões em células de **Input** tem regras de sintaxe relativamente simples e muito mais flexíveis que noutras linguagens compiladas: a maior parte das funções nativas do Mathematica e as próprias funções definidas pelo utilizador podem ter um número variável de argumentos correspondendo a opções próprias das funções que têm valores padrão pré-definidos caso não sejam explicitamente usadas.

#### □ AJUDA E ATALHOS DE INPUT

Para facilitar a introdução de nomes de funções reconhecidas pelo Mathematica numa sessão (incluindo as que foram definidas e avaliadas pelo utilizador) existe a combinação de teclas **ALT** **[K]** que, pressionadas após a introdução parcial de um nome numa célula de **Input**, mostra um menu de possíveis terminações para a expressão que se começou a escrever, podendo-se assim escolher o nome correcto da função pretendida. (Na Figura 4 escrevendo **Plo** **ALT** **[K]** permite escolher **Plot3D** como o nome pretendido).



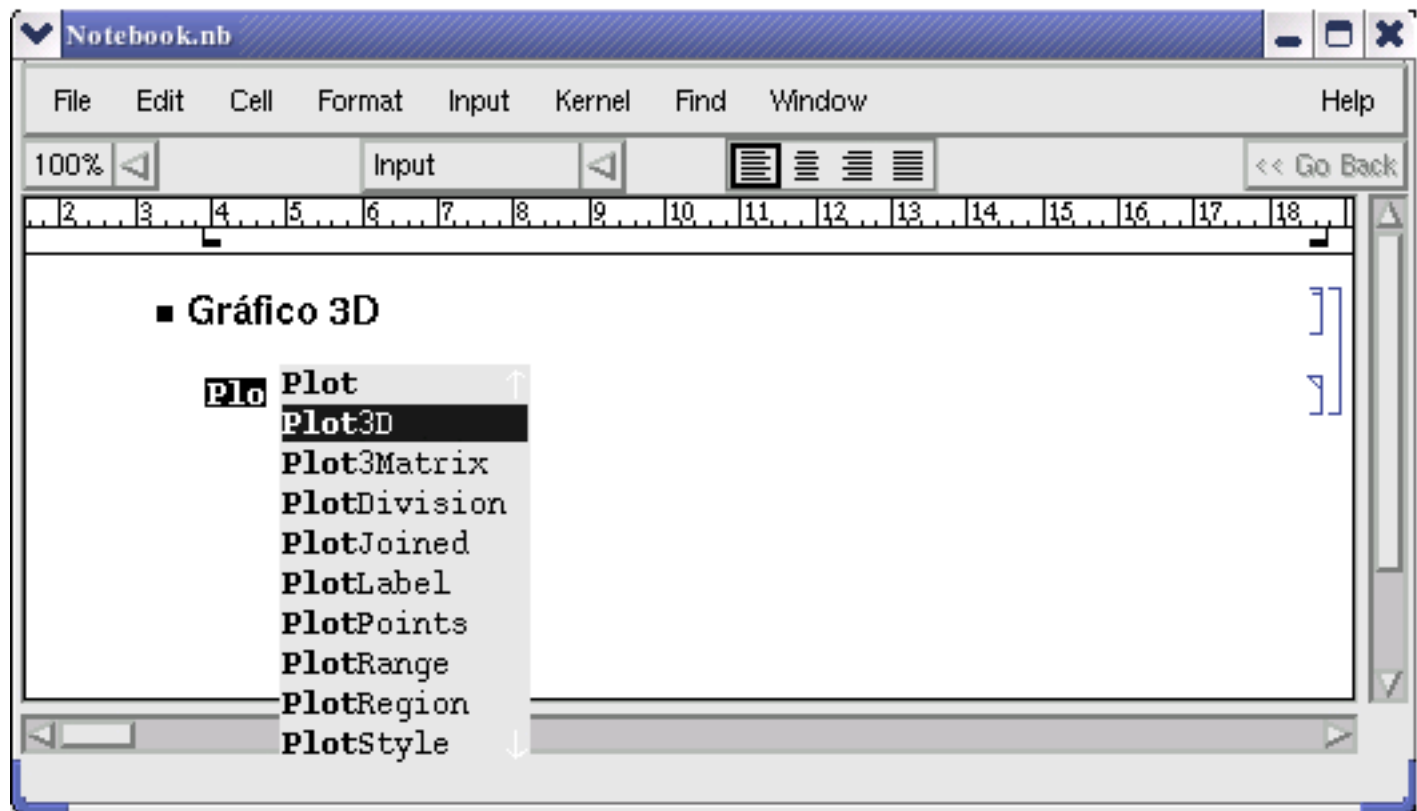


Figura 4

Adicionalmente, após escrever o nome completo de uma função nativa do Mathematica a combinação de teclas `ALT|SHIFT|k` adiciona-lhe um modelo de utilização incluindo o número e tipo de argumentos necessários para a sua boa utilização. Por exemplo, escrevendo **Plot**`ALT|SHIFT|k` fica-se com **Plot**`[f, {x, xmin, xmax}]` indicando que se deve especificar uma expressão **f** que dependa da variável **x**, a qual irá variar no intervalo  $x \in [xmin, xmax]$ .

Finalmente, para saber que opções estão disponíveis para a uma dada função basta escrever e seleccionar o seu nome e usar a combinação `ALT|SHIFT|f` para abrir a janela de Ajuda (**Help Browser**) na página que contém toda a informação sobre a função seleccionada. Alternativamente pode-se avaliar uma célula de **Input** começando com o sinal de interrogação **?** seguido do nome completo da função nativa (por ex. **?Plot**) e pressionar sobre o hyperlink **More...** que aparece no fim do texto do output resultante.

## ?Plot

`Plot[f, {x, xmin, xmax}]` generates a plot of `f` as a function of `x` from `xmin` to `xmax`. `Plot[{f1, f2, ... }, {x, xmin, xmax}]` plots several functions `fi`. More...

O menu **Help** ▸ **Help Browser...** abre em qualquer altura a janela de Ajuda que, para além de uma cópia completa do manual **The Mathematica Book**, contém informação sobre utilização, opções e exemplos de:

- (a) – **Built – In Functions**, ou seja todas as funções nativas do *Mathematica*, quer para fazer cálculos e gráficos quer para manipulação do FrontEnd.
- (b) – **Add – ons & Links** com todos os módulos (ing. **Packages**) de funções adicionais que se podem importar durante uma sessão, bem como os protocolos **MathLink**, **JLink**, **DatabaseLink** e **.NETLink** de comunicação com outros programas exteriores ao *Mathematica*.

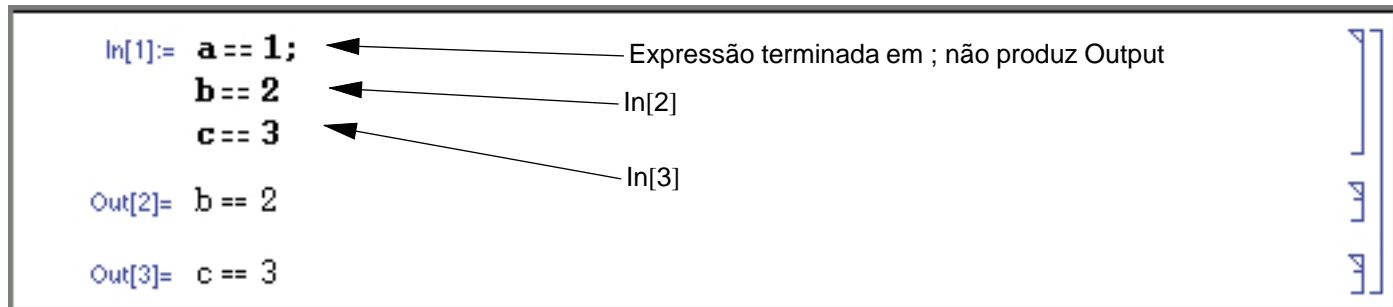
- (c) – **Front End**, incluindo a funcionalidade de todos os comandos de menus e a operação dos **StyleSheets**, opções dos Notebooks e combinações de teclas (ing. **Keyboard Shortcuts**) nos diversos sistemas operativos MS Windows, Macintosh e X.
- (d) – **Demos**, incluindo Notebooks de tutoria e Paletas adicionais que se podem importar e usar, como por exemplo uma Tabela Periódica dos Elementos ou uma paleta de Constantes Físicas. Inclui ainda o manual de funcionamento da paleta **Notations** que é de particular utilidade na adaptação da notação matemática ao gosto do utilizador sem comprometer a funcionalidade de cálculo simbólico do *Mathematica*.

Além destas categorias existe ainda um **Master Index** que inclui um índice alfabético de nomes usados no *Mathematica*, indicando onde se podem encontrar referências a um dado nome em toda a documentação disponível online, e um tutorial **Getting Started** que dá uma breve introdução ao funcionamento do Mathematica bem como informação específica sobre a instalação e resolução de alguns problemas de funcionamento nos três sistemas Windows, Mac e Unix.

#### □ CARACTERÍSTICAS GERAIS DUMA SESSÃO DE TRABALHO

- (a) – Uma sessão de trabalho do Kernel acaba quando se sai do *Mathematica*, i.e. terminando o FrontEnd e o Kernel, através do menu **File** › **Quit**, ou quando se termina só o Kernel com o menu **Kernel** › **Quit Kernel** › **Local**, mantendo o FrontEnd activo para editar e salvar Notebooks ou eventualmente iniciar nova sessão do Kernel com o menu **Kernel** › **Start Kernel** › **Local**.
- (b) – O Kernel comunica com o FrontEnd através do protocolo **mathlink**, por isso o Kernel e o FrontEnd podem estar em computadores diferentes.
- (c) – O mesmo Kernel pode responder a comandos de várias janelas numa mesma sessão do FrontEnd mas não pode em geral simultaneamente responder a sessões diferentes do FrontEnd.
- (d) – Em contrapartida, um FrontEnd pode simultaneamente ligar-se a múltiplos Kernels locais e remotos.
- (e) – Uma mesma célula de **Input** pode ser re-escrita e/ou avaliada várias vezes durante uma sessão. O Kernel armazena cada avaliação como uma expressão diferente associada ao símbolo In[\$Line].
- (f) – A célula de **Output** que se segue à de **Input** que foi re-avaliada é automaticamente apagada e re-escrita pelo FrontEnd com o resultado que lhe corresponde.
- ⌘ – Se se acrescentar um ponto-e-vírgula isolado (;) no fim de uma expressão avaliável numa célula de **Input** o FrontEnd não produz uma célula de **Output** associada à avaliação, e apaga qualquer **Output** de avaliações prévias da célula de **Input** em questão. Contudo, comandos como **Print** e **Plot** continuam a gerar as respectivas células.
- (h) – Numa célula de **Input** podem existir várias expressões que são avaliadas individualmente pelo Kernel se separadas por um ponto-e-vírgula (;) ou por um fim-de-linha (↵), bastando para isso que sejam sintaticamente correctas e completas. Se um fim-de-linha ocorrer a meio de uma expressão, as linhas seguintes são processadas até que a expressão seja válida. Isto permite partir arbitrariamente uma expressão complexa em várias linhas.

Se houver várias expressões avaliadas individualmente numa só célula de **Input** o Kernel atribui-lhes diferentes números **\$Line**, e vai enviando os respectivos resultados para o FrontEnd que os imprime (ou não se a expressão terminar em ;) como células de **Output** separadas.



**Figura 5**

- (j) – Uma avaliação em curso pode ser abortada com a combinação **ALT** **]** ou o menu **Kernel** **› Abort Evaluation** ou meramente interrompida com a combinação **ALT** **]** ou através do menu **Kernel** **› Interrupt Evaluation...**. Neste estado interrompido o Kernel pode ser usado para avaliar expressões ou modificar valores de variáveis envolvidas na avaliação que estava em curso. De seguida pode optar por retomar a avaliação ou abortá-la.

## ■ NOTEBOOKS E CÉLULAS

A introdução de dados num **Notebook** é SEMPRE feita dentro dum objecto designado **Célula** (ing. **Cell**). Se o cursor não estiver já dentro de uma **Célula**, o **FrontEnd** encarrega-se de criar uma para receber o que escrevermos. Há células de diferentes tipos e com diferentes propriedades, mas aparte o estilo, interessa saber que:

- (a) – Há células avaliáveis (**Input**, **Button**,...) e células não avaliáveis (**Text**, **Output**,...) , i.e. cujo conteúdo não é enviado para o Kernel.
- (b) – Das células avaliáveis há as que são activas (**Button**, **Sound**, ...) i.e. alguma acção é tomada quando o cursor é pressionado sobre elas, ou seja a avaliação é iniciada pelo cursor.
- (c) – Num Notebook as células podem ser agrupadas, algumas automaticamente, como as de **Input** e o respectivo **Output**, e outras manualmente, se forem consecutivas, seleccionando-as e agrupando-as com **Cell** **› Cell Grouping** **› Group Cells**.
- (d) – Os grupos de células podem ser fechados ou abertos para facilitar a leitura de Notebooks extensos com **Cell** **› Cell Grouping** **› Open / Close Group** ou fazendo a selecção do grupo e pressionando duplamente no seu delimitador direito **]**.
- (e) – Uma célula pode ser tornada invisível (embora permaneça no Notebook e o seu delimitador não desapareça) alterando a sua propriedade **Open**  $\rightarrow$  True/False com **Cell** **› Cell Properties** **› Cell Open**.

Em geral as células do *Mathematica* podem ser modificadas e editadas, com excepção

- (f) – notável das células de **Output** ou **Graphics**, embora estas últimas possam ser aumentadas ou diminuídas. Contudo qualquer célula pode ser tornada

alterável/inalterável mudando a propriedade **Cell** › **Cell Properties** › **Cell Editable** de forma a que a célula explicite a opção **Editable** → **True/False**.

Os Notebooks podem ser guardados com todas as células de **Input** e **Output** e texto, gráficos etc. que são geradas durante uma sessão. Ao iniciar uma nova sessão do FrontEnd e do Kernel pode-se recuperar as definições e expressões associadas a símbolos desenvolvidas em sessões anteriores e guardadas num Notebook **nbk\_1.nb** de várias formas:

- (a) – A mais directa é abrir **nbk\_1.nb** a partir do menu **File** › **Open...**, seleccionar aí as células de **Input** que se querem avaliar (usando se necessário a tecla de **CTRL** com o apontador para seleccionar várias células não adjacentes) e executar com o comando do menu **Kernel** › **Evaluation** › **Evaluate Cells**. Isto permite avaliar apenas as células que se pretendem e pela ordem que se deseja.
- (b) – Também é possível recuperar as definições de **nbk\_1.nb** a partir de outro Notebook dando o comando nativo **Get["/directory/of/notebook/nbk\_1.nb"]**. Note que aqui todas as células de Input de **nbk\_1.nb** são avaliadas na ordem em que estão escritas. Todas as outras células de tipo diferente são também lidas mas não são avaliadas.

