1 Mathematica を始める

■ 起動，計算，保存，終了，再開

■ 起動

Mathematica の起動を行う．そのためには，次のどちらかを実行する．

- 画面上部のメニューバー左端の記号（足跡のようなもの）をクリックして，アプリケーションを選び，その中の Mathematica をクリックする．
- 画面上部のメニューバーにある，日の丸のついた画面のアイコンをクリックし，起動して kterm の画面で， mathematica と入力する．

■ 計算

起動して開いたウインドウに，キーボードから1+2と入力し，それに続いてSHIFT RETを押すと（SHIFTキーを押しながらRETキーを押す），計算を実行して，結果が表示される．RETキーだけでは，改行するだけで計算は実行されない．

```
In[1]:= 1 + 2 SHIFT RET

Out[1]= 1 + 2 SHIFT RET

        1 + 2
```

以降、SHIFT RETは省略する．

起動直後に計算を始めるときには， MathematicaKernelを立ち上げる時間が必要なので，すこしだけ時間がかかる．

もう少し高等な計算を行ってみる．

```
In[2]:= t = Pi / 3

Out[2]= π/3

In[3]:= Cos[t] + Sin[t]

Out[3]= 1/2 + √3/2
```

■ 保存

ここで，以上の計算結果をfirstStepという名前のファイルに保存する．

Mathematica のウインドウの最上部（メニューバー）にある，File（ファイル）メニューから Save（保存）を選択し，現れたファイル選択ブラウザでファイルを保存するディレクトリを選択した後，ファイル名の所に firstStep を入力して，OKを押す．

ktermのウインドウ内で，
ls
を実行して，firstStep.nb というファイルができていることを確認する．（拡張子 nb は notebook の略である）

■ 終了

ここで，とりあえず Mathematica を終了してみる．

今は，保存を行ってから終了したので何事も起こらなかったが，保存を行わないで Mathematica を終了したりウインドウを閉じようとすると，保存をするかどうかを聞いてくるダイアログボックスが開く．その時点でもファイル名を指定して保存を行うことができる．

■ 再開

さきほど，保存したファイルを開いて，計算の続きを行う．

まず Mathematica を起動する．次に File メニューから Open...（開く...） を選択し，KagakuKeisanKiso ディレクトリ内にある firstStep.nb というファイルを選択し，OK または Open（開く）を押す．

開いたウインドウには，先ほど入力したものとその計算結果が表示されている．ここで次の計算を入力して実行してみる．

```
In[4]:= Cos[t] + Sin[t]
```

$$\text{Out[4]=}\quad \frac{1}{2}+\frac{\sqrt{3}}{2}$$

t の値が $\frac{\pi}{3}$ ではなく， t そのままに戻っている．

Mathematica で保存を行っても，保存されるのはウインドウの状態だけで，計算の状態（xにPi/3が代入されていることなどの状態）は保存されない（このことについては，後の“Notebook と Kernel”の項で解説する）．したがって，計算を再開するには，ウインドウの先頭から，もう一度，順に計算を実行する必要がある．
以前に入力した計算式をもう一度計算するには，その部分をクリックして，そこにカーソルを持っていき，そこでSHIFT RETを押せばよい．または，全ての計算式を再計算するには， Editメニューから Select Allを選び（このとき，ウインドウの右端にある全てのカギ形（セル表示という）の色が変化する），SHIFT RETを押す．

■ 計算例

Mathematica でどのようなことができるのか，例をいくつか見てみる．

■ 有理数の計算

有理数は近似値ではなく，分数として計算する．

```
In[5]:= 123 / 4321 + 32 / 7
```

$$\text{Out[5]}= \frac{139133}{30247}$$

■ 巨大な数の計算

次は，100 の階乗の計算結果である．

```
In[6]:= 100!
```

```
Out[6]= 93326215443944152681699238856266700490715968264381621468592963895217599993229915608941463976156518286253697920827223758251185210916864000000000000000000000000
```

■ 円周率，自然対数の底

円周率は Pi で表される．次は，円周率の値を100桁の精度で求めた結果である．

```
In[7]:= N[Pi, 100]
```

```
Out[7]= 3.14159265358979323846264338327950288419716939937510582097494459230781640628620899862803482534211706 8
```

■ 素数

関数Prime[n]はn番目の素数を返す．

```
In[8]:= Prime[100000000]
```

```
Out[8]= 2038074743
```

