Chapter 2

数と式

Mathematica は数値計算だけでなく，因数分解，部分分数展開といった文字式の計算もこなしてくれる．*Mathematica* の基礎のキソとして，数値の扱い（厳密値 vs 近似値），文字式の変形，値の代入といった操作を身につけよう．

2.1 厳密値 vs 近似値

電卓に 1 ÷ 3 と入力すると，0.33333333 のような答が帰ってくる．もちろんこれは近似値にすぎず，厳密な意味で正しい値ではない．電卓で $\frac{1}{3}$ という量を正確に扱うことはできないのである．

Mathematica はその点が違っていて，$\frac{1}{3}$ のような分数もわれわれと同じように認識し計算することができる．

[1] たとえば $\frac{1}{2} - \frac{1}{3}$ を計算させる[*1]：

```
In[1]:= 1/2 - 1/3
```

$$\text{Out[1]=}\quad \frac{1}{6}$$

[2] 文字式も同様で，たとえば半径 2 の円の面積は次のように計算される[*2]：

```
In[2]:= 2*2*Pi
```

$$\text{Out[2]=}\quad 4\,\pi$$

[*1] 四則演算は加減乗除にそれぞれ `+ - * /`，べき乗には `^` を用いるのであった（1 章 1.4 節）

[*2] 1 章 1.5 節 (3) で述べたように，乗算 `2*2*Pi` は `*` のかわりに半角空白 ␣ をはさんで `2␣2␣Pi` と書いてもよい．本書はこれらの記法を併用するが，数字と数字の積に半角空白を使うとさすがに紛らわしい（たとえば 「`2 2.0`」 と 「`22.0`」）ので，意識的に `*` を用いている．

ここで Pi は円周率 π を表す *Mathematica* の組込み定数である．（出力ではこのように認識しやすいフォントに置き換わる．）*Mathematica* は文字や分数をできるだけ厳密な形で扱おうとするのである．

[3] 逆に数値化したい場合は，明示的に指示しなければならない．たとえば組込み関数 N を用いると，[1] の結果を数値化できる：

```
In[3]:= N[1/2 - 1/3]

Out[3]= 0.166667
```

ここでは有効数字 6 桁のみが表示されているが，カーネル内では**機械精度**とよばれる有効数字 16 桁程度の精度で計算されている．

[4] InputForm を用いると，その実際の値がわかる：

```
In[4]:= InputForm[N[1/2 - 1/3]]

Out[4]//InputForm= 0.1666666666666666
```

[5] N 関数では数値化の精度を指定することもできる．たとえば有効数字 27 桁にする：

```
In[5]:= N[1/2 - 1/3, 27]

Out[5]= 0.166666666666666666666666667
```

[6] [1] の 3 の部分を小数表示 3.0 に変えても，[3] の N 関数と同じ結果が得られる：

```
In[6]:= 1/2 - 1/3.0

Out[6]= 0.166667
```

Mathematica はただの 3 を**厳密値**，小数で表された 3.0 （3. や 3.00000 と書いても同様）を**近似値**として，計算を進める上で厳格に区別する．計算式中にひとつでも近似値が含まれると，*Mathematica* は分数による計算をあきらめ，全体を近似値で数値計算するのである．

厳密数による計算は誤差がまったくないという点で理論的には優れているが，機械精度の近似値計算に比べると時間がかかってしまう．理論か応用か，目的に応じて上手に使い分けよう．

2.2　組込み定数・組込み関数

さきほど用いた数値化関数 N のように，*Mathematica* にあらかじめ定義されている関数を**組込み関数**とよぶ．また，円周率 π を表す Pi のような厳密値の定数を**組込み定数**とよぶ．組込み関数はすべて大文字から始まることに注意しよう（1 章 1.5 節

(5)).

われわれが思いつくような関数（定数）はだいたい組込み関数（定数）になっているし，われわれが知らないような関数（定数）もたくさん組込み関数（定数）になっている．なにか必要な関数や定数があったら，まずはドキュメントセンター（1 章 1.10 節）を開いて検索してみるとよいだろう．

ここではまず，ごく基本的な組込み定数（厳密値）と，数値化に関連する組込み関数をまとめておく：

円周率（定数）：	Pi	x を数値化：	N[x]
自然対数の底（定数）：	E	x を m 桁で数値化：	N[x, m]
虚数単位（定数）：	I	x の整数部分（厳密値）：	Floor[x]

[7] たとえば複素数 $2(\pi + e - 3\,i)$ は次のように入力する：

```
In[7]:= 2*(Pi + E - 3*I)
Out[7]= 2 (-3 i + e + π)
```

出力においても数値化されず厳密値のままである．また，記号はより認識しやすいフォントに，掛け算を表す * は（わかりづらいが）空白 ␣ に置き換わっている．*Mathematica* なりのルールで項の順序も置き換えられることに注意しよう．