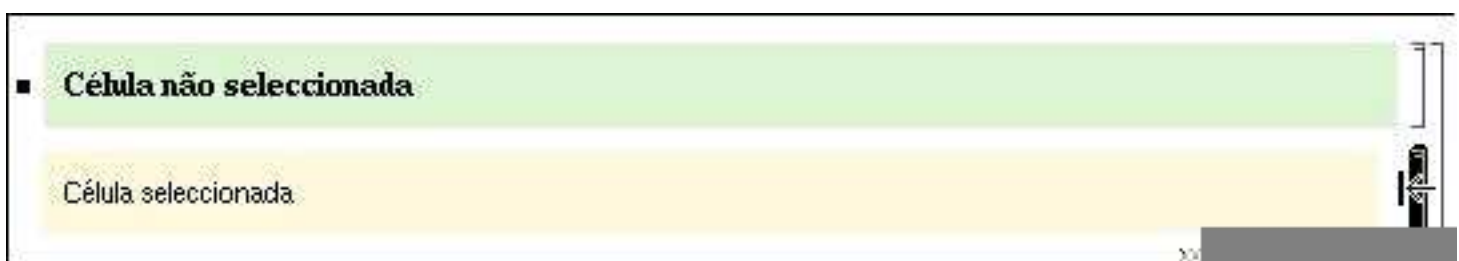
## □ TIPOS DE CÉLULAS


Existem vários tipos de Células que o FrontEnd reconhece, mas por regra (e excluindo indicação em contrário) toda a vez que se cria uma célula nova ela é do tipo Input, o que quer dizer que, para além de outras propriedades, é uma célula que pode ser enviada ao Kernel para avaliação. Também por regra, quando o Kernel devolve o resultado da avaliação, este é escrito numa ou mais células inertes (i.e. que não é possível alterar directamente) do tipo Output, geralmente logo a seguir à célula de Input que se enviou para avaliação do Kernel. Se o resultado da avaliação for uma expressão simbólica ou numérica, as células inertes são automaticamente formatadas em notação bi-dimensional. O Kernel pode também devolver gráficos, e nessa altura é criada uma ou mais células do tipo Graphics seguida duma célula de Output com referência ao gráfico criado. Os tipos de células mais vulgarmente usadas são introduzidas usando o menu **Format** ▸ **Style** e escolhendo um dos estilos apresentados na lista do submenu representado na imagem junta. O nome e a ordem dos tipos de células depende de informação guardada em Notebooks de estilos-padrão. Qualquer novo Notebook vem com os tipos de células definidas em Default.nb. Estes podem ser alterados e aumentados (efectivamente criando novos tipos de células) editando a folha de estilos através do menu **Format** ▸ **Edit Style Sheet...**. Estas novas definições ficam sempre associadas ao Notebook (físicamente escrita no Notebook) e não alteram as definições dos estilos-padrão que podem ser invocados para outros Notebooks.

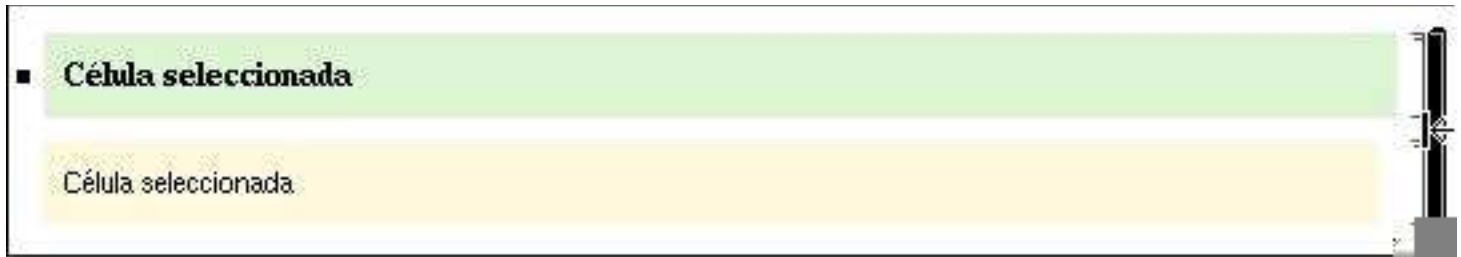
Title	Mod1+1
Subtitle	Mod1+2
Subsubtitle	Mod1+3
Section	Mod1+4
Subsection	Mod1+5
Subsubsection	Mod1+6
Text	Mod1+7
SmallText	Mod1+8
Input	Mod1+9
Output	
InlineFormula	
DisplayFormula	
Program	
Outline1	
Outline2	
Outline3	
Outline4	
Other...	Mod1+0

## □ SELECÇÃO DE CÉLULAS

Qualquer célula dum Notebook vem dotada de um **delimitador** que se parece com um semi-parentheses recto de fecho **]** no lado direito da janela onde está inscrita. Este delimitador é colorido de forma uniforme para todas as células do Notebook, e tem uma altura que geralmente abarca o número de linhas que a célula contém, deformando-se se necessário quando se aumenta ou diminui este número com a largura da janela ou da largura de página. Este delimitador serve entre outras coisas para **Seleccionar** toda a célula, o que acontece se pressionar o cursor sobre delimitador no instante em que muda de aspecto para algo como o icon **|←**.



As células dum Notebook podem ser agrupadas num conjunto designado **CellGroup** (que é apenas explicitamente visível no formato ASCII do Notebook e não na forma desformatada das células no FrontEnd), e estes por sua vez podem também ser agrupados noutros **CellGroups**. No FrontEnd, a cada grupo corresponde um delimitador adicional hierarquicamente colocado à direita dos delimitadores das células que agrupa. Alguns destes grupos são gerados sempre automaticamente, como por exemplo o do par Input e Output correspondente, outros podem ser criados manualmente através do menu **Cell** ▸ **Cell Grouping** ▸ **Group Cells** desde que as células seleccionadas sejam contíguas e que a opção **Cell** ▸ **Cell Grouping** ▸ **Manual Grouping** esteja activada. A selecção de um grupo de células faz-se pressionando o cursor  sobre o delimitador do grupo.







Todas as células do **Mathematica** são seleccionáveis e podem ser

- (a) – copiadas ou duplicadas (Select + **ALT** **C**) ou no menu **Edit** ▸ **Copy**,
- (b) – cortadas ou apagadas (Select + **ALT** **X**) ou no menu **Edit** ▸ **Cut**,
- (c) – divididas em duas (**ALT** **SHIFT** **D**) na posição do cursor,
- (d) – coladas ou inseridas (**ALT** **V**) noutro local do mesmo ou outro **Notebook**.

Cada célula de tipo **Input** em qualquer parte dum **Notebook** pode ser avaliada individualmente, bastando para isso **seleccioná-la** com o cursor (o que acontece se o cursor estiver dentro da célula, por exemplo se acabou de introduzir ou alterar uma expressão nela, ou ), e depois carregar na tecla **ENTER** ou na combinação **SHIFT** **RET** para a sua **avaliação**.

A avaliação de várias células sequencialmente (i.e. pela ordem que aparecem no **Notebook** do princípio para o fim) pode ser obtida seleccionando simultaneamente várias células de **Input** antes de pedir avaliação com as teclas indicadas acima.

- (a) – Para seleccionar células consecutivas basta começar na primeira e arrastar o cursor  mantendo-o pressionado.
- (b) – Pode-se igualmente seleccionar a primeira e usar a combinação de teclas **SHIFT**  ou **SHIFT** .
- (c) – Para seleccionar células que não são consecutivas use **CTRL** e pressione o cursor  sobre os delimitadores das células pretendidas.

## ■ ESTILOS E OPÇÕES PARA CÉLULAS

Cada *Notebook* vem dotado de uma folha de estilos (ing. **StyleSheet**), em que se definem os formatos a dar a cada um dos tipos de células disponíveis. Existem estilos pré-definidos em Notebooks–padrão que se podem importar e depois alterar ao gosto de cada um, use **Format** ▸ **Style Sheet** ▸ **Classic** por exemplo para alterar o visual do seu **Notebook** para um estilo designado **Classic**. A janela de estilos (StyleSheet) associada a cada **Notebook** é editável através do menu **Format** ▸ **Edit Style Sheet...**, e as alterações feitas ao estilo de cada tipo de célula (Title, ...,Section, ...,Text,...,Input, Output, etc) reflectem-se imediatamente no respectivo **Notebook**, com excepção das células do Notebook que tenham sofrido individualmente alterações de estilo.



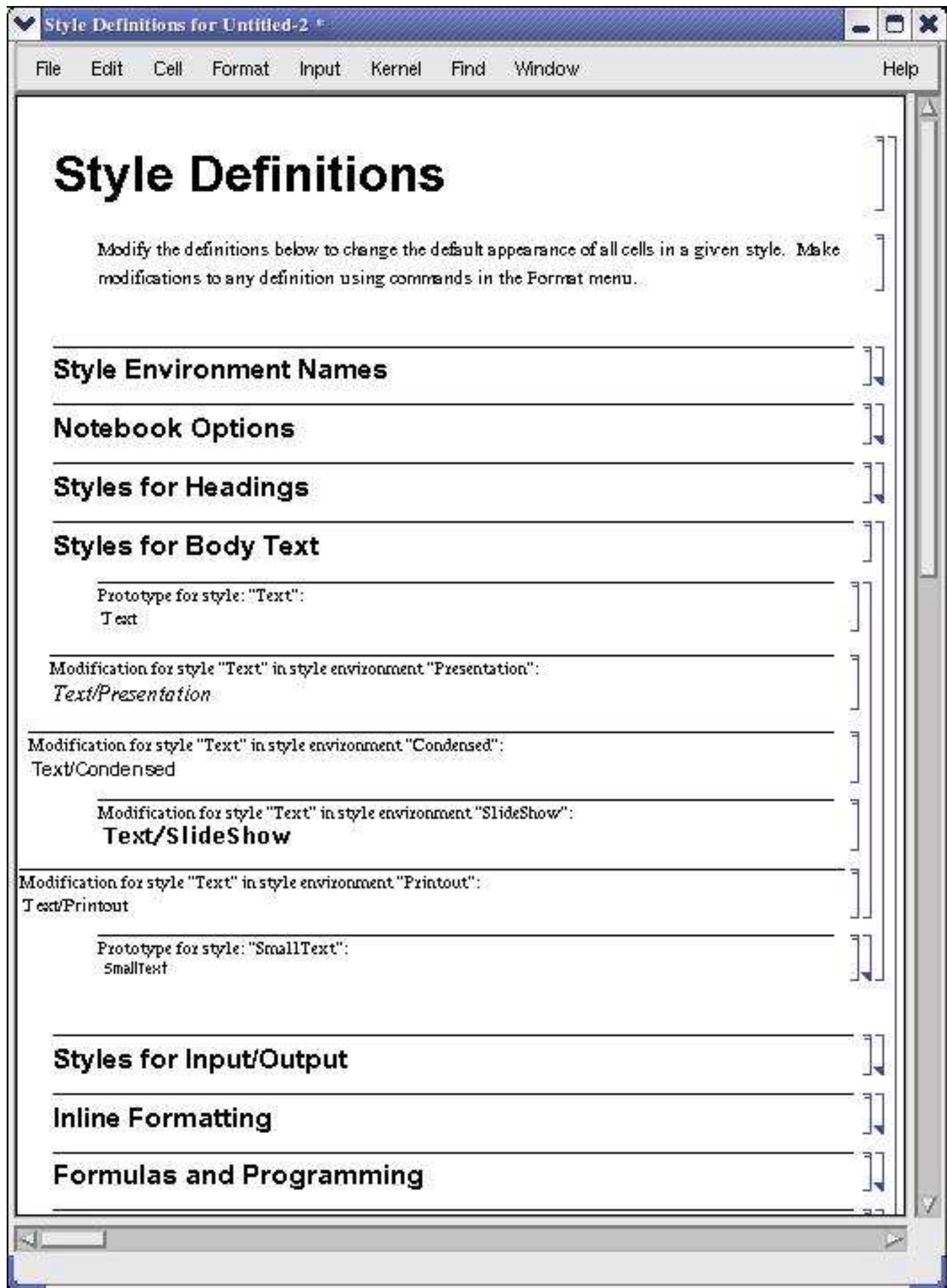


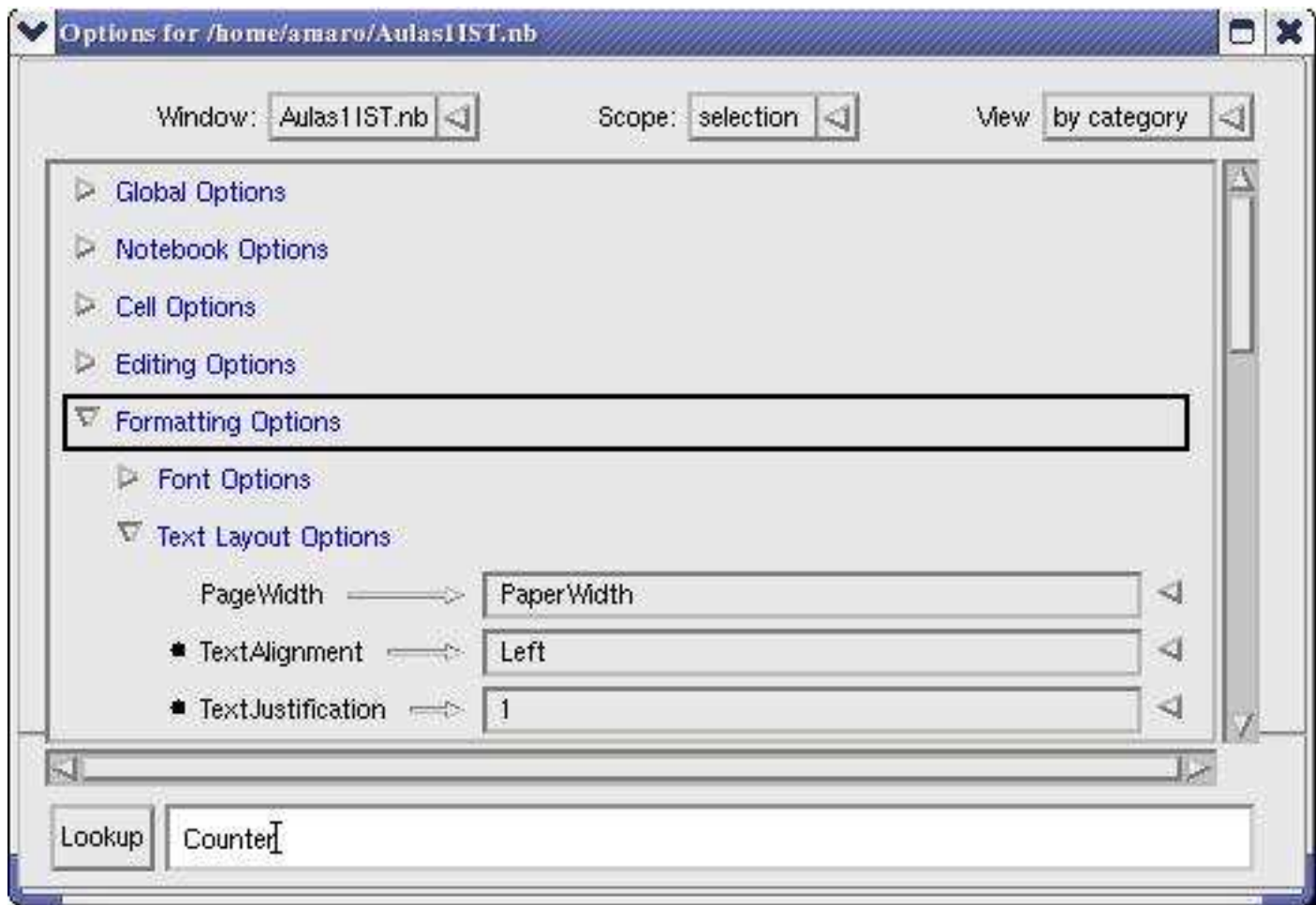
Figura 6



Na janela de estilos pode-se ver que cada tipo de célula (por exemplo **Text** na Figura 6) tem vários estilos diferentes dependendo do "Ambiente" em que se pretende usar o Notebook. De acordo com a folha de estilos da figura 6, o Notebook em questão tem para além do ambiente normal (**Working**), os ambientes **Presentation**, **Condensed**, **SlideShow** e **Printout**. É importante salientar que **Text/Printout** por exemplo mostra como sairá uma impressão das células de tipo **Text** do Notebook. Assim, o aspecto do Notebook no ambiente normal de trabalho **Working** não é necessariamente aquele com que ficará no papel, porque as definições de estilo do ambiente de trabalho escolhido para o écran podem ser diferentes dos de impressão. O utilizador tem a opção de seleccionar um ambiente de trabalho (por exemplo **Printout**) no menu **Format** ▸ **Screen Style Environment** ▸ **Printout** para visualizar o Notebook no écran e corrigir o aspecto do documento a imprimir.