■ 素因数分解

関数 FactorInteger[n] は整数 n の素因数分解を返す．次の計算結果は，98601032853536486160375 の素因数分解が$3^2 \times 5^3 \times 97 \times 4406023 \times 205073926877$ であることを意味する．

```
In[9]:= FactorInteger[98601032853536486160375]
```

```
Out[9]= {{3, 2}, {5, 3}, {97, 1}, {4406023, 1}, {205073926877, 1}}
```

■ 逆行列

関数 Inverse は行列の逆行列を返す．たとえば行列 $\begin{pmatrix} 1 & 2 \\ 3 & 5 \end{pmatrix}$ の逆行列を求めてみよう．なお，$\begin{pmatrix} 1 & 2 \\ 3 & 5 \end{pmatrix}$ という行列は {{1,2},{3,5}} という形で入力する．

```
In[10]:= Inverse[{{1, 2}, {3, 5}}]

Out[10]= {{-5, 2}, {3, -1}}
```

■ 多項式の展開と因数分解

$(a+b)^{10}$ の展開(expand) と $x^{15}+y^{15}$ の因数分解(factor)とをする．

```
In[11]:= Expand[(a + b) ^ 10]

Out[11]= a^10 + 10 a^9 b + 45 a^8 b^2 + 120 a^7 b^3 + 210 a^6 b^4 +
          252 a^5 b^5 + 210 a^4 b^6 + 120 a^3 b^7 + 45 a^2 b^8 + 10 a b^9 + b^10

In[12]:= Factor[x ^ 15 + y ^ 15]

Out[12]= (x + y) (x^2 - x y + y^2) (x^4 - x^3 y + x^2 y^2 - x y^3 + y^4) (x^8 + x^7 y - x^5 y^3 - x^4 y^4 - x^3 y^5 + x y^7 + y^8)
```

■ 数列の和

数列 $a_k = k^2$ の初項から第 n 項までの和を求める．（具体的な値ではなくて，一般式の形で計算できていることに注意）

```
In[13]:= Sum[k^2, {k, n}]

Out[13]= 1/6 n (1 + n) (1 + 2 n)
```

■ 微分と積分

sin((log x)) を微分し，その結果を積分してみる．

```
In[14]:= D[Sin[Log[x]], x]

Out[14]= Cos[Log[x]]
         ---------
             x

In[15]:= Integrate[Cos[Log[x]] / x, x]

Out[15]= Sin[Log[x]]
```

■ 方程式の解

$x^3 - 2x^2 - 2x + 1 = 0$ の解を求める．

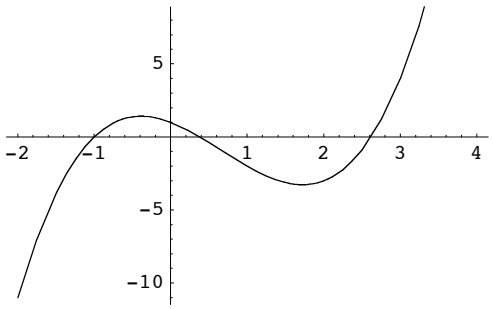
```
In[16]:= Solve[x^3 - 2 x^2 - 2 x + 1 == 0, x]
```

$$\text{Out[16]= } \left\{\{x \to -1\}, \left\{x \to \frac{1}{2}\left(3-\sqrt{5}\right)\right\}, \left\{x \to \frac{1}{2}\left(3+\sqrt{5}\right)\right\}\right\}$$

■ 2次元プロット

先ほどの方程式の左辺をグラフにしてみると，ちょうど解の 3 点で x-軸と交わっていることがわかる．

```
In[17]:= Plot[x^3 - 2 x^2 - 2 x + 1, {x, -2, 4}]
```



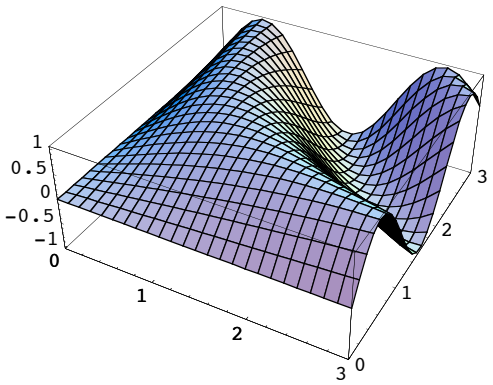

```
Out[17]= - Graphics -
```

次は， $z = \sin(x\,y)$ という 2 変数の関数のグラフである．

■ 3次元プロット

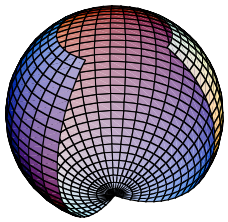
z=sin(x y) という2変数の関数のグラフを描く．

```
In[18]:= Plot3D[Sin[x y], {x, 0, 3}, {y, 0, 3}]
```