[8] Floor[x] は x を超えない最大の整数を厳密値で与える：

```
In[8]:= Floor[2*Pi]
Out[8]= 6
```

N や Floor のほかにも，*Mathematica* には三角関数などの有名な関数があらかじめ組込み関数として定義されている．ここでは最低限のものとして，高校や大学でおなじみの関数を紹介しておこう：

x の絶対値：	Abs[x]	x の三角関数：	Sin[x]
x の平方根：	Sqrt[x]		Cos[x]
x の指数関数：	Exp[x]		Tan[x]
x の自然対数：	Log[x]	x の逆三角関数：	ArcSin[x] など
x の対数（底 a）：	Log[a , x]		

[9] $\sin(\pi/4) - \sin(\pi/7)$ を計算：

```
In[9]:= Sin[Pi/4] - Sin[Pi/7]
```

$$\text{Out[9]=}\quad \frac{1}{\sqrt{2}} - \text{Sin}\left[\frac{\pi}{7}\right]$$

このように，厳密性を失わない範囲でできるだけ具体的な形にしてくれる．

[10] 4 を 4.0 に変えると，近似値で計算される：

```
In[10]:= Sin[Pi/4.0] - Sin[Pi/7]

Out[10]=  0.273223
```

[11] 一見ありえない値も，複素関数として処理される[*3]：

```
In[11]:= ArcSin[3.0]

Out[11]=  1.5708 - 1.76275 i
```

このように，*Mathematica* の動作にはある種の「くせ」がある．その点を理解しておけば，期待した結果が得られなくても対策が練りやすくなるものである．とくに関数ごとの「くせ」は，ドキュメントセンターを検索して調べるのが一番である．

では，これまでに学んだ知識を活用して例題を解いてみよう．以下とくに断らない限り，「近似値」といったら N 関数で数値化した値とする．

問題 2.1 e^{π}, π^{e}, $\pi + 20$ の近似値をそれぞれ有効数字 7 桁で求め，もっとも小さなものを求めよ．

【解答】 次のように入力・評価し，値を比較する：

```
In[ ]:= N[E^Pi, 7]

Out[ ]=  23.14069

In[ ]:= N[Pi^E, 7]

Out[ ]=  22.45916

In[ ]:= N[Pi + 20, 7]

Out[ ]=  23.14159
```

これより π^{e} が最小であることがわかる． ■

[*3] 実関数としてのアークサイン arcsin の定義域は $-1 \leq x \leq 1$ であった．

2.3 文字式の計算と定数の割り当て・上書き

われわれは習慣的に文字 e を自然対数の底 2.71828... という定数として用いている．同様に，*Mathematica* ではユーザーが好きな文字列に「定数」を割り当てることができる．[*4]

[12] **イコール** `=` を用いて，文字 `n` に定数 2^{16} を**割り当て**る：

```
In[12]:= n = 2^16
Out[12]= 65 536
```

この入力式は「左辺と右辺が等しい」（等式）という意味ではなく，「左辺の文字に右辺の値を割り当てよ」という意味である．[*5] より正確には，「左辺の文字 `n` に『右辺 `2^16` を評価したもの』を割り当てよ」という意味になる．出力はその『右辺 `2^16` を評価したもの』が 65536 であることを教えてくれている．

[13] 「定数」`n` を用いた計算（$\log_2 2^{16}$）をさせる：

```
In[13]:= Log[2, n]
Out[13]= 16
```

こうして定められた `n` は，上書きもしくはクリアされない限り，*Mathematica* のセッションが終了するまで「定数」65536 であり続ける．これが意外とエラーの原因になってしまうので，十分に注意しておこう（1 章 1.6 節）．

[14] イコール `=` を用いると，文字（列）に数値以外のものを割り当てることもできる．たとえば文字列 `abc` に文字式 `x^2 + 1` を割り当てる：

```
In[14]:= abc = x^2 + 1
Out[14]= 1 + x^2
```

[15] `abc` の多項式を計算させてみる：

```
In[15]:= abc^2 - abc + 1
Out[15]= -x^2 + (1 + x^2)^2
```

あまりきれいな式ではないが，ちゃんと `abc` が `x^2 + 1` に置き換わっていることは

[*4] 組込み定数や関数名（`Pi` や `E`，`Sin` のような大文字で始まる文字列）は *Mathematica* 内で保護されているので，これらに別の値を割り当てることはできない．また，数字や記号から始めることはできない．（たとえば，`2a` は `2*a` と解釈される．）

[*5] プログラミングの世界では「代入演算子」とよばれるものに対応する．

わかるだろう．

[16] 式がイメージどおりでないときは，とりあえず組込み関数 Simplify を施してみるとよい[*6]：

```
In[16]:= Simplify[abc^2 - abc + 1]
```

$$\text{Out[16]=}\quad 1 + x^2 + x^4$$

Simplify はさまざまな式変形を施して「できるだけ単純な式」をみつけてくれる関数である．ただし，いつもうまくいくとは限らない．

[17] さて練習問題．次の入出力の意味を説明せよ：

```
In[17]:= abc = x * abc
```

$$\text{Out[17]=}\quad x\left(1 + x^2\right)$$

答．これは，まず右辺 x * abc を評価し，その結果として得られる x (x^2 + 1) を左辺の文字列 abc に割り当てよ，ということである．すなわちもとの abc の値 x^2 + 1 は上書きされて，以後 x (x^2 + 1) となる．