Para alterar um estilo-padrão use **Format** ▸ **Edit Style Sheet ...** para abrir a folha de estilos ou, se ainda não tiver uma, criar uma nova que ficará associada ao Notebook em que está a trabalhar. Poderá então editar os estilos de cada célula e definir até novos tipos de células nesta folha de estilos, e em geral opções globais ou de todo o notebook, usando o menu **Format** ▸ **Option Inspector ...** ou **Edit** ▸ **Preferences**. Em geral e por norma a janela de Preferências mostra as opções disponíveis e activadas para uma selecção de células (quer na folha de estilos afectando todas as células do mesmo tipo no Notebook, quer no próprio Notebook afectando apenas essas células seleccionadas: podemos assim ter células de tipo **Input** com estilos completamente diferentes no mesmo Notebook).

Um menu (**Scope**) no topo da janela de Preferências permite seleccionar o âmbito de aplicação das alterações efectuadas conforme indica **Selection** (apenas dá acesso a opções das células seleccionadas), **Notebook** (permite alterar opções aplicáveis a todo o Notebook) e **Global** (opções que alteram o comportamento de todos os Notebooks em todas as sessões de trabalho subsequentes – de preferência não usar ou se tiver mesmo que usar faça-o com muito cuidado e apenas se souber o que faz!!!)



**Figura 7**

Seleccionando uma célula ou grupo de células na Janela de Estilos (Figura 6) de um dado Notebook, a aplicação de uma alteração nas opções mostradas na Janela de Preferências (Figura 7) tornar-se-há no novo padrão desse tipo de células e reflecte-se imediatamente em todas as células do Notebook que sejam dos tipos seleccionados, desde que estas não tenham tido individualmente alterações explícitas ao seu estilo-padrão. De facto, seleccionando uma célula directamente no Notebook e alterando algumas das suas opções através da Janela de Preferências ou através de menus apenas afecta a célula seleccionada no Notebook e estas opções sobrepõem-se às do padrão para o seu tipo de célula. Para remover alterações ao estilo-padrão de uma célula seleccione-a e use **Format** ▸ **Remove Options...**.

## ■ FORMATAÇÃO DE CÉLULAS

Toda a formatação das expressões introduzidas corresponde a informação (opções e estilos) que é automaticamente incluída na célula, mas fica invisível a menos que se desformate a mesma com o comando da Barra de Menus **Format** ▸ **Show Expression** (ou a equivalente combinação de teclas **ALT** **SHIFT** **[E]**). Por exemplo, a seguinte célula aparece formatada por regra no **FrontEnd**:

$$\alpha = \text{Tan}[\theta^2]$$

Contudo (embora seja muito raramente necessário ou útil) pode-se ver qual o código que está subjacente a esta formatação: seleccionando a célula anterior e usando **Format** ▸ **Show Expression** obtém-se

```
Cell[BoxData[ RowBox[{"α", "=", RowBox[{"Tan", "[", SuperscriptBox["θ", "2"], "]" } ]}], "Input"]
```

A estrutura dum célula de **Input** é assim uma sequência de caixas (**RowBox[]**) para cada sub-expressão que não possa ser reduzida a um símbolo como a função nativa **Tan[]**. Isto facilita a verificação sintática das expressões, como por exemplo a falta de emparelhamento de parênteses. A própria notação bidimensional para  $\theta^2$  é representada como uma caixa do tipo **SuperscriptBox[]** e todo o conjunto de caixas é incluído dentro de uma caixa maior **BoxData[]** característica de expressões que se pretendem enviar ao Kernel para avaliação.

Uma célula de texto tem uma estrutura semelhante. Por exemplo:

Célula de tipo "Text".

*A expressão  $x^2$  está inserida numa célula de texto.*

tem a representação desformatada:

```
Cell[TextData[ {
  "A expressão ",
  Cell[BoxData[SuperscriptBox["x", "2"]]],
  " está inserida numa célula de texto."
} ], "Text",
CellLabel->{"Célula de tipo \"Text\"."}, ShowCellTags->True,
Background->RGBColor[0.9, 0.9, 0.7]]
```

Note-se que agora a estrutura desta célula mudou para **TextData[]** indicando que a célula não é em geral avaliável e envolvendo uma lista de "expressões textuais" (ing. **Strings**) ou células inteiras como a da expressão bi-dimensional  $x^2$  que aparece inserida como uma célula dentro da célula de texto, i.e. o que se chama uma **Inline Cell**.

A forma geral do código subjacente a uma célula num Notebook do Mathematica é assim **Cell[XData[boxes], Type, formatting\_options & labels]** onde:

- (a) – **XData** representa em geral **TextData**, **BoxData** ou **GraphicsData**, respectivamente para células de texto, fórmulas ou gráficos;
- (b) – **boxes** representa geralmente uma lista de **Strings** e de elementos do tipo **RowBox[...]** para expressões bi-dimensionais ou **GridBox[...]** para matrizes ou ainda outros tipos de **Box**. Pode igualmente conter outras células **Cell[...]** designadas de **Inline**. O comando **Names["\*Box"]** mostra todos os símbolos terminados em **Box** que o sistema conhece:

```
Names["*Box"]
```

AdjustmentBox	ErrorBox	OverscriptBox	SubsuperscriptBox
Box	FormBox	RadicalBox	SuperscriptBox
ButtonBox	FractionBox	RowBox	TagBox
CellBoundingBox	FrameBox	ShrinkWrapBoundingBox	TextBoundingBox
CellElementsBoundingBox	GridBox	SqrtBox	UnderoverscriptBox
ContentsBoundingBox	InterpretationBox	StyleBox	UnderscriptBox
CounterBox	OptionValueBox	SubscriptBox	ValueBox

(c) – **Type** representa o tipo e estilo de célula, para ser formatada de acordo com as definições padrão armazenadas no **StyleSheet** do notebook.

Estas definições de estilo são copiadas de Notebooks de estilos armazenados em **SystemFiles/FrontEnd/StyleSheets**, mas são escritas no fim do próprio Notebook onde se trabalha (embora apenas sejam visíveis e alteráveis na janela de **StyleSheet** do notebook ou abrindo este num editor de texto ASCII).

(d) – os argumentos opcionais **formatting\_options & labels** representam uma sequência de regras **Propriedade → valor** implementando alterações locais da formatação–padrão imposta por **Type**.

O comando **DisplayForm** permite formatar uma célula no **Output** como uma célula de tipo **Title**.

```
DisplayForm[
  Cell[BoxData[SuperscriptBox["Mathematica",
    StyleBox["TM", FontSize → 9, FontColor → RGBColor[1, 0, 0]]],
    "Title", Background → White, FontFamily → "Times", FontSize → 24,
    CellFrame → True, TextAlignment → Center]]
```

**Mathematica<sup>TM</sup>**

As diferentes estruturas do tipo **Box** são usadas internamente para implementar, directamente ou através de opções explicitadas, as alterações ao estilo padronizado do texto num determinado tipo de célula. Use o comando **Options[nomeBox]** para ter uma ideia das opções que pode alterar naquele tipo de **Box**. Por exemplo:

**Options[GridBox]**

AllowScriptLevelChange → True	GridFrameMargins → {{0.4, 0.4}, {0.5, 0.5}}
AutoDelete → True	MultilineFunction → None
ColumnAlignments → {Center}	RowAlignments → Baseline
ColumnLines → False	RowLines → False
ColumnsEqual → False	RowMinHeight → 1.
ColumnSpacings → 0.8	RowsEqual → False
ColumnWidths → Automatic	RowSpacings → 1.
GridBaseline → Axis	GridDefaultElement → □
GridFrame → False	

**GridBox** é a estrutura de qualquer tabela formatada no FrontEnd. A maneira mais cómoda de mudar as suas opções uma vez gerada é seleccioná-la, abrir a Janela de Opções e procurar pelo seu nome. Alterando por exemplo **ColumnLines** → **True** e **RowLines** → **True** aparecem traços verticais entre colunas e horizontais entre linhas. **ColumnAlignments** pode ser **Left**, **Center**, **Right** ou uma lista destes elementos a aplicar sucessivamente às diferentes colunas.

## ■ FORMATAÇÃO DE TEXTO

### □ ESTILOS DE TEXTO

O texto introduzido num **Notebook** pode ser formatado de inúmeras formas, geralmente seleccionando e usando os menus adequados para o efeito. Por exemplo, seleccionando uma expressão como "Isto é um texto em ..." e escolhendo um tipo (**Font**) no menu **Format** ▶ **Font** obtém-se:

Courier	→	Isto é um texto em Courier
Arial Black	→	<b>Isto é um texto em Arial Black</b>
Helvetica	→	Isto é um texto em Helvetica
Lucida Handwriting	→	<i>Isto é um texto em Lucida Handwriting</i>

**Format** ▶ **Text Color** ▶ **Red** **Isto é um texto em vermelho.**

**Format** ▶ **Background Color** ▶ **Blue** **Isto é um texto em fundo azul.**

Existem combinações de teclas apropriadas para uma série de alterações de estilo no texto, algumas das quais funcionam em modo de alternância (ing. **Toggle**) (**True** / **False**):

**[ALT] [SHIFT] [b]** ou **Format** ▶ **Face** ▶ **Bold**

**[ALT] [i]** ou **Format** ▶ **Face** ▶ **Italic**

**[ALT] [+]** ou **Format** ▶ **Size** ▶ **Larger**

**[ALT] [-]** ou **Format** ▶ **Size** ▶ **Smaller**

Estas são apenas algumas das opções de formatação disponíveis no FrontEnd do *Mathematica*. Para aceder a todas as possibilidades de formatação de um texto, seleccione-o e abra a Janela de Preferências com **[ALT] [SHIFT] [O]**. Aí procure pelo título **Formatting Options** e aplique as propriedades que estão disponíveis. Abrindo a Janela de Ajuda com **Help** ▶ **Help Browser**, o capítulo **Front End** ▶ **Fron End Options** ▶ **Formatting Options** explica o significado das opções que encontra disponíveis na Janela de Preferências.

Note que em células de **Input** as alterações de estilo não afectam a identidade ou o processamento das expressões.

## □ AMBIENTES DE TRABALHO E IMPRESSÃO DE NOTEBOOKS

A visualização do aspecto real dos parágrafos numa impressão em papel dum Notebook pode ser obtida mudando o ambiente de trabalho do Notebook para **Printout** através do menu **Format** ▸ **Screen Style Environment** ▸ **Printout**. Neste ambiente são aplicadas as alterações de estilo **Printout** dos diferentes tipos de células consignadas pela Janela de Estilos do Notebook, pelo que todas as alterações necessárias para efeitos de impressão devem ser feitos de preferência nessa versão das células-tipo para não alterar o aspecto do Notebook no ambiente padrão (**Working**).

Note que o processamento de tipos de letra (ing. **Fonts**) pelo FrontEnd do *Mathematica* nem sempre se reflecte na impressão em papel do Notebook. Em Linux todo o processo de impressão passa por um utilitário **psrender** incluído na distribuição do *Mathematica*, que converte o Notebook (ou a parte seleccionada para impressão) para PostScript. Para garantir a impressão correcta de símbolos exclusivos do Mathematica é sempre conveniente activar a opção **Include Mathematica fonts in document** que aparece em geral na janela de impressão quando se faz **File** ▸ **Print...** ( **ALT** **P** ) ou **File** ▸ **PrintSelection...** ( **ALT** **SHIFT** **P** ). Embora o *Mathematica* tenha uma tabela de substituição de tipos para garantir alguma funcionalidade de impressão, a utilização de tipos mais exóticos para além do Courier, Helvetica, Times e Symbol está condicionada às capacidades de associar os nomes gerados por **psrender** ao nomes reais dos tipos disponíveis no sistema para impressão. A opção de estilo de texto `PrivateFontOptions → {FontPostScriptName → "some font name"}` permite alguma compatibilidade entre nomes de tipos no *Mathematica* e da impressora, mas há mais variáveis que podem inviabilizar essa compatibilidade.

## ■ ETIQUETAS E NUMERAÇÃO AUTOMÁTICA DE CÉLULAS

### □ HYPERLINKS

Qualquer texto numa célula se pode transformar numa conexão ou **hyperlink** apontando para outras células no mesmo **Notebook**, ou para outros **Notebooks** ou mesmo para **URLs** na Internet. Basta seleccioná-lo e usar **Input** ▸ **Create Hyperlink...** e definir o alvo desejado.

### □ ETIQUETAS

Por vezes é conveniente "etiquetar" (ing. **Tag**) e "enumerar" Células e Equações para poder rapidamente localizá-las através de **hyperlinks** como acima indicado. Para isso é preciso etiquetar com um nome (por ex. **tag1**) uma ou mais células seleccionadas previamente usando a combinação de teclas **ALT** **J** ou usando uma sequência de comandos do menu **Find** ▸ **Add/Remove Cell Tags...** ▸ **Cell Tag:** **tag1** e posteriormente referenciá-la por exemplo através dum **hyperlink** com o seu nome (mesmo que noutro notebook) com a sequência **Input** ▸ **Create Hyperlink...** ▸ ♦ **Cells with Tag** ▸ **tag1**.

Etiqueta 1, Etiqueta 2, TeX

Esta célula tem `CellTags → {"Etiqueta 1", "Etiqueta 2", "TeX"}`

Como se pode ver uma célula pode armazenar múltiplas etiquetas. Uma utilização muito frequente de etiquetas é a de seleccionar duma vez só todas as células que no Notebook têm uma

determinada etiqueta, por exemplo a etiqueta "**TeX**". Para isso basta dar o comando do Front-End,

```
Nbk = SelectedNotebook[]; NotebookFind[Nbk, "TeX", All, CellTags];
```

onde **SelectedNotebook[]** indica o Notebook onde se está a trabalhar.

#### □ NUMERAÇÃO DE CÉLULAS E LISTAS DE ITEMS

A numeração automática de certas células (em particular Capítulos, Secções e Equações ou Gráficos) é um pouco mais complexa, mas uma vez disponível na folha de estilos tipos de células com numeração (por exemplo `NumberedEquation`, `NumberedFigure` ou **NumberedTable**) é fácil introduzir em qualquer ponto do Notebook um hyperlink exibindo o número da célula numerada com o contador `CounterBox["NumberedCellType"]` usando o menu **Input** ▸ **Create Automatic Numbering Object...**.