原点を中心とした半径 1 の球面上の点を，緯度 s と経度 t でパラメータ表示すると ((cos s)(cos t), (cos s)(sin t), sin s)
$-\frac{\pi}{2} \le s \le \frac{\pi}{2}$, $0 \le t \le 2\pi$ となる．これを用いて球面の一部を表示する．

```
ParametricPlot3D[{Cos[s] Cos[t], Cos[s] Sin[t], Sin[s]},
 {s, -Pi / 2, Pi / 3}, {t, 0, 2 Pi - Pi / 2}, Boxed → False, Axes → False]
```



```
- Graphics3D -
```

■ 微分方程式の解

微分方程式 y''+y+sin(x)=0 を解く.

```
DSolve[y''[x] + y[x] + Sin[x] == 0, y[x], x]
```

$$\left\{\left\{y[x] \to C[1]\ \mathrm{Cos}[x] + C[2]\ \mathrm{Sin}[x] + \frac{1}{4}\left(2\,x\,\mathrm{Cos}[x] + 2\,\mathrm{Cos}[x]^2\,\mathrm{Sin}[x] - \mathrm{Cos}[x]\,\mathrm{Sin}[2\,x]\right)\right\}\right\}$$

■ Notebook と Kernel

Mathematica は，Notebook と Kernel という二つのプログラム（アプリケーション）で成り立っている．それらは，次のように働く．

Notebook はユーザーからの入力を受け取り，SHIFT RETが押されたときに，それを Kernel に送る．Kernel は Notebook から送られた計算式を評価し，計算を実行し，その結果を Notebook へ返す．Notebook は，Kernel から返された計算結果を，きれいな式の形，もしくはグラフィックスとして表示し，ユーザーに見せる．

また，ファイルへの保存を行ったときに，保存されるのは Notebook の現在の状態であって， Kernel の状態は保存されない．そこで，保存されたファイルを開いて計算を再開するためには，開いた Notebook にある計算式のところにカーソルを持っていってSHIFT RETを押すことにより， Kernel にもう一度計算をさせる必要がある．

たとえていうと，計算をしている人の頭脳が Kernel であり，計算式を書き留めたノートが Notebook である．一晩経ってから計算を再開するには，もう一度ノートを最初から読んでもらう必要があるのと同じ事である．

■ ヘルプ

Mathematica を使っているときに，関数名や使い方が分からなくなった場合，ヘルプを利用するとよい．

■ 簡易ヘルプ

Mathematica を使っていて，組み込み関数のちょっとした使い方を知りたいとき，あるいは，組み込み関数名をあいまいにしか覚えていないときに，便利である．

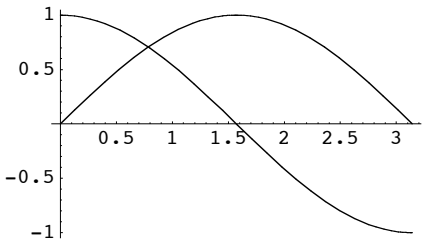
例えば，sin(x) と cos(x) のグラフを 0<x<π の範囲で描きたいとする．関数のグラフを表示するPlotという関数の使い方を知るには，?Plot と入力し，SHIFT RETを押す．

```
? Plot
```

```
Plot[f, {x, xmin, xmax}]は，fをxminからxmaxの範囲のxの関数としてプロットを作成する．
   Plot[{f1, f2, ... }, {x, xmin, xmax}]は，複数の関数fiを作図する． 詳細
```

関数 Plot の簡単な使い方が表示される．これを読むと，次のようにすればよいことがわかる．

```
Plot[{Sin[x], Cos[x]}, {x, 0, Pi}]
```




```
- Graphics -
```

また，使用法の最後にある 詳細 をクリックすることで，さらに詳しい解説が得られる．

次に，グラフを描く関数には，他にどのようなものがあるか知りたいとする．そのような関数は，...-Plot という名前が付いているので，名前の最後に Plot が付くような組み込み関数の一覧が欲しい．それは ?*Plot とすることにより，得られる．

```
? *Plot
```

System`

```
ArrayPlot   DensityPlot     ListDensityPlot ParametricPlot
ContourPlot ListContourPlot ListPlot        Plot
```

