

Um Guia Curto de Mathematica Para Principiantes

Maria Madrugo

Kernel e Ctrl-S

Em Mathematica, os comandos e funções são avaliadas pelo Kernel. Eis alguns cuidados que é preciso ter ao usar mathematica, bem como os principais erros e problemas que o kernel pode criar:

- O Kernel no início demora um bocadinho a começar. Se quiseres sentir-te mais inteligente que um computador, tenta-lhe perguntar quanto é "2+2" mal ligas o Mathematica e vê o tempo que ele demora. De qualquer forma, sê paciente no início, se demorar um bocadinho a correr em princípio é só o Kernel a começar.
- O Kernel tenta sempre avaliar o que lhes damos. Às vezes, damos-lhe loops infinitos, ou funções chamadas com parâmetros demasiado grandes, ou qualquer outra coisa que faça o programa crashar. Assim sendo,
 - Se o Mathematica ainda funciona e está a demorar muito a correr, ou te apercebeste que tens um loop infinito, **aborta a avaliação** (Ctrl-ponto), e se isso não funcionar tenta **abortar o kernel** (Evaluation, Quit Kernel, Local), e recomeçá-lo.
 - **Grava frequentemente** (Ctrl-S) o teu programa. Não te queiras juntar ao clube de pessoas que crasham o Mathematica e perdem horas de progresso porque não estava guardado.
- O Kernel tende a guardar informação sempre que possível. Isso faz com que às vezes aconteçam coisas esquisitas, que se podem dever a ele ter os valores antigos das variáveis e funções, quer de outros notebooks como de versões antigas das funções que tens. Assim sendo,
 - Se o teu programa estiver a dar coisas estranhas, tenta **Reiniciar o Kernel**, pode ser que esteja a usar versões antigas das variáveis, ou de outro notebook;
 - Se o teu programa que funcionava perfeitamente estiver a dar erro sem teres feito nenhuma mudança no código e reiniciar o Kernel não ajudar, é possível que em algum momento tenhas feito mudanças na tua função que não correram bem, e que levaram o Kernel a usar versões antigas da última vez. Agora que o Kernel reiniciou, está a usar a versão mais recente de certeza, e a desmascarar erros nos quais não tiveste oportunidade de reparar da última vez. Assim sendo, tenta ver que **mudanças fizeste** ao código desde que ele funcionava bem, isso deve ajudar a diagnosticar o problema.

Ajuda e Comandos

Quando se começa a usar uma linguagem, há três grandes coisas a aprender:

- Como funciona a linguagem em geral: como é a sintaxe, que tipo de código é comum fazer, etc;
- Como cada comando funciona;
- Que comandos tens ao teu dispor.

Esta disciplina tenta-te ajudar com todos eles, mas felizmente tens as seguintes ferramentas ao teu dispor para fazer essa aprendizagem mais fácil:

- Se quiseres aceder a toda a informação sobre o Mathematica, tens aqui a Documentação oficial do Mathematica
- Se estiveres a programar num notebook, tens um **pedido de ajuda built-in** para facilitar o acesso ao funcionamento de um dado comando. Isso significa que se precisares de saber como um comando ou função funciona, podes pesquisar o nome do comando, precedido por um ponto de interrogação.

- Por exemplo, podes pesquisar **?While** para descobrir como este funciona em Mathematica, e obténs como resposta *"While[test,body] evaluates test, then body, repetitively, until test first fails to give True."*, descobrindo então que lhe deves dar dois parâmetros, o "test" e o "body", que fazem o indicado.
- Se precisares de mais informação, ver exemplos, ver usos específicos, etc, clicares no círculo com um i leva-te à secção da documentação oficial dedicada a esse comando.
- Se não souberes bem o comando que procuras, podes utilizar asteriscos para descobrir todos os comandos que contenham uma certa palavra. Por exemplo, **?*While*** mostraria os comandos LengthWhile, NestWhileList, While, NestWhile e TakeWhile, nos quais poderias clicar para descobrir mais sobre cada um.
- Podes usar o comando **Options** para ver as opções para um dado comando. Por exemplo, Options[Plot] dá uma lista grande de modificações que podes fazer a gráficos, muitas das quais estéticas. Podes usar o Help para ver o que cada uma faz. Nota que para este tipo de coisa talvez seja melhor veres a documentação ou pesquisares no google, porque obtens resultados mais específicos e com exemplos.
- Se estiveres a ver documentação e não perceberes bem um comando, tenta ver os exemplos. Não só o seu funcionamento fica mais claro, como por vezes a descrição é mais geral do que a que precisas e faz com que não percebas o que lá está escrito
 - Por exemplo, imagina que queres pôr um número em binário e descobres que a forma de o fazer é usar "BaseForm", mas quando vais ver a descrição lês *"BaseForm[expr,n] prints with the numbers in expr given in base n."*, e não sabes o que é uma base. Se vires os exemplos, o primeiro tem logo a informação que procuras, uma vez que diz "A binary integer: In[1]:= BaseForm[33, 2]", que tem como output 10001_2 (reconhecivelmente 33 em binário). "
- Se não souberes que comando faz o que queres, tenta **pesquisar no Google em inglês**, geralmente dá os resultados que queres. Uma coisa super útil a ter em mente é que o Mathematica tem uma variedade de funções muito grande, por isso se precisares de alguma função tenta pesquisar no google uma descrição geral antes de te pões a implementar à mão: é muito provável que ela exista.
 - Imagina que queres escrever um número em binário. A pesquisa "Mathematica number to binary" dá como resultado o comando BaseForm, que vimos acima ser exactamente o que procuramos.
 - Existe uma função que dadas duas listas (ou uma lista e um elemento) cria uma lista nova com os elementos alternados.
 - Existe uma função que indica as formas que tens de escrever um número inteiro como a soma de potências inteiras.
 - Experimenta correr "Sunset[Today]", "Sunrise[Today]" ou "SunPosition[Now]". Infelizmente não vem com imagens bonitas, mas deve ajudar a provar o ponto que Mathematica tem mesmo muitas funções, por isso é provável que tenha o que precisas.