Para criar um novo tipo de célula numerada é necessário dizer que os diferentes estilos de células dum *Notebook* são definidos e alterados na janela de estilos do Notebook que se abre usando **Format** ▸ **Edit Style Sheet...**.

- (a) – Selecciona na janela de estilos o tipo de célula que pretende enumerar automaticamente, por ex. **Section**.
- (b) – Use **Format** ▸ **Option Inspector...** ou pressione **ALT** **SHIFT** **O** para abrir a janela de opções para a célula seleccionada.
- (c) – Localize a opção **CellFrameLabels** usando a tecla virtual para pesquisas nos títulos da janela de preferências **Lookup** **CellFrameLab**.
- (d) – Escrevendo `CounterBox["Section"]` no campo indicado "left", obter-se-há uma enumeração automática à esquerda de cada célula de tipo **Section**.
- (e) – Para definir o estilo do contador procure com **Lookup** **CounterBoxOptions** e seleccione um dos estilos disponíveis de enumeração.

Finalmente, se pretende re-inicializar o contador de **Section** quando começa um novo capítulo encabeçado por **Subtitle**, por exemplo, é necessário indicar na definição do estilo de célula **Subtitle** que deve pôr a zero o contador das células de tipo **Section**. Selecciona a célula **Subtitle**, e na janela de opções aberta com **ALT** **SHIFT** **O** procure por **Lookup** **CounterAssignments** e altere para **CounterAssignments** → `{{"Section", 0}}`



## ■ EXPRESSÕES BI-DIMENSIONAIS

No meio de texto que não seja do estilo **Input**, expressões bi-dimensionais como  $x^2$ ,  $\sqrt{x}$  ou  $\frac{y}{x}$  são introduzidas directamente com a combinação adequada de teclas indicadas na tabela seguinte, mas o **FrontEnd** cria automaticamente sub-células designadas **InlineCell** na célula de texto. (Para criar uma destas sub-células vazia use **CTRL** **9**.) A formatação de texto nestas sub-células é independente da do texto que as rodeia e do estilo da célula em que estão inscritas.

Em células do estilo **Input**, apenas se devem escrever dados para avaliação pelo **Kernel** do **Mathematica**, e embora se possa ainda formatar o que aí se escreve (colorir texto, aumentar o tamanho, etc.), esta formatação é ignorada na avaliação, com as excepções representadas na tabela seguinte.

Acção	Teclas	Exemplo	Resultado
Superscript	<b>CTRL</b> <b>6</b>	x <b>CTRL</b> <b>6</b> 2	$x^2$
Subscript	<b>CTRL</b> <b>-</b>	x <b>CTRL</b> <b>-</b> k	$x_k$
Above	<b>CTRL</b> <b>7</b>	x <b>CTRL</b> <b>7</b> ~	$\tilde{x}$
Below	<b>CTRL</b> <b>=</b>	D <b>CTRL</b> <b>=</b> n	$D_n$
Opposite Position	<b>CTRL</b> <b>5</b>	D <b>CTRL</b> <b>=</b> n <b>CTRL</b> <b>5</b> k	$D_n^k$
Fraction	<b>CTRL</b> <b>/</b>	a <b>CTRL</b> <b>/</b> b	$\frac{a}{b}$
Radical	<b>CTRL</b> <b>2</b>	<b>CTRL</b> <b>2</b> a	$\sqrt{a}$
Add Row	<b>CTRL</b> <b>↵</b>	( a b □ <b>CTRL</b> <b>↵</b> )	$\begin{pmatrix} a & b & \square \\ \square & \square & \square \end{pmatrix}$
Add Column	<b>CTRL</b> <b>,</b>	$\begin{pmatrix} a & b & \square \\ \square & \square & \square \end{pmatrix}$	$\begin{pmatrix} a & b & \square \\ \square & \square & \square \end{pmatrix}$
End Subexpression	<b>CTRL</b> <b>SPACE</b>	x <b>CTRL</b> <b>6</b> 2 t <b>CTRL</b> <b>SPACE</b> y	$x^{2t}y$
Extend Selection	<b>CTRL</b> <b>.</b>	<b>a</b> + b <sup>2</sup>	<b>a + b<sup>2</sup></b>
Start Inline Cell	<b>CTRL</b> <b>9</b>	valor <b>CTRL</b> <b>9</b> x <b>CTRL</b> <b>6</b> 2	valor <b>x<sup>2</sup></b>
End Inline Cell	<b>CTRL</b> <b>0</b>	este <b>x<sup>2</sup></b>	este <b>x<sup>2</sup></b> é ...

Tabela 1

## ■ CARACTERES ESPECIAIS

O *Mathematica* aceita todo o tipo de caracteres como parte de texto, dados ou resultados. É possível introduzir caracteres especiais de 3 formas num *Notebook*:

☞ **\[UnicodeName]** —

Todos os caracteres especiais têm um nome **Unicode**. Por exemplo, o nome **\[Alpha]** representa  $\alpha$ , **\[DoubleContourIntegral]** é o nome do símbolo  $\oint$  e **\[Infinity]** é a designação para  $\infty$ . Na forma desformatada das células os caracteres especiais aparecem substituídos pelos seus nomes **Unicode**. Mais à frente pode-se ver uma tabela com uma lista completa dos nomes **Unicode** para os caracteres especiais que o Mathematica conhece.



## ☞☞☞ Paleta de Caracteres especiais —

Com a seguinte sequência de comandos da barra de menus do Mathematica

**File** ▸ **Palettes** ▸ **CompleteCharacters**

abre-se uma janela dum tipo não editável denominado Palette (literalmente Paleta) com teclas virtuais para quase todos os caracteres especiais mais frequentemente usados. Movimentando o ponteiro sobre esta janela seleccione o símbolo que pretende inserir na posição onde vai escrever o próximo carácter no Notebook em que está actualmente a trabalhar. Pressione a tecla virtual para inserir o símbolo nela representado. Note que quando o ponteiro está sobre uma tecla virtual da paleta, o nome do respectivo carácter especial aparece escrito na borda inferior da janela da paleta. No caso da imagem o símbolo  $\sqrt{\phantom{x}}$  tem o nome Unicode `\[Sqrt]` ou a abreviatura `[ESC]sqrt[ESC]` como indicado no canto inferior esquerdo. Alguns símbolos não têm abreviatura, mas é possível adicionar definições particulares de abreviaturas usando o comando `AddInputAlias` definido no módulo de extensão do Mathematica (Package) denominado `Notations`, e que pode ser carregada usando o menu

**File** ▸ **Palettes** ▸ **Notations**







☞☞☞ `[ESC]abrev[ESC]` — Muitos caracteres especiais têm abreviaturas que, quando enquadradas pelos símbolos `⌈` introduzidos pela tecla `[ESC]`, reproduzem o carácter como se se tivesse escrito o seu nome como indicado antes. Por exemplo, `⌈a⌋` é  $\alpha$ , ou `⌈inf⌋` que é  $\infty$ . Como é natural `⌈:-)⌋` é ☺, o `\[HappySmiley]`. Nas paletas referidas no método anterior também aparece escrito na borda inferior da janela da paleta a abreviatura dum carácter especial quando se coloca o cursor sobre o respectiva tecla virtual. Note que nem todos os caracteres especiais têm abreviaturas, embora todos tenham um nome **Unicode**.

□ CÓDIGOS UNICODE DE CARACTERES ESPECIAIS

<code>\[CapitalEBar]</code>	Ē	<code>\[CapitalZeta]</code>	Z
<code>\[CapitalECup]</code>	Ė	<code>\[CCedilla]</code>	ç
<code>\[CapitalEDoubleDot]</code>	Ê	<code>\[Cedilla]</code>	¸
<code>\[CapitalEGrave]</code>	È	<code>\[Cent]</code>	¢
<code>\[CapitalEHat]</code>	Ê	<code>\[CenterDot]</code>	·
<code>\[CapitalEpsilon]</code>	Ε	<code>\[CenterEllipsis]</code>	⋯
<code>\[CapitalEta]</code>	Η	<code>\[CHacek]</code>	č
<code>\[CapitalEth]</code>	Ð	<code>\[CheckedBox]</code>	☑
<code>\[CapitalGamma]</code>	Γ	<code>\[Checkmark]</code>	✓
<code>\[CapitalIAcute]</code>	Í	<code>\[CheckmarkedBox]</code>	☑
<code>\[CapitalICup]</code>	İ	<code>\[Chi]</code>	χ
<code>\[CapitalIDoubleDot]</code>	Î	<code>\[CircleDot]</code>	⊙
<code>\[CapitalIGrave]</code>	Ì	<code>\[CircleMinus]</code>	⊖
<code>\[CapitalIHat]</code>	Î	<code>\[CirclePlus]</code>	⊕
<code>\[CapitalIota]</code>	Ι	<code>\[CircleTimes]</code>	⊗
<code>\[CapitalKappa]</code>	Κ	<code>\[ClockwiseContourIntegral]</code>	∮
<code>\[CapitalKoppa]</code>	Ϟ	<code>\[CloseCurlyDoubleQuote]</code>	”
<code>\[CapitalLambda]</code>	Λ	<code>\[CloseCurlyQuote]</code>	’
<code>\[CapitalLSlash]</code>	Ł	<code>\[CloverLeaf]</code>	♣
<code>\[CapitalMu]</code>	Μ	<code>\[ClubSuit]</code>	♣
<code>\[CapitalNTilde]</code>	Ñ	<code>\[Colon]</code>	:
<code>\[CapitalNu]</code>	Ν	<code>\[CommandKey]</code>	⌨
<code>\[CapitalOAcute]</code>	Ó	<code>\[Congruent]</code>	≡
<code>\[CapitalODoubleAcute]</code>	Ő	<code>\[Conjugate]</code>	*
<code>\[CapitalODoubleDot]</code>	Ö	<code>\[ConjugateTranspose]</code>	↔
<code>\[CapitalOGrave]</code>	Ò	<code>\[ConstantC]</code>	c
<code>\[CapitalOHat]</code>	Ô	<code>\[Continuation]</code>	⋮
<code>\[CapitalOmega]</code>	Ω	<code>\[ContourIntegral]</code>	∮
<code>\[CapitalOmicron]</code>	Ο	<code>\[ControlKey]</code>	⌨
<code>\[CapitalOSlash]</code>	Ø	<code>\[Coproduct]</code>	∐
<code>\[CapitalOTilde]</code>	Ȯ	<code>\[Copyright]</code>	©
<code>\[CapitalPhi]</code>	Φ	<code>\[CounterClockwiseContourIntegral]</code>	∮
<code>\[CapitalPi]</code>	Π	<code>\[Cross]</code>	×
<code>\[CapitalPsi]</code>	Ψ	<code>\[Cup]</code>	∪
<code>\[CapitalRho]</code>	Ρ	<code>\[CupCap]</code>	∩
<code>\[CapitalSampi]</code>	Ϻ	<code>\[CurlyCapitalUpsilon]</code>	Υ
<code>\[CapitalSHacek]</code>	Š	<code>\[CurlyEpsilon]</code>	ε
<code>\[CapitalSigma]</code>	Σ	<code>\[CurlyKappa]</code>	κ
<code>\[CapitalStigma]</code>	Ϛ	<code>\[CurlyPhi]</code>	φ
<code>\[CapitalTau]</code>	Τ	<code>\[CurlyPi]</code>	π
<code>\[CapitalTheta]</code>	Θ	<code>\[CurlyRho]</code>	ρ
<code>\[CapitalThorn]</code>	Þ	<code>\[CurlyTheta]</code>	θ
<code>\[CapitalUAcute]</code>	Ú	<code>\[Currency]</code>	¤
<code>\[CapitalUDoubleAcute]</code>	Û	<code>\[Dagger]</code>	†
<code>\[CapitalUDoubleDot]</code>	Ü	<code>\[Dalet]</code>	ד
<code>\[CapitalUGrave]</code>	Ù	<code>\[Dash]</code>	-
<code>\[CapitalUHat]</code>	Û	<code>\[Degree]</code>	°
<code>\[CapitalUpsilon]</code>	Υ	<code>\[Del]</code>	∇
<code>\[CapitalXi]</code>	Ξ	<code>\[DeleteKey]</code>	⌨
<code>\[CapitalYAcute]</code>	Ý	<code>\[Delta]</code>	δ