これらの関数名をクリックすると，詳しい解説が得られる．

また，?*Plot* とすれば，名前の中にPlotを含むようなものがすべて表示される．

```
? *Plot*
```

System`

ArrayPlot	ListPlot	Plot	PlotLabel
ContourPlot	ListPlot3D	Plot3D	PlotPoints
DensityPlot	MaxPlotPoints	Plot3Matrix	PlotRange
ListContourPlot	ParametricPlot	PlotDivision	PlotRegion
ListDensityPlot	ParametricPlot3D	PlotJoined	PlotStyle

このヘルプで表示されるのは，関数だけではなく，変数や定数などの全てのシンボル（関数，変数，定数などにつけられた名前）が表示される．たとえば，上のリスト表にある PlotStyle というものは関数ではない．また，組み込みのシンボルだけではなく，ユーザーが使用したシンボルについても，その定義を表示する．

```
a = 3
```

```
3
```

```
? a
```

```
Global`a
```

```
a = 3
```

■ ヘルプ・ブラウザ

Mathematica のウインドウの上部にあるメニューの右端に，Helpという項目がある．これの中のHelp...を選ぶと，ヘルプ・ブラウザが開かれる．このヘルプ・ブラウザには， Mathematica の分厚いマニュアル 1 冊分の情報が入っている．これを使いこなしたら，本のマニュアルは不要である．

ブラウザの上半分には，表示する内容を絞り込むための選択肢が表示されている．左側から右側に行くに従い，細かい選択肢となっている．

例えば， 3 次元グラフィックス関係の組み込み関数について調べるとする．まず，左端の選択肢の中から組み込み関数 を選ぶ．次の選択肢の中から グラフィックスとサウンド を選ぶ．その次の選択肢の中から3Dプロット を選ぶ．すると，最後の選択肢に，Plot3D, ListPlot3D, ParametricPlot3D という三つの関数が現れるので，例えば Plot3D をクリックすると，その詳しい使用方法が表示される．

解説の最後に 他の例 という項目がある．その左にある三角印をクリックすると，その内容を見ることができる．ここにある入力式は，そのまま実行することができる．そのためには，入力式をクリックして，その上にカーソルを持っていき，SHIFT RET を押す．

■ エラーメッセージと警告メッセージ

Mathematica を使っていると，いろいろなエラーメッセージや警告メッセージが表示される．中には無視してもよいメッセージもあるが，ほとんどの場合は，括弧が足りないか，組み込み関数の使い方を間違っているか，保護された組み込みのシンボルを書き換えようとしたか，計算途中で処理のできない状態になったか，などである．これらの場合には，もう一度やり直す必要がある．

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

```
Syntax::bktmcp : 式"{x, 0, 2 Pi"には閉じる"}"がありません. 詳細
```

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

この場合は，}が足りない．

```
Plot[Sin[x]]
```

```
Plot::argmu : Plotは1つの引数で呼ばれました．2個あるいはそれ以上の引数が想定されています. 詳細
```

```
Plot[Sin[x]]
```

この場合は，Plotの使い方を間違えている．Plotの引き数は2個かそれ以上必要である．

```
D = 1
```

```
Set::wrsym : シンボルDはProtectedです. 詳細
```

```
1
```

D は他の用途で使用されているシンボルであり，それに代入したりすることはできない．

```
{1, 2, 3}[[4]]
```

```
Part::partw : {1, 2, 3}の部分4は存在しません. 詳細
```

```
{1, 2, 3}〚4〛
```

リスト{1,2,3}には4番目の要素は存在しないので，取り出すことができない．

```
max = 10
```

```
General::spell1 : スペル間違いの可能性があります．新規シンボル"max"はすでにあるシンボル"Max"に似ています. 詳細
```

```
10
```

この場合に発せられる警告メッセージは，「maxというシンボルが，既に定義されているMaxというシンボルに似ているので，綴り間違いではないか？」という親切なメッセージであり，無視してもよい．
しかしながら，本当に綴りが間違っていることも多い．次の例の場合はcosではなくCosが正しい．

```
y = Log[cos[x]] + Sin[x]
```

```
General::spell1 : スペル間違いの可能性があります．新規シンボル"cos"はすでにあるシンボル"Cos"に似ています. 詳細
```

```
Log[cos[x]] + Sin[x]
```

■ 演習問題

[1-1] Listable, NestList などのように，List を含む組み込みのシンボルがいくつあるか．簡易ヘルプ ? を用いて調べよ．

[1-2]　y = sin(x) のグラフを描画しようと思って次の式を入力したが，エラーメッセージが出る．正しく直せ．

```
Plot[sin[x], {x, 0,2Pi}
```