Sintaxe do Mathematica

Linguagens de programação nem sempre são muito consistentes com notação, nomeadamente com pontos e vírgulas. Felizmente, o Mathematica é bastante consistente nesse sentido:

- Usamos **vírgulas** para separar parâmetros de funções, sempre.
- Usamos **ponto e vírgulas** para separar comandos dentro do mesmo parâmetro.
- Quase todos os comandos são funções, incluindo If, While, etc. Funções são avaliadas usando **parêntesis rectos**.
- Números não inteiros usam **ponto final**. Isto é, escrevemos números decimais como 2.5 em vez de 2,5.

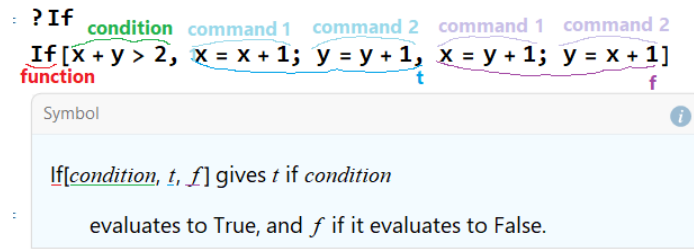


Figura 1: Exemplo de sintaxe

Outras Dicas

Nesta secção, está uma colectânea de dicas miscelâneas, algumas das quais são simples, enquanto outras têm pdfs dedicados a elas.

- O Mathematica tende a fazer cálculo simbólico. Assim sendo, tende a não fazer arredondamentos a não ser que seja necessário. Para forçar este comportamento, existe o comando **N**.
 - Dado o input $3/4$, o Mathematica retorna $\frac{3}{4}$.
 - Dado o input $N[3/4]$, o Mathematica retorna 0.75.
- O Mathematica é uma linguagem maioritariamente funcional, que também tem comandos imperativos implementados. Isto significa que muito do código de outras linguagens (nomeadamente os comandos If, For, While) pode ser facilmente traduzido, mas significa também que isso nem sempre é a melhor forma de fazer as coisas.
 - Neste pdf *Do, For, While, e melhor ainda: Nenhuma das anteriores*, podes encontrar mais informações sobre quando evitar abordagens iterativas.
- Como seguimento do item anterior, o Mathematica é muito prático para manipulação de listas, com muitas operações de criação, modificação e aplicação de funções a listas.
 - Neste pdf *Listas e Programação Funcional.*, podes encontrar mais informações e explicações sobre como trabalhar com listas e como programar funcionalmente.
- Os parâmetros de funções¹ não podem ser modificados. Isso quer dizer que se queres modificar algum dos parâmetros da tua função, deves usar uma variável local auxiliar (usando Module).

Exemplo: $f[x_] := (x=x+1; Sqrt[x])$ dá erro; $f[x_] := Module\{y=x\}, y=y+1; Sqrt[y]$ tem o comportamento esperado.
- Existem alguns comandos que quando usados de certa forma são surpreendentemente lentos, e que fazem o código desnecessariamente demorado. Se tiveres um código mais lento do que esperarias, tenta usar este pdf *O meu código demora a correr: e agora?*, para ver se alguma das dicas te ajuda.
 - Refiro aqui a manifestação mais comum deste fenómeno. Ao criar ou modificar uma lista *listy*, usar o comando “ $listy=Append[listy,elem]$ ” é muito lento se se adicionarem muitos elementos, porque o Mathematica está sempre a criar a lista de novo. Uma alternativa mais rápida consiste em usar $AppendTo[listy,elem]$, e mais rápido ainda é construir as tuas listas usando o comando Table. Nota que esta diferença na velocidade deve-se ao facto de o Append e o AppendTo serem comandos ligeiramente diferentes. Enquanto o segundo actualiza a lista listy, adicionando elem no final, o primeiro cria uma lista nova (igual à listy, mas com o elemento elem no final) e mantém listy na mesma.

¹Se não sabes ainda como funcionam funções em Mathematica, podes encontrar os básicos neste pdf *Funções de Mathematica 101*.