<code>\[DescendingEllipsis]</code>	$\ddots$	<code>\[DoubleStruckCapitalU]</code>	$\mathbb{U}$
<code>\[Diameter]</code>	$\varnothing$	<code>\[DoubleStruckCapitalV]</code>	$\mathbb{V}$
<code>\[Diamond]</code>	$\diamond$	<code>\[DoubleStruckCapitalW]</code>	$\mathbb{W}$
<code>\[DiamondSuit]</code>	$\diamond$	<code>\[DoubleStruckCapitalX]</code>	$\mathbb{X}$
<code>\[DifferentialD]</code>	$d$	<code>\[DoubleStruckCapitalY]</code>	$\mathbb{Y}$
<code>\[Digamma]</code>	$f$	<code>\[DoubleStruckCapitalZ]</code>	$\mathbb{Z}$
<code>\[DiscretionaryHyphen]</code>	-	<code>\[DoubleStruckD]</code>	$\mathbb{d}$
<code>\[Divide]</code>	$\div$	<code>\[DoubleStruckE]</code>	$\mathbb{e}$
<code>\[DotEqual]</code>	$\doteq$	<code>\[DoubleStruckEight]</code>	$\mathbb{8}$
<code>\[DotlessI]</code>	$i$	<code>\[DoubleStruckF]</code>	$\mathbb{f}$
<code>\[DotlessJ]</code>	$j$	<code>\[DoubleStruckFive]</code>	$\mathbb{5}$
<code>\[DottedSquare]</code>	$\boxdot$	<code>\[DoubleStruckFour]</code>	$\mathbb{4}$
<code>\[DoubleContourIntegral]</code>	$\oint$	<code>\[DoubleStruckG]</code>	$\mathbb{g}$
<code>\[DoubleDagger]</code>	$\ddagger$	<code>\[DoubleStruckH]</code>	$\mathbb{h}$
<code>\[DoubledGamma]</code>	$\gamma$	<code>\[DoubleStruckI]</code>	$\mathbb{i}$
<code>\[DoubleDot]</code>	$\ddot{\phantom{x}}$	<code>\[DoubleStruckJ]</code>	$\mathbb{j}$
<code>\[DoubleDownArrow]</code>	$\Downarrow$	<code>\[DoubleStruckK]</code>	$\mathbb{k}$
<code>\[DoubledPi]</code>	$\pi$	<code>\[DoubleStruckL]</code>	$\mathbb{l}$
<code>\[DoubleLeftArrow]</code>	$\Leftarrow$	<code>\[DoubleStruckM]</code>	$\mathbb{m}$
<code>\[DoubleLeftRightArrow]</code>	$\Leftrightarrow$	<code>\[DoubleStruckN]</code>	$\mathbb{n}$
<code>\[DoubleLeftTee]</code>	$\Leftarrow$	<code>\[DoubleStruckNine]</code>	$\mathbb{9}$
<code>\[DoubleLongLeftArrow]</code>	$\Longleftarrow$	<code>\[DoubleStruckO]</code>	$\mathbb{o}$
<code>\[DoubleLongLeftRightArrow]</code>	$\Longleftrightarrow$	<code>\[DoubleStruckOne]</code>	$\mathbb{1}$
<code>\[DoubleLongRightArrow]</code>	$\Longrightarrow$	<code>\[DoubleStruckP]</code>	$\mathbb{p}$
<code>\[DoublePrime]</code>	$\prime\prime$	<code>\[DoubleStruckQ]</code>	$\mathbb{q}$
<code>\[DoubleRightArrow]</code>	$\Rightarrow$	<code>\[DoubleStruckR]</code>	$\mathbb{r}$
<code>\[DoubleRightTee]</code>	$\Rightarrow$	<code>\[DoubleStruckS]</code>	$\mathbb{s}$
<code>\[DoubleStruckA]</code>	$\mathbb{a}$	<code>\[DoubleStruckSeven]</code>	$\mathbb{7}$
<code>\[DoubleStruckB]</code>	$\mathbb{b}$	<code>\[DoubleStruckSix]</code>	$\mathbb{6}$
<code>\[DoubleStruckC]</code>	$\mathbb{c}$	<code>\[DoubleStruckT]</code>	$\mathbb{t}$
<code>\[DoubleStruckCapitalA]</code>	$\mathbb{A}$	<code>\[DoubleStruckThree]</code>	$\mathbb{3}$
<code>\[DoubleStruckCapitalB]</code>	$\mathbb{B}$	<code>\[DoubleStruckTwo]</code>	$\mathbb{2}$
<code>\[DoubleStruckCapitalC]</code>	$\mathbb{C}$	<code>\[DoubleStruckU]</code>	$\mathbb{u}$
<code>\[DoubleStruckCapitalD]</code>	$\mathbb{D}$	<code>\[DoubleStruckV]</code>	$\mathbb{v}$
<code>\[DoubleStruckCapitalE]</code>	$\mathbb{E}$	<code>\[DoubleStruckW]</code>	$\mathbb{w}$
<code>\[DoubleStruckCapitalF]</code>	$\mathbb{F}$	<code>\[DoubleStruckX]</code>	$\mathbb{x}$
<code>\[DoubleStruckCapitalG]</code>	$\mathbb{G}$	<code>\[DoubleStruckY]</code>	$\mathbb{y}$
<code>\[DoubleStruckCapitalH]</code>	$\mathbb{H}$	<code>\[DoubleStruckZ]</code>	$\mathbb{z}$
<code>\[DoubleStruckCapitalI]</code>	$\mathbb{I}$	<code>\[DoubleUpArrow]</code>	$\Uparrow$
<code>\[DoubleStruckCapitalJ]</code>	$\mathbb{J}$	<code>\[DoubleUpDownArrow]</code>	$\Updownarrow$
<code>\[DoubleStruckCapitalK]</code>	$\mathbb{K}$	<code>\[DoubleVerticalBar]</code>	$\parallel$
<code>\[DoubleStruckCapitalL]</code>	$\mathbb{L}$	<code>\[DownArrow]</code>	$\downarrow$
<code>\[DoubleStruckCapitalM]</code>	$\mathbb{M}$	<code>\[DownArrowBar]</code>	$\downdownarrows$
<code>\[DoubleStruckCapitalN]</code>	$\mathbb{N}$	<code>\[DownArrowUpArrow]</code>	$\Updownarrow$
<code>\[DoubleStruckCapitalO]</code>	$\mathbb{O}$	<code>\[DownBreve]</code>	$\breve{\phantom{x}}$
<code>\[DoubleStruckCapitalP]</code>	$\mathbb{P}$	<code>\[DownExclamation]</code>	$\jmath$
<code>\[DoubleStruckCapitalQ]</code>	$\mathbb{Q}$	<code>\[DownLeftRightVector]</code>	$\searrow$
<code>\[DoubleStruckCapitalR]</code>	$\mathbb{R}$	<code>\[DownLeftTeeVector]</code>	$\swarrow$
<code>\[DoubleStruckCapitalS]</code>	$\mathbb{S}$	<code>\[DownLeftVector]</code>	$\swarrow$
<code>\[DoubleStruckCapitalT]</code>	$\mathbb{T}$	<code>\[DownLeftVectorBar]</code>	$\swarrow$

<code>\[DownPointer]</code>	▼	<code>\[FilledSquare]</code>	■
<code>\[DownQuestion]</code>	¿	<code>\[FilledUpTriangle]</code>	▲
<code>\[DownRightTeeVector]</code>	↘	<code>\[FilledVerySmallSquare]</code>	▪
<code>\[DownRightVector]</code>	→	<code>\[FinalSigma]</code>	ς
<code>\[DownRightVectorBar]</code>	→	<code>\[FirstPage]</code>	⌞
<code>\[DownTee]</code>	⊥	<code>\[FivePointedStar]</code>	★
<code>\[DownTeeArrow]</code>	↓	<code>\[Flat]</code>	♭
<code>\[EAcute]</code>	é	<code>\[FLigature]</code>	■
<code>\[Earth]</code>	♁	<code>\[Florin]</code>	■
<code>\[EBar]</code>	ē	<code>\[ForAll]</code>	∀
<code>\[ECup]</code>	ě	<code>\[FreakedSmiley]</code>	☺
<code>\[EDoubleDot]</code>	ë	<code>\[Gamma]</code>	γ
<code>\[EGrave]</code>	è	<code>\[Gimel]</code>	ג
<code>\[EHat]</code>	ê	<code>\[GothicA]</code>	Ɽ
<code>\[EighthNote]</code>	♪	<code>\[GothicB]</code>	ⱥ
<code>\[Element]</code>	∈	<code>\[GothicC]</code>	ⱦ
<code>\[Ellipsis]</code>	...	<code>\[GothicCapitalA]</code>	Ⱨ
<code>\[EmptyCircle]</code>	○	<code>\[GothicCapitalB]</code>	ⱨ
<code>\[EmptyDiamond]</code>	◇	<code>\[GothicCapitalC]</code>	Ⱪ
<code>\[EmptyDownTriangle]</code>	▽	<code>\[GothicCapitalD]</code>	ⱪ
<code>\[EmptyRectangle]</code>	□	<code>\[GothicCapitalE]</code>	ⱬ
<code>\[EmptySet]</code>	∅	<code>\[GothicCapitalF]</code>	Ɑ
<code>\[EmptySmallCircle]</code>	◦	<code>\[GothicCapitalG]</code>	Ɱ
<code>\[EmptySmallSquare]</code>	◻	<code>\[GothicCapitalH]</code>	Ɐ
<code>\[EmptySquare]</code>	□	<code>\[GothicCapitalI]</code>	Ɒ
<code>\[EmptyUpTriangle]</code>	△	<code>\[GothicCapitalJ]</code>	Ⱳ
<code>\[EmptyVerySmallSquare]</code>	◼	<code>\[GothicCapitalK]</code>	ⱳ
<code>\[EnterKey]</code>		<code>\[GothicCapitalL]</code>	Ⱶ
<code>\[EntityEnd]</code>	␣	<code>\[GothicCapitalM]</code>	ⱶ
<code>\[EntityStart]</code>	␣	<code>\[GothicCapitalN]</code>	ⱷ
<code>\[Epsilon]</code>	ε	<code>\[GothicCapitalO]</code>	ⱸ
<code>\[Equal]</code>	=	<code>\[GothicCapitalP]</code>	ⱹ
<code>\[EqualTilde]</code>	≈	<code>\[GothicCapitalQ]</code>	ⱺ
<code>\[Equilibrium]</code>	⇌	<code>\[GothicCapitalR]</code>	ⱻ
<code>\[ErrorIndicator]</code>	☒	<code>\[GothicCapitalS]</code>	ⱼ
<code>\[EscapeKey]</code>		<code>\[GothicCapitalT]</code>	ⱽ
<code>\[Eta]</code>	η	<code>\[GothicCapitalU]</code>	Ȿ
<code>\[Eth]</code>	ð	<code>\[GothicCapitalV]</code>	Ɀ
<code>\[Euro]</code>	€	<code>\[GothicCapitalW]</code>	Ɀ
<code>\[Exists]</code>	∃	<code>\[GothicCapitalX]</code>	Ɀ
<code>\[ExponentialE]</code>	e	<code>\[GothicCapitalY]</code>	Ɀ
<code>\[FiLigature]</code>	■	<code>\[GothicCapitalZ]</code>	Ɀ
<code>\[FilledCircle]</code>	●	<code>\[GothicD]</code>	d
<code>\[FilledDiamond]</code>	◆	<code>\[GothicE]</code>	e
<code>\[FilledDownTriangle]</code>	▼	<code>\[GothicEight]</code>	8
<code>\[FilledLeftTriangle]</code>	◀	<code>\[GothicF]</code>	f
<code>\[FilledRectangle]</code>	■	<code>\[GothicFive]</code>	5
<code>\[FilledRightTriangle]</code>	▶	<code>\[GothicFour]</code>	4
<code>\[FilledSmallCircle]</code>	●	<code>\[GothicG]</code>	g
<code>\[FilledSmallSquare]</code>	■	<code>\[GothicH]</code>	h

<code>\[GothicI]</code>	i	<code>\[Implies]</code>	$\Rightarrow$
<code>\[GothicJ]</code>	j	<code>\[Infinity]</code>	$\infty$
<code>\[GothicK]</code>	k	<code>\[Integral]</code>	$\int$
<code>\[GothicL]</code>	l	<code>\[Intersection]</code>	$\cap$
<code>\[GothicM]</code>	m	<code>\[Iota]</code>	$\iota$
<code>\[GothicN]</code>	n	<code>\[Jupiter]</code>	$\Jupiter$
<code>\[GothicNine]</code>	9	<code>\[Kappa]</code>	$\kappa$
<code>\[GothicO]</code>	o	<code>\[KernelIcon]</code>	
<code>\[GothicOne]</code>	1	<code>\[KeyBar]</code>	$\bar{\cdot}$
<code>\[GothicP]</code>	p	<code>\[Koppa]</code>	$\varphi$
<code>\[GothicQ]</code>	q	<code>\[Lambda]</code>	$\lambda$
<code>\[GothicR]</code>	r	<code>\[LastPage]</code>	$\P$
<code>\[GothicS]</code>	s	<code>\[LeftAngleBracket]</code>	$\langle$
<code>\[GothicSeven]</code>	7	<code>\[LeftArrow]</code>	$\leftarrow$
<code>\[GothicSix]</code>	6	<code>\[LeftArrowBar]</code>	$\bar{\leftarrow}$
<code>\[GothicT]</code>	t	<code>\[LeftArrowRightArrow]</code>	$\Leftrightarrow$
<code>\[GothicThree]</code>	3	<code>\[LeftBracketingBar]</code>	$ $
<code>\[GothicTwo]</code>	2	<code>\[LeftCeiling]</code>	$\lceil$
<code>\[GothicU]</code>	u	<code>\[LeftDoubleBracket]</code>	$\llbracket$
<code>\[GothicV]</code>	v	<code>\[LeftDoubleBracketingBar]</code>	$\llbracket$
<code>\[GothicW]</code>	w	<code>\[LeftDownTeeVector]</code>	$\lrcorner$
<code>\[GothicX]</code>	x	<code>\[LeftDownVector]</code>	$\searrow$
<code>\[GothicY]</code>	y	<code>\[LeftDownVectorBar]</code>	$\bar{\searrow}$
<code>\[GothicZ]</code>	z	<code>\[LeftFloor]</code>	$\lfloor$
<code>\[GothicZero]</code>	0	<code>\[LeftGuillemet]</code>	$\llcorner$
<code>\[GrayCircle]</code>		<code>\[LeftModified]</code>	$\lmod$
<code>\[GraySquare]</code>		<code>\[LeftPointer]</code>	$\blacktriangleleft$
<code>\[GreaterEqual]</code>	$\geq$	<code>\[LeftRightArrow]</code>	$\leftrightarrow$
<code>\[GreaterEqualLess]</code>	$\gtrless$	<code>\[LeftRightVector]</code>	$\rightleftarrows$
<code>\[GreaterFullEqual]</code>	$\equiv$	<code>\[LeftSkeleton]</code>	$\llcorner$
<code>\[GreaterGreater]</code>	$\gg$	<code>\[LeftTee]</code>	$\dashv$
<code>\[GreaterLess]</code>	$\gtrsim$	<code>\[LeftTeeArrow]</code>	$\bar{\dashv}$
<code>\[GreaterSlantEqual]</code>	$\gtrapprox$	<code>\[LeftTeeVector]</code>	$\bar{\dashv}$
<code>\[GreaterTilde]</code>	$\gtrsim$	<code>\[LeftTriangle]</code>	$\triangleleft$
<code>\[Hacek]</code>	$\checkmark$	<code>\[LeftTriangleBar]</code>	$\triangleleft$
<code>\[HappySmiley]</code>		<code>\[LeftTriangleEqual]</code>	$\trianglelefteq$
<code>\[HBar]</code>	$\hbar$	<code>\[LeftUpDownVector]</code>	$\Uparrow$
<code>\[HeartSuit]</code>	$\heartsuit$	<code>\[LeftUpTeeVector]</code>	$\Uparrow$
<code>\[HermitianConjugate]</code>	$\mathcal{H}$	<code>\[LeftUpVector]</code>	$\Uparrow$
<code>\[HorizontalLine]</code>	$-$	<code>\[LeftUpVectorBar]</code>	$\bar{\Uparrow}$
<code>\[HumpDownHump]</code>	$\approx$	<code>\[LeftVector]</code>	$\leftarrow$
<code>\[HumpEqual]</code>	$\approx$	<code>\[LeftVectorBar]</code>	$\bar{\leftarrow}$
<code>\[Hyphen]</code>	$-$	<code>\[LessEqual]</code>	$\leq$
<code>\[IAcute]</code>	$\acute{a}$	<code>\[LessEqualGreater]</code>	$\lessgtr$
<code>\[ICup]</code>	$\grave{a}$	<code>\[LessFullEqual]</code>	$\equiv$
<code>\[IDoubleDot]</code>	$\ddot{a}$	<code>\[LessGreater]</code>	$\lessgtr$
<code>\[IGrave]</code>	$\grave{a}$	<code>\[LessLess]</code>	$\ll$
<code>\[IHat]</code>	$\hat{a}$	<code>\[LessSlantEqual]</code>	$\lessapprox$
<code>\[ImaginaryI]</code>	$i$	<code>\[LessTilde]</code>	$\lesssim$
<code>\[ImaginaryJ]</code>	$j$	<code>\[LightBulb]</code>	$\varphi$

<code>\[LongDash]</code>	—	<code>\[NotLeftTriangleBar]</code>	⊄
<code>\[LongEqual]</code>	=	<code>\[NotLeftTriangleEqual]</code>	≠
<code>\[LongLeftArrow]</code>	←	<code>\[NotLess]</code>	≠
<code>\[LongLeftRightArrow]</code>	↔	<code>\[NotLessEqual]</code>	≠
<code>\[LongRightArrow]</code>	→	<code>\[NotLessFullEqual]</code>	≠
<code>\[LowerLeftArrow]</code>	↙	<code>\[NotLessGreater]</code>	≠
<code>\[LowerRightArrow]</code>	↘	<code>\[NotLessLess]</code>	≠
<code>\[LSlash]</code>	/	<code>\[NotLessSlantEqual]</code>	≠
<code>\[Mars]</code>	♂	<code>\[NotLessTilde]</code>	≠
<code>\[MathematicaIcon]</code>	☼	<code>\[NotNestedGreaterGreater]</code>	≠
<code>\[MeasuredAngle]</code>	∠	<code>\[NotNestedLessLess]</code>	≠
<code>\[MediumSpace]</code>		<code>\[NotPrecedes]</code>	≠
<code>\[Mercury]</code>	♀	<code>\[NotPrecedesEqual]</code>	≠
<code>\[Mho]</code>	Ω	<code>\[NotPrecedesSlantEqual]</code>	≠
<code>\[Micro]</code>	μ	<code>\[NotPrecedesTilde]</code>	≠
<code>\[MinusPlus]</code>	±	<code>\[NotReverseElement]</code>	∉
<code>\[Mod1Key]</code>	<small>MOD1</small>	<code>\[NotRightTriangle]</code>	⊈
<code>\[Mod2Key]</code>	<small>MOD2</small>	<code>\[NotRightTriangleBar]</code>	⊉
<code>\[Mu]</code>	μ	<code>\[NotRightTriangleEqual]</code>	≠
<code>\[Nand]</code>	⋈	<code>\[NotSquareSubset]</code>	⊈
<code>\[Natural]</code>	ℕ	<code>\[NotSquareSubsetEqual]</code>	⊉
<code>\[NegativeMediumSpace]</code>		<code>\[NotSquareSuperset]</code>	⊉
<code>\[NegativeThickSpace]</code>		<code>\[NotSquareSupersetEqual]</code>	⊈
<code>\[NegativeThinSpace]</code>		<code>\[NotSubset]</code>	⊈
<code>\[NegativeVeryThinSpace]</code>		<code>\[NotSubsetEqual]</code>	⊉
<code>\[Neptune]</code>	♆	<code>\[NotSucceeds]</code>	≠
<code>\[NestedGreaterGreater]</code>	≫	<code>\[NotSucceedsEqual]</code>	≠
<code>\[NestedLessLess]</code>	≪	<code>\[NotSucceedsSlantEqual]</code>	≠
<code>\[NeutralSmiley]</code>	☺	<code>\[NotSucceedsTilde]</code>	≠
<code>\[NoBreak]</code>		<code>\[NotSuperset]</code>	⊈
<code>\[NonBreakingSpace]</code>		<code>\[NotSupersetEqual]</code>	⊉
<code>\[Nor]</code>	∨	<code>\[NotTilde]</code>	≠
<code>\[Not]</code>	¬	<code>\[NotTildeEqual]</code>	≠
<code>\[NotCongruent]</code>	≢	<code>\[NotTildeFullEqual]</code>	≠
<code>\[NotCupCap]</code>	≠	<code>\[NotTildeTilde]</code>	≠
<code>\[NotDoubleVerticalBar]</code>	⋈	<code>\[NotVerticalBar]</code>	⋈
<code>\[NotElement]</code>	∉	<code>\[NTilde]</code>	ñ
<code>\[NotEqual]</code>	≠	<code>\[Nu]</code>	ν
<code>\[NotEqualTilde]</code>	≠	<code>\[NumberSign]</code>	#
<code>\[NotExists]</code>	∄	<code>\[OAcute]</code>	ó
<code>\[NotGreater]</code>	≠	<code>\[ODoubleAcute]</code>	ö
<code>\[NotGreaterEqual]</code>	≠	<code>\[ODoubleDot]</code>	ö
<code>\[NotGreaterFullEqual]</code>	≠	<code>\[OGrave]</code>	ò
<code>\[NotGreaterGreater]</code>	≠	<code>\[OHat]</code>	ô
<code>\[NotGreaterLess]</code>	≠	<code>\[Omega]</code>	ω
<code>\[NotGreaterSlantEqual]</code>	≠	<code>\[Omicron]</code>	ο
<code>\[NotGreaterTilde]</code>	≠	<code>\[OpenCurlyDoubleQuote]</code>	“
<code>\[NotHumpDownHump]</code>	≠	<code>\[OpenCurlyQuote]</code>	‘
<code>\[NotHumpEqual]</code>	≠	<code>\[OptionKey]</code>	<small>OPTION</small>
<code>\[NotLeftTriangle]</code>	⊄	<code>\[Or]</code>	∨

<code>\[OSlash]</code>	ø	<code>\[RawSlash]</code>	/
<code>\[OTilde]</code>	õ	<code>\[RawSpace]</code>	
<code>\[OverBrace]</code>	⏟	<code>\[RawStar]</code>	*
<code>\[OverBracket]</code>	⏟	<code>\[RawTilde]</code>	~
<code>\[OverParenthesis]</code>	⏟	<code>\[RawUnderscore]</code>	_
<code>\[Paragraph]</code>	¶	<code>\[RawVerticalBar]</code>	
<code>\[PartialD]</code>	∂	<code>\[RawWedge]</code>	^
<code>\[Phi]</code>	φ	<code>\[RegisteredTrademark]</code>	®
<code>\[Pi]</code>	π	<code>\[ReturnIndicator]</code>	↵
<code>\[Piecewise]</code>	{	<code>\[ReturnKey]</code>	⏎
<code>\[Placeholder]</code>	□	<code>\[ReverseDoublePrime]</code>	″
<code>\[PlusMinus]</code>	±	<code>\[ReverseElement]</code>	⊃
<code>\[Pluto]</code>	♅	<code>\[ReverseEquilibrium]</code>	⇌
<code>\[Precedes]</code>	<	<code>\[ReversePrime]</code>	′
<code>\[PrecedesEqual]</code>	≦	<code>\[ReverseUpEquilibrium]</code>	⇌
<code>\[PrecedesSlantEqual]</code>	≲	<code>\[Rho]</code>	ρ
<code>\[PrecedesTilde]</code>	≍	<code>\[RightAngle]</code>	∟
<code>\[Prime]</code>	′	<code>\[RightAngleBracket]</code>	⟩
<code>\[Product]</code>	∏	<code>\[RightArrow]</code>	→
<code>\[Proportion]</code>	::	<code>\[RightArrowBar]</code>	→
<code>\[Proportional]</code>	∝	<code>\[RightArrowLeftArrow]</code>	↔
<code>\[Psi]</code>	ψ	<code>\[RightBracketingBar]</code>	
<code>\[QuarterNote]</code>	♪	<code>\[RightCeiling]</code>	⌈
<code>\[RawAmpersand]</code>	&	<code>\[RightDoubleBracket]</code>	⌋
<code>\[RawAt]</code>	@	<code>\[RightDoubleBracketingBar]</code>	⌋
<code>\[RawBackquote]</code>	`	<code>\[RightDownTeeVector]</code>	⌞
<code>\[RawBackslash]</code>	\	<code>\[RightDownVector]</code>	↓
<code>\[RawColon]</code>	:	<code>\[RightDownVectorBar]</code>	⌞
<code>\[RawComma]</code>	,	<code>\[RightFloor]</code>	⌋
<code>\[RawDash]</code>	–	<code>\[RightGuillemet]</code>	»
<code>\[RawDollar]</code>	\$	<code>\[RightModified]</code>	]
<code>\[RawDot]</code>	.	<code>\[RightPointer]</code>	▸
<code>\[RawDoubleQuote]</code>	"	<code>\[RightSkeleton]</code>	≫
<code>\[RawEqual]</code>	=	<code>\[RightTee]</code>	⊢
<code>\[RawEscape]</code>		<code>\[RightTeeArrow]</code>	⊢
<code>\[RawExclamation]</code>	!	<code>\[RightTeeVector]</code>	⊢
<code>\[RawGreater]</code>	>	<code>\[RightTriangle]</code>	▷
<code>\[RawLeftBrace]</code>	{	<code>\[RightTriangleBar]</code>	▷
<code>\[RawLeftBracket]</code>	[	<code>\[RightTriangleEqual]</code>	≅
<code>\[RawLeftParenthesis]</code>	(	<code>\[RightUpDownVector]</code>	↕
<code>\[RawLess]</code>	<	<code>\[RightUpTeeVector]</code>	⌞
<code>\[RawNumberSign]</code>	#	<code>\[RightUpVector]</code>	↑
<code>\[RawPercent]</code>	%	<code>\[RightUpVectorBar]</code>	↑
<code>\[RawPlus]</code>	+	<code>\[RightVector]</code>	→
<code>\[RawQuestion]</code>	?	<code>\[RightVectorBar]</code>	→
<code>\[RawQuote]</code>	'	<code>\[RoundImplies]</code>	⇒
<code>\[RawRightBrace]</code>	}	<code>\[RoundSpaceIndicator]</code>	⋅
<code>\[RawRightBracket]</code>	]	<code>\[Rule]</code>	→
<code>\[RawRightParenthesis]</code>	)	<code>\[RuleDelayed]</code>	⇒
<code>\[RawSemicolon]</code>	;	<code>\[SadSmiley]</code>	☹

<code>\[Sampi]</code>	$\mathfrak{s}$	<code>\[ScriptP]</code>	$p$
<code>\[Saturn]</code>	$\mathfrak{h}$	<code>\[ScriptQ]</code>	$q$
<code>\[ScriptA]</code>	$a$	<code>\[ScriptR]</code>	$r$
<code>\[ScriptB]</code>	$b$	<code>\[ScriptS]</code>	$s$
<code>\[ScriptC]</code>	$c$	<code>\[ScriptSeven]</code>	7
<code>\[ScriptCapitalA]</code>	$\mathcal{A}$	<code>\[ScriptSix]</code>	6
<code>\[ScriptCapitalB]</code>	$\mathcal{B}$	<code>\[ScriptT]</code>	$t$
<code>\[ScriptCapitalC]</code>	$\mathcal{C}$	<code>\[ScriptThree]</code>	3
<code>\[ScriptCapitalD]</code>	$\mathcal{D}$	<code>\[ScriptTwo]</code>	2
<code>\[ScriptCapitalE]</code>	$\mathcal{E}$	<code>\[ScriptU]</code>	$u$
<code>\[ScriptCapitalF]</code>	$\mathcal{F}$	<code>\[ScriptV]</code>	$v$
<code>\[ScriptCapitalG]</code>	$\mathcal{G}$	<code>\[ScriptW]</code>	$w$
<code>\[ScriptCapitalH]</code>	$\mathcal{H}$	<code>\[ScriptX]</code>	$x$
<code>\[ScriptCapitalI]</code>	$\mathcal{I}$	<code>\[ScriptY]</code>	$y$
<code>\[ScriptCapitalJ]</code>	$\mathcal{J}$	<code>\[ScriptZ]</code>	$z$
<code>\[ScriptCapitalK]</code>	$\mathcal{K}$	<code>\[ScriptZero]</code>	0
<code>\[ScriptCapitalL]</code>	$\mathcal{L}$	<code>\[Section]</code>	§
<code>\[ScriptCapitalM]</code>	$\mathcal{M}$	<code>\[SelectionPlaceholder]</code>	■
<code>\[ScriptCapitalN]</code>	$\mathcal{N}$	<code>\[SHacek]</code>	š
<code>\[ScriptCapitalO]</code>	$\mathcal{O}$	<code>\[Sharp]</code>	#
<code>\[ScriptCapitalP]</code>	$\mathcal{P}$	<code>\[ShiftKey]</code>	SHIFT
<code>\[ScriptCapitalQ]</code>	$\mathcal{Q}$	<code>\[ShortDownArrow]</code>	↓
<code>\[ScriptCapitalR]</code>	$\mathcal{R}$	<code>\[ShortLeftArrow]</code>	←
<code>\[ScriptCapitalS]</code>	$\mathcal{S}$	<code>\[ShortRightArrow]</code>	→
<code>\[ScriptCapitalT]</code>	$\mathcal{T}$	<code>\[ShortUpArrow]</code>	↑
<code>\[ScriptCapitalU]</code>	$\mathcal{U}$	<code>\[Sigma]</code>	$\sigma$
<code>\[ScriptCapitalV]</code>	$\mathcal{V}$	<code>\[SixPointedStar]</code>	★
<code>\[ScriptCapitalW]</code>	$\mathcal{W}$	<code>\[SkeletonIndicator]</code>	-
<code>\[ScriptCapitalX]</code>	$\mathcal{X}$	<code>\[SmallCircle]</code>	◦
<code>\[ScriptCapitalY]</code>	$\mathcal{Y}$	<code>\[SpaceIndicator]</code>	␣
<code>\[ScriptCapitalZ]</code>	$\mathcal{Z}$	<code>\[SpaceKey]</code>	SPACE
<code>\[ScriptD]</code>	$d$	<code>\[SpadeSuit]</code>	♠
<code>\[ScriptDotlessI]</code>	$i$	<code>\[SpanFromAbove]</code>	⋮
<code>\[ScriptDotlessJ]</code>	$j$	<code>\[SpanFromBoth]</code>	⋮
<code>\[ScriptE]</code>	$e$	<code>\[SpanFromLeft]</code>	⋯
<code>\[ScriptEight]</code>	8	<code>\[SphericalAngle]</code>	∠
<code>\[ScriptF]</code>	$f$	<code>\[Spooky]</code>	☠
<code>\[ScriptFive]</code>	5	<code>\[Sqrt]</code>	√
<code>\[ScriptFour]</code>	4	<code>\[Square]</code>	□
<code>\[ScriptG]</code>	$g$	<code>\[SquareIntersection]</code>	⊠
<code>\[ScriptH]</code>	$h$	<code>\[SquareSubset]</code>	⊐
<code>\[ScriptI]</code>	$i$	<code>\[SquareSubsetEqual]</code>	⊑
<code>\[ScriptJ]</code>	$j$	<code>\[SquareSuperset]</code>	⊒
<code>\[ScriptK]</code>	$k$	<code>\[SquareSupersetEqual]</code>	⊓
<code>\[ScriptL]</code>	$\ell$	<code>\[SquareUnion]</code>	⊞
<code>\[ScriptM]</code>	$m$	<code>\[Star]</code>	*
<code>\[ScriptN]</code>	$n$	<code>\[StepperDown]</code>	↓
<code>\[ScriptNine]</code>	9	<code>\[StepperLeft]</code>	←
<code>\[ScriptO]</code>	$o$	<code>\[StepperRight]</code>	→
<code>\[ScriptOne]</code>	1	<code>\[StepperUp]</code>	↑



<code>\[StepperUp]</code>	↑	<code>\[UnderParenthesis]</code>	⏟
<code>\[Sterling]</code>	£	<code>\[Union]</code>	∪
<code>\[Stigma]</code>	ς	<code>\[UnionPlus]</code>	⊕
<code>\[Subset]</code>	⊂	<code>\[UnknownGlyph]</code>	■
<code>\[SubsetEqual]</code>	⊆	<code>\[UpArrow]</code>	↑
<code>\[Succeeds]</code>	⋗	<code>\[UpArrowBar]</code>	↑̄
<code>\[SucceedsEqual]</code>	⋗̄	<code>\[UpArrowDownArrow]</code>	↕
<code>\[SucceedsSlantEqual]</code>	⋗̸	<code>\[UpDownArrow]</code>	↕
<code>\[SucceedsTilde]</code>	⋗̃	<code>\[UpEquilibrium]</code>	↕̄
<code>\[SuchThat]</code>	∋	<code>\[UpperLeftArrow]</code>	↖
<code>\[Sum]</code>	∑	<code>\[UpperRightArrow]</code>	↗
<code>\[Superset]</code>	⊃	<code>\[UpPointer]</code>	⬆
<code>\[SupersetEqual]</code>	⊇	<code>\[Upsilon]</code>	υ
<code>\[SZ]</code>	ß	<code>\[UpTee]</code>	⊥
<code>\[TabKey]</code>	<code>\[TAB]</code>	<code>\[UpTeeArrow]</code>	⬆
<code>\[Tau]</code>	τ	<code>\[Uranus]</code>	♅
<code>\[Therefore]</code>	∴	<code>\[Vee]</code>	∨
<code>\[Theta]</code>	θ	<code>\[Venus]</code>	♀
<code>\[ThickSpace]</code>		<code>\[VerticalBar]</code>	
<code>\[ThinSpace]</code>		<code>\[VerticalEllipsis]</code>	⋮
<code>\[Thorn]</code>	þ	<code>\[VerticalLine]</code>	
<code>\[Tilde]</code>	~	<code>\[VerticalSeparator]</code>	
<code>\[TildeEqual]</code>	≈	<code>\[VerticalTilde]</code>	⋮̃
<code>\[TildeFullEqual]</code>	≅	<code>\[VeryThinSpace]</code>	
<code>\[TildeTilde]</code>	≈	<code>\[Villa]</code>	□
<code>\[Times]</code>	×	<code>\[WarningSign]</code>	⚠
<code>\[Trademark]</code>	™	<code>\[WatchIcon]</code>	🕒
<code>\[Transpose]</code>	ᵀ	<code>\[Wedge]</code>	^
<code>\[TripleDot]</code>	...	<code>\[WeierstrassP]</code>	℘
<code>\[UAcute]</code>	ú	<code>\[Wolf]</code>	🐺
<code>\[UDoubleAcute]</code>	ü	<code>\[Xi]</code>	ξ
<code>\[UDoubleDot]</code>	ü	<code>\[Xor]</code>	⊕
<code>\[UGrave]</code>	ù	<code>\[YAcute]</code>	ý
<code>\[UHat]</code>	û	<code>\[YDoubleDot]</code>	ÿ
<code>\[UnderBrace]</code>	⏟	<code>\[Yen]</code>	¥
<code>\[UnderBracket]</code>	⏟	<code>\[Zeta]</code>	ζ

## ■ TECLAS VIRTUAIS E PALETAS

No *Mathematica* é possível criar "Teclas Virtuais" (ing. Buttons) para automatizar a introdução de expressões ou criar funcionalidades adicionais no FrontEnd. A função mais frequente numa tecla virtual é a de inserir, na posição do cursor de texto, o símbolo ou a função codificada nessa tecla virtual. Por exemplo: `∇ ×  $\vec{B}$`  quando pressionado, insere  $\nabla \times \vec{B}$  na posição e no Notebook onde se encontra o cursor. Outro exemplo é uma tecla virtual cuja função é abrir um outro Notebook ou implementar um Hyperlink para abrir um endereço URL num Browser adequado.  $\varepsilon$

## □ UTILIZAÇÃO DE MARCADORES (ING. PLACEHOLDER)

Note-se que sempre que aparece um quadrado preto ■ numa tecla virtual isto representa o símbolo \[SelectionPlaceholder], significando que o que quer que esteja actualmente seleccionado no Notebook quando a tecla virtual for pressionada será substituído na expressão que a tecla virtual implementa na posição do quadrado preto. Um quadrado branco □ representa o símbolo \[Placeholder] que apenas indica as posições que é necessário preencher para completar a expressão implementada pela tecla virtual. Sempre que estiver a editar uma expressão que inclui Placeholders como ■ ou □ é possível passar rapidamente para o próximo ou o anterior usando a tecla TAB ou SHIFT TAB.

Por exemplo, seleccionando o denominador do integrando na expressão seguinte

$$2 \int_{-\infty}^{+\infty} \frac{1}{\blacksquare x^2 - 1} dx$$

e pressionando a tecla virtual  $\sqrt{\square}$  disponível numa paleta ou Notebook aberto obtém-se

$$2 \int_{-\infty}^{+\infty} \frac{1}{\sqrt{\square} x^2 - 1} dx$$

ou seja, a expressão seleccionada  $x^2 - 1$  é substituída no lugar do SelectionPlaceholder ■ na expressão  $\sqrt{\square}$  implementada pela tecla virtual. O resultado é colocado no lugar da selecção inicial. A seguir o Placeholder □ deve ser seleccionado com TAB (ou com o cursor) e preenchido de forma a completar a expressão:

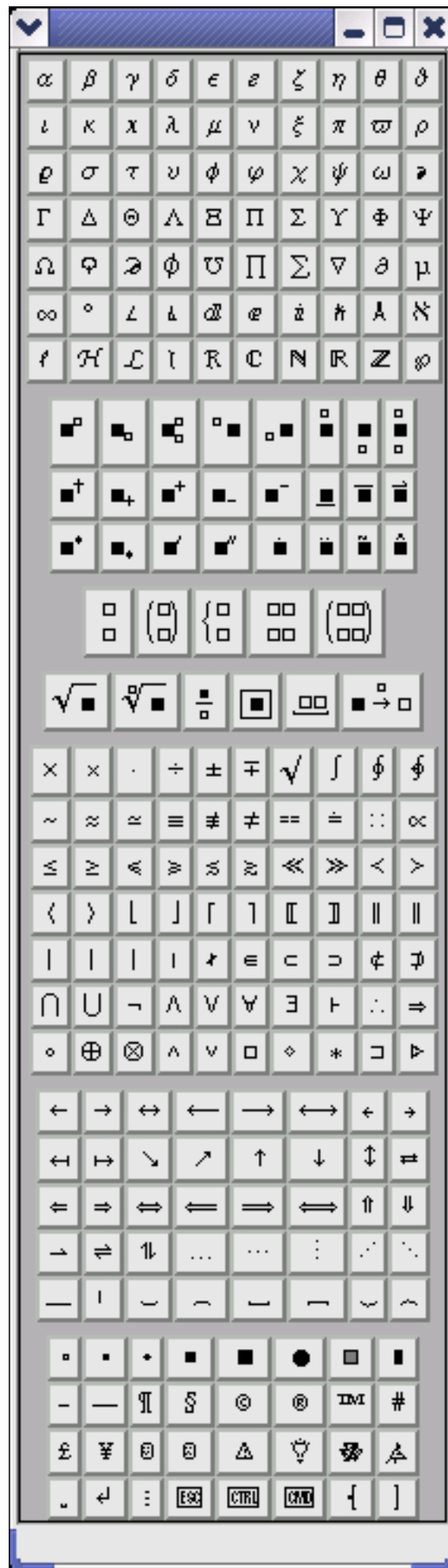
$$2 \int_{-\infty}^{+\infty} \frac{1}{\sqrt[3]{x^2 - 1}} dx$$

## □ UTILIZAÇÃO DE PALETAS

Paletas são Notebooks especiais que agrupam por assunto colecções destas teclas virtuais. No menu File ▶ Palettes encontram-se várias paletas nativas que podem ser invocadas para auxiliar a introdução de expressões numa sessão do *Mathematica*. Na figura seguinte mostram-se três destas paletas intituladas CompleteCharacters, que agrupa por tópicos a colecção completa de caracteres especiais do *Mathematica*, BasicTypesetting, que inclui os elementos mais frequentes para a construção de texto científico, e BasicInput que inclui as expressões mais frequentemente utilizadas para escrever fórmulas que serão avaliadas pelo Kernel.



CompleteCharacters



BasicTypesetting



BasicInput

□ COMO CRIAR UMA TECLA VIRTUAL:

- Utilize o menu **Input** ▸ **Create Button** ▸ **Paste** para criar um molde como □.
- Seleccione e escreva sobre o quadrado vazio □ que aparece dentro da tecla virtual.
- Seleccione a tecla virtual inteira e torne-a activa usando o menu **Cell** ▸ **Cell Properties** ▸ **Cell Active** .

Quando pressionada uma tecla virtual executa uma função especificada pela opção

$$\text{ButtonFunction} \rightarrow \text{Function}[\{v_1, v_2\}, \text{expr}[v_1, v_2]]$$

Os argumentos  $v_1$  e  $v_2$  são sempre os valores de **ButtonSource** e **ButtonData**. Este último é em geral escondido, excepto quando invocado como segundo argumento pela função que a tecla virtual activa. Quando **ButtonSource** é omissso toma o valor de **ButtonData**. Outra forma é usar

$$\text{ButtonSource} \rightarrow \text{ButtonContents}$$

que é a expressão visível dentro da tecla virtual (i.e. o primeiro argumento da **ButtonBox**).

```
NotebookWrite[SelectedNotebook[],
  Cell[BoxData[
    ButtonBox["Hello",
      Active -> True,
      ButtonEvaluator -> Automatic,
      ButtonSource -> ButtonContents,
      ButtonData -> "GoodBye",
      ButtonFunction ->
        Function[{v1, v2}, Print["ButtonContents = ", v1, "\n", "ButtonData = ", v2]]],
    "SmallText"]]
```

Hello

```
ButtonContents = BoxData[Hello]
ButtonData = GoodBye
```

Teclas virtuais com um **ButtonStyle** nativo **Paste**, **Evaluate**, **Hyperlink**, ... não necessitam que se explicita a **ButtonFunction** que recebem por inerência da definição do estilo (vide o item **Palette Styles** na **StyleSheet**).

Por exemplo, a tecla virtual **hyperlink** cujo código é

```
ButtonBox["hyperlink", ButtonData -> "Bot\[/OTilde]es e Paletas",
  Active -> True, ButtonStyle -> "Hyperlink"]
```

efectivamente implementa a definição de **ButtonStyle** → **"Hyperlink"** a qual inclui

$$\text{ButtonFunction} \rightarrow (\text{FrontEndExecute}[\{\text{NotebookLocate}[\#2]\}] \&)$$

ou o que é o mesmo

$$\text{ButtonFunction} \rightarrow \text{Function}[\{v_1, v_2\}, \text{NotebookLocate}[v_2]]$$

onde  $v_2$  é o valor de **ButtonData** → " Botões e Paletas " que é a etiqueta da célula de título desta secção. Assim, pressionando esta tecla o **FrontEnd** localiza e selecciona essa célula.

Para criar os nossos próprios botões (i.e. um que não esteja associado a um estilo pré-definido na StyleSheet) basta especificar a **ButtonFunction**, quer directamente dentro da **ButtonBox** ou indirectamente através das opções **ButtonBoxOptions** que aparecem quando se abre a janela de preferências com a tecla seleccionada.

O seguinte exemplo implementa uma tecla virtual **Inverte e soma x** que substitui qualquer expressão seleccionada **expr** pela expressão  $\text{Hello} = x + \frac{1}{\text{expr}}$ . Adicionalmente imprime os valores de **v<sub>1</sub>** e **v<sub>2</sub>**, ou seja o **ButtonContents** e o **ButtonData**. Para a criar directamente através de uma expressão de **Input** escrever-se-ia:

```
NotebookApply[SelectedNotebook[],
  ButtonBox["Inverte e soma x",
    Active -> True,
    ButtonEvaluator -> Automatic,
    ButtonSource -> ButtonContents,
    ButtonData -> "Hello",
    ButtonFunction ->
      Function[{v1, v2}, NotebookApply[SelectedNotebook[], StringJoin[v2, "= x+ 1/■"];
        Print["ButtonContents = ", v1, "\n", "ButtonData = ", v2]]]]
```

**Inverte e soma x**

A expressão seleccionada


**a<sup>2</sup>**

quando a tecla **Inverte e soma x** é pressionada é substituída por

**Hello = x + 1 / a<sup>2</sup>**

```
ButtonContents = BoxData[RowBox[{Inverte, , e, , soma, , x}]]
ButtonData = Hello
```

#### □ COMO CRIAR PALETAS DE TECLAS VIRTUAIS

- Utilize o menu **Input** ▸ **Create Table / Matrix / Palette** ou a combinação de teclas **ALT** **SHIFT** **C** para abrir uma janela de opções para um objecto do tipo **GridBox**.
- Indique na janela que aparece quantas linhas e colunas deve ter a paleta e pressione a tecla **OK**. Obterá, por exemplo, .
- Edite cada tecla virtual da paleta (desformatando-a se necessário). Torne tudo activo usando o menu **Cell** ▸ **Cell Properties** ▸ **Cell Active**.

## ■ IMPORTAÇÃO E EXPORTAÇÃO DE DEFINIÇÕES

<<"file"	lê as definições que estão em "file".
!!"file"	mostra o conteúdo de "file" sem abrir ou avaliar.
Save["file", expr <sub>1</sub> , expr <sub>2</sub> , ...]	Guarda as definições completas de expr <sub>1</sub> , expr <sub>2</sub> , ..., em "file".
expr >> "file"	Escreve a definição de 'expr' em "file" (apagando o que aí estiver escrito!)
expr >>> "file"	Adiciona apenas a definição de 'expr' no fim de "file".

**Tabela 2**

```
SetAttributes[f, Listable];
```

```
f[x_] := x2 + log[x]
```

```
log[z_] := Log[z, 10]
```

```
Save["mydefs.m", f]
```

```
!! mydefs.m
```

```
Attributes[f] = {Listable}
f[x_] := x^2 + log[x]
log[z_] := Log[z, 10]
```

```
" any expression" >>> mydefs.m
```

```
!! mydefs.m
```

```
Attributes[f] = {Listable}
f[x_] := x^2 + log[x]
log[z_] := Log[z, 10]
" any expression"
```

```
Definition[f] >> mydefs.m
```

```
!! mydefs.m
```

```
Attributes[f] = {Listable}
f[x_] := x^2 + log[x]
```

## ■ IMPORTAÇÃO DE MÓDULOS ADICIONAIS (PACKAGES E CONTEXTOS)

O Mathematica dispõe de uma grande variedade de módulos adicionais (ing. Packages) que podem ser importados em cada sessão para estender as funcionalidades nativas do Kernel. Estes módulos adicionais são documentos de texto com nome do tipo SomeName.m que contêm definições de funções associadas a um Contexto particular. Por exemplo, o módulo Graphics/ContourPlot3D.m define mais funções no Contexto Graphics que permitem ao Mathematica fazer gráficos 3D com informação topográfica por exemplo. Uma listagem do directório AddOns/StandardPackages mostra os tipos de módulos que é possível incorporar numa sessão. (A variável de sistema \$TopDirectory indica o directório onde se encontra a instalação do Mathematica.)

```
SetDirectory[StringJoin @@ {$TopDirectory, "/AddOns", "/StandardPackages"}];
FileNames[] // TableForm
```

```
Algebra
Calculus
DiscreteMath
Geometry
Graphics
LinearAlgebra
Miscellaneous
NumberTheory
NumericalMath
StartUp
Statistics
Utilities
```

Desta lista, a entrada 5 indica o directório Graphics que contém vários módulos.

```
FileNames[StringJoin[{%[[5]], "/*.m"}]] // TableForm
```

Graphics/Animation.m  
 Graphics/ArgColors.m  
 Graphics/Arrow.m  
 Graphics/Colors.m  
 Graphics/ComplexMap.m  
 Graphics/ContourPlot3D.m  
 Graphics/FilledPlot.m  
 Graphics/Graphics3D.m  
 Graphics/Graphics.m  
 Graphics/ImplicitPlot.m  
 Graphics/InequalityGraphics.m  
 Graphics/Legend.m  
 Graphics/MultipleListPlot.m  
 Graphics/ParametricPlot3D.m  
 Graphics/PlotField3D.m  
 Graphics/PlotField.m  
 Graphics/Polyhedra.m  
 Graphics/Shapes.m  
 Graphics/Spline.m  
 Graphics/SurfaceOfRevolution.m  
 Graphics/ThreeScript.m

□ GET (<<), NEEDS

Para importar as definições do módulo Graphics/ContourPlot3D.m pode-se usar qualquer das formas:

```
Get["Graphics/ContourPlot3D.m"]
```

que é o mesmo que fazer

```
<< Graphics`ContourPlot3D`
```

e importam sempre o módulo **Graphics`ContourPlot3D`** definido no documento ASCII **ContourPlot3D.m**. A função **Needs** por outro lado verifica primeiro se o módulo já foi importado e só o faz se necessário.

```
Needs["Graphics`ContourPlot3D`"]
```

O módulo importado define as seguintes funções:

```
Names["Graphics`ContourPlot3D`*"]
```

```
{ContourPlot3D, ListContourPlot3D}
```

Estas funções podem agora ser usadas como funções nativas do *Mathematica*. Informação sobre a sua utilização é também carregada com o módulo:



## ? ContourPlot3D

ContourPlot3D[ fun, {x, xmin, xmax}, {y, ymin, ymax}, {z, zmin, zmax}] plots the surface implicitly defined by fun[ x, y, z] == 0. Setting Contours -> {val1, val2,...} will plot the surfaces for values val1, val2, etc. More...

## ? "ListContourPlot3D"

ListContourPlot3D[ data, opts] takes a three-dimensional data set interpreted as a representation of a function fun[ x, y, z], where the ranges of x, y, and z are set by the MeshRange option. It then plots the surface implicitly defined by fun[x, y, z] == 0. Setting Contours -> {val1, val2,...} will plot the surfaces for values val1, val2, etc. More...

### □ REMOVE[SYMBOL]

Note que se usar o nome da função **ContourPlot3D** antes de importar o módulo **Graphics/ContourPlot3D.m**, o Kernel memoriza o nome **ContourPlot3D** no contexto **Global**, que é o primeiro a ser procurado para encontrar definições associadas a símbolos. Assim, ao importar posteriormente o módulo que implementa um símbolo com o mesmo nome **ContourPlot3D** mas num contexto diferente **Graphics'ContourPlot3D**, irá receber uma mensagem de erro dizendo que o símbolo **Global'ContourPlot3D** mascara a definição associada ao símbolo **Graphics'ContourPlot3D'ContourPlot3D**. Para evitar este problema é necessário remover o símbolo **ContourPlot3D** com o prefixo **Global** usando a função **Remove[symbol]** antes de importar o módulo, ou **Remove[Global'symbol]** depois deste já ter sido importado.

## Remove[Global'ContourPlot3D]

Em qualquer altura a variável de sistema \$Packages indica quais os módulos importados na sessão actual do Kernel.

## \$Packages // TableForm

```
System'
Global'
Utilities'Notation'
Graphics'ContourPlot3D'
Utilities'FilterOptions'
```

### □ CONTEXTS

A função **Contexts[]** dá a lista de diferentes Contextos a serem utilizados pelo Kernel num dado momento. A lista de contextos do Mathematica permite ao Kernel encontrar funções definidas em módulos importados durante uma sessão sem que o utilizador tenha que invocar explicitamente **Contexto'NomeDeFunção[...]** para as usar. Isto também permite que não haja confusão entre variáveis com o mesmo nome definidas em módulos diferentes, porque internamente elas são representadas como **Contexto'Private'VariávelLocal** quando invocadas por uma função definida em **Contexto**. Para ver os nomes de funções e variáveis definidas num contexto já importado use a função **Names["Contexto'\*"]**

Contexts["Graphics\*"]

Graphics‘	Graphics‘Arrow‘Private‘	Graphics‘Shapes‘Private‘
Graphics‘Arrow‘	Graphics‘Shapes‘	

Names["Graphics‘Arrow‘\*"]

Absolute	HeadCenter	HeadScaling	HeadWidth	ZeroShape
Arrow	HeadLength	HeadShape	Relative	

A informação seguinte podia ser obtida simplesmente com **?? Arrow** mas usamos o nome com o contexto completo para mostrar que internamente **Arrow** realmente se encontra no contexto **Graphics‘Arrow‘**.

?? Graphics‘Arrow‘Arrow

Arrow[start, finish, (opts)] is a graphics primitive representing an arrow starting at start and ending at finish. More...

Options[Arrow] = {HeadScaling → Automatic, HeadLength → Automatic, HeadCenter → 1, HeadWidth → 0.5, HeadShape → Automatic, ZeroShape → Automatic}

## ■ FORMATAÇÃO DE EXPRESSÕES: INPUT E OUTPUT

As expressões introduzidas para cálculo no Mathematica podem ser formatadas de três formas diferentes:

Acção	Combinação
<i>InputForm</i>	CTRL SHIFT I
<i>StandardForm</i>	CTRL SHIFT T
<i>TraditionalForm</i>	CTRL SHIFT L

- (a) – **InputForm** é a notação unidimensional que se pode usar tanto no FrontEnd como numa sessão num terminal para introduzir expressões para avaliação. É muito parecida com a notação usada para representar operações em linguagem C.
- (b) – **StandardForm** é a notação bi-dimensional que é o padrão tanto para Input como para Output.
- (c) – **TraditionalForm** é uma representação bi-dimensional que se parece mais com a convenção matemática de escrita, em particular na utilização de parêntesis curvos para delimitar argumentos de funções (enquanto o *Mathematica* tem que usar sempre parêntesis rectos para esse efeito) e na conversão dos nomes de funções matemáticas convencionais para minúsculas (enquanto as funções nativas do *Mathematica* começam sempre por uma Maiúscula)

Não se deve usar **TraditionalForm** para escrever Input porque a interpretação dos parêntesis é ambígua! Por exemplo, em **TraditionalForm**  $x^2(a + b)$  é avaliado como  $(x^2)[a + b]$  e não como  $x^2*(a + b)$  como se pretendia.

Todo o **Output** do Mathematica vem em notação bi-dimensional StandardForm excepto se explicitamente formatado doutra maneira.

A função HoldForm permite manter qualquer expressão sem ser processada, tal como foi escrita, sem substituições de variáveis pré-definidas ou outras simplificações. Para retomar o processamento normal da expressão usa-se a função ReleaseHold.

A função FullForm mostra a estrutura completa duma expressão tal como é enviada para o Kernel. A função TreeForm faz o mesmo e adicionalmente separa os diversos níveis de composição de funções que definem a expressão.

A função nativa do Mathematica TeXForm permite gerar Output com a forma L<sup>A</sup>T<sub>E</sub>X do texto ou expressão em argumento. Note-se que para evitar que o Mathematica entenda uma frase como um produto de expressões e as reordene lexicograficamente é importante fornecer a TeXForm uma String ou uma expressão em HoldForm.

### TeX

Isto é o formato T<sub>E</sub>X de  $\int_0^{\infty} z^3 dz$  // HoldForm // TeXForm

Isto é o formato T<sub>E</sub>X de  $\int_0^{\infty} z^3 \, dz$

A seguinte tabela mostra uma lista de todas as formas diferentes possíveis de representar expressões no *Mathematica*, quer para efeitos de avaliação, debugging ou exportação para outras linguagens.

InputForm $\left[\text{Sin}\left[\frac{x^2}{4}\right]\right]$	Sin[x^2/4]
StandardForm $\left[\text{Sin}\left[\frac{x^2}{4}\right]\right]$	$\text{Sin}\left[\frac{x^2}{4}\right]$
TraditionalForm $\left[\text{Sin}\left[\frac{x^2}{4}\right]\right]$	$\sin\left(\frac{x^2}{4}\right)$
CForm $\left[\text{Sin}\left[\frac{x^2}{4}\right]\right]$	Sin (Power (x, 2)/4.)
FortranForm $\left[\text{Sin}\left[\frac{x^2}{4}\right]\right]$	$\sin\left(\frac{x^2}{4}\right)$
TeXForm $\left[\text{Sin}\left[\frac{x^2}{4}\right]\right]$	$\sin\left(\frac{x^2}{4}\right)$
FullForm $\left[\text{Sin}\left[\frac{x^2}{4}\right]\right]$	Sin[Times[Rational[1, 4], Power[x, 2]]]
TreeForm $\left[\text{Sin}\left[\frac{x^2}{4}\right]\right]$	$\text{Sin}\left[\begin{array}{c}   \\ \text{Times}\left[\begin{array}{c}   \\ \text{Rational}[1, 4] \end{array}, \begin{array}{c}   \\ \text{Power}[x, 2] \end{array} \right] \end{array}\right]$
OutputForm $\left[\text{Sin}\left[\frac{x^2}{4}\right]\right]$	$\frac{x^2}{4} \sin[--]$
MathMLForm $\left[\text{Sin}\left[\frac{x^2}{4}\right]\right]$	$\begin{array}{l} < \text{math} > \\ < \text{mrow} > \\ < \text{mi} > \sin < / \text{mi} > \\ < \text{mo} > \text{ \&\#8289; } < / \text{mo} > \\ < \text{mo} > ( < / \text{mo} > \\ < \text{mfrac} > \\ < \text{msup} > \\ < \text{mi} > x < / \text{mi} > \\ < \text{mn} > 2 < / \text{mn} > \\ < / \text{msup} > \\ < \text{mn} > 4 < / \text{mn} > \\ < / \text{mfrac} > \\ < \text{mo} > ) < / \text{mo} > \\ < / \text{mrow} > \\ < / \text{math} > \end{array}$

**Tabela 3**

No caso de números reais, as diferentes notações são:

AccountingForm $[e^{8.}, 9]$	2980.95799
EngineeringForm $[e^{8.}, 9]$	$2.98095799 \times 10^3$
ScientificForm $[e^{8.}, 9]$	$2.98095799 \times 10^3$
BaseForm $[e^{8.}, 16]$	ba4.f5 <sub>16</sub>
NumberForm $[e^{8.}, \text{NumberSigns} \rightarrow \{ "+", "-" \}, \text{NumberPoint} \rightarrow ", "]$	+2980,96
PaddedForm $[e^{8.}, \{12, 4\}, \text{NumberSigns} \rightarrow \{ "+", "-" \}, \text{SignPadding} \rightarrow \text{True}]$	+ 2980.9580

**Tabela 4**

No caso de listas também existem várias formatações possíveis:

TableForm[{{a, b}, {c, d}}]	$\begin{array}{cc} a & b \\ c & d \end{array}$
ColumnForm[{{a, b}, {c, d}}]	$\begin{array}{c} \{a, b\} \\ \{c, d\} \end{array}$
MatrixForm[{{a, b}, {c, d}}]	$\begin{pmatrix} a & b \\ c & d \end{pmatrix}$

**Tabela 5**

## ■ FORMATOS DE EXPORTAÇÃO: TEX, HTML, EPS

### □ EXPORTAÇÃO DE TEXTO FORMATADO

É possível exportar Notebooks inteiros para documentos HTML ou  $\text{T}_{\text{E}}\text{X}$ , usando por exemplo o menu **File** ▶ **Save As Special ...** ▶ **HTML** ou **TeX**.

Note-se que no caso de  $\text{T}_{\text{E}}\text{X}$ , torna-se necessário editar o documento gerado para incluir estilos auxiliares (*notebook.sty*, *notebook2e.sty*, *wrisym.sty*) que incluem definições para os símbolos e "fonts" usados pelo *Mathematica*. Adicionalmente é possível usar funções para o mesmo efeito:

HTMLSave["file.html", notebook, options] exporta todo o notebook para um documento file.html, gerando imagens .gif para tudo o que seja expressões bi-dimensionais e gráficos.

TeXSave["file.tex", notebook, options] exporta todo o notebook para um documento L<sup>A</sup>T<sub>E</sub>X.

#### □ EXPORTAÇÃO DE DADOS

Export["file.ext",expr, "format"] exporta os dados associados a expr para o documento file.ext onde a extensão .ext pode ser associada a um formato particular de entre muitos reconhecidos. Por exemplo .csv é comum entre folhas de cálculo, significando "valores separados por vírgulas", o que significa que exportando uma lista com várias colunas do Mathematica para um ficheiro file.csv este pode ser lido por qualquer programa que processe folhas de cálculo, como o Excel ou Gnumeric. Outros formatos podem ser especificados desde que reconhecidos pelo Mathematica (vide **Help** ▶ **Help Browser...** ▶ **Export**)

"CSV"	comma-separated value tabular data	(.csv)
"Lines"	list of strings to be placed on separate lines	
"List"	list of numbers or strings to be placed on separate lines	
"Table"	list of lists of numbers or strings to be placed in a 2 D array	(.dat)
"Text"	single string of ordinary characters	(.txt)
"TSV"	tab-separated value tabular data	(.tsv)
"UnicodeText"	single string of 16 -bit Unicode characters	
"Words"	list of strings to be separated by spaces	
"DIF"	Lotus Data Interchange Format	(.dif)
"FITS"	FITS astronomical data format	(.fit, .fits)
"HarwellBoeing"	Harwell-Boeing matrix format	
"HDF"	Hierarchical Data Format	(.hdf)
"HDF5"	HDF5 format	(.h5)
"MAT"	MAT matrix format	(.mat)
"MTX"	Matrix Market format	(.mtx)
"XLS"	Excel spreadsheet format	(.xls)

#### □ EXPORTAÇÃO DE GRÁFICOS

Gráficos podem igualmente ser exportados para documentos PostScript (.eps) seleccionando-os e fazendo **Edit** ▶ **Save Selection As** ▶ **EPS...** .

Outra alternativa é usar uma modificação da variável de sistema que indica para onde devem ser enviados os gráficos gerados pelo Kernel. Em geral esta variável \$DisplayFunction está definida para enviar gráficos e Output para o canal "stdout", i.e. \$DisplayFunction = Display[{"stdout"}, #1] &. Contudo, ao gerar os gráficos pode-se especificar a opção DisplayFunction → Display["channel",#, "format"] & onde "channel" pode ser um documento para onde o gráfico é exportado. Alternativamente, depois de gerado o gráfico com o nome graph, pode-se invocar Display["file.ext", graph, "format"] para o converter numa imagem de formato "format" guardado em file.ext. Os formatos podem ser qualquer um de:

"APS"	<i>Mathematica</i> abbreviated PostScript	(.aps)
"EPS"	Encapsulated PostScript	(.eps)
"PDF"	Adobe Acrobat portable document format	(.pdf)
"PICT"	Macintosh PICT	
"SVG"	Scalable Vector Graphics	(.svg)
"WMF"	Windows metafile format	(.wmf)
"AVI"	Audio Video Interleave format	(.avi)
"BMP"	Microsoft bitmap format	(.bmp)
"DICOM"	DICOM medical imaging format	(.dcm, .dic)
"EPSI"	Encapsulated PostScript with device-independent preview	(.epsi)
"EPSTIFF"	Encapsulated PostScript with TIFF preview	
"GIF"	GIF and animated GIF	(.gif)
"JPEG"	JPEG	(.jpg, .jpeg)
"MGF"	<i>Mathematica</i> system-independent raster graphics format	(.mgf)
"PBM"	portable bitmap format	(.pbm)
"PCX"	PCX format	(.pcx)
"PGM"	portable graymap format	(.pgm)
"PNG"	PNG format	(.png)
"PNM"	portable anymap format	(.pnm)
"PPM"	portable pixmap format	(.ppm)
"TIFF"	TIFF	(.tif, .tiff)
"XBitmap"	X window system bitmap	(.xbm)
"DXF"	AutoCAD drawing interchange format	(.dxf)
"STL"	STL stereolithography format	(.stl)