[18] もう一度 [17] を繰り返す：

```
In[18]:= abc = x * abc
```

$$\text{Out[18]=}\quad x^2\left(1 + x^2\right)$$

このように文字（列）に割り当てられた値は，必要に応じて上書きし更新していくことができる．くどいようだが，好きなだけ上書きできるというこの性質がしばしばエラーを誘発する．要注意である．

問題 2.2 $a = \sqrt{3} + \sqrt{2}\,i$, $b = \sqrt{3} - \sqrt{2}\,i$ のとき，$c = \dfrac{a}{b} + \dfrac{b}{a}$ と $d = a^4 - 2a^2$ の値を求めよ．**（センター追試）**

【解答】　まずは a と b の値を定義:

```
In[ ]:= a = Sqrt[3] + Sqrt[2] I;
        b = Sqrt[3] - Sqrt[2] I;
```

何も出力されないが，これは行末にセミコロン（;）をタイプすることで無駄な出力を避けたからである（1 章 1.5 節 (8)）.[*7]さらに c と d の値を定義する：

[*6] ここで [15] の式を入力し直す必要はない．こういうときこそ，Ctrl + L（Mac は command + L）を用いるのである．1 章 1.9 節を参照せよ．

[*7] セミコロンには入力式を区切る役割があるため，一行に

```
In[ ]:= c = a/b + b/a
        d = a^4 - 2 a^2
Out[ ]= (-i Sqrt[2] + Sqrt[3])/(i Sqrt[2] + Sqrt[3]) + (i Sqrt[2] + Sqrt[3])/(-i Sqrt[2] + Sqrt[3])
Out[ ]= -2 (i Sqrt[2] + Sqrt[3])^2 + (i Sqrt[2] + Sqrt[3])^4
```

これでは a, b の値が代入されただけであるから，[16] のように **Simplify** を施してみる：

```
In[ ]:= Simplify[c]
        Simplify[d]
Out[ ]= 2/5
Out[ ]= -25
```

■

問題 2.3 $\sqrt{42+12\sqrt{6}}$ の整数部分を p, 小数部分を q とするとき, p, q, $r=\dfrac{p}{q(q+4)}$ の値を求めよ．（**成蹊大**）

【解答】 $x=\sqrt{42+12\sqrt{6}}$ と定義：

```
In[ ]:= x = Sqrt[42 + 12 Sqrt[6]];
```

ここでもセミコロン（;）を用いることで出力を抑えてみた．次に，実数の整数部分を与える組込み関数 **Floor** を用いると：

```
In[ ]:= p = Floor[x]
Out[ ]= 8
```

これより，x の小数部分は ：

```
In[ ]:= q = x - p
Out[ ]= -8 + Sqrt[42 + 12 Sqrt[6]]
```

と表される．さらに $r=p/q(q+4)$ を求めると，

```
In[ ]:= r = p/(q*(q + 4))
Out[ ]= 8/((-8 + Sqrt[42 + 12 Sqrt[6]]) (-4 + Sqrt[42 + 12 Sqrt[6]]))
```

となる．しかし入試問題の解答としては，二重根号をはずしてもっときれいな式にすることが要求されるだろう．とりあえず **q** と **r** に **Simplify** を施しても，ほとんど変化がない．そこ

```
a = Sqrt[3] + Sqrt[2] I; b = Sqrt[3] - Sqrt[2] I;
```

とまとめて入力してもよい．本書でも紙面を節約するために，しばしばこのセミコロンを活用する．

で Simplify をより強力にした FullSimplify という組込み関数（[29] 参照）を適用する．少しだけ時間がかかるが，手計算なみの結果となる：

```
In[ ]:= FullSimplify[q]
Out[ ]= -2 + √6
In[ ]:= FullSimplify[r]
Out[ ]= 4
```

■

2.4　式の操作（展開，因数分解，代入など）

式を単純化するのに Simplify は強力だが，もっときめ細かに式変形をしたい場合がある．たとえば [16] の Simplify で得られた $1+x^2+x^4$ はさらに $(1-x+x^2)(1+x+x^2)$ と因数分解できる．

因数分解や展開など，*Mathematica* の組込み関数でできる基本的な式変形をいくつか紹介しておこう．他にもたくさんあるので，ドキュメントセンターを参照してほしい．きっと必要な関数が見つかると思う．

[19] Factor を用いて $1+x^2+x^4$ を**因数分解**：

```
In[19]:= Factor[1 + x^2 + x^4]
Out[19]= (1 - x + x^2) (1 + x + x^2)
```

ここで上のような結果が出ず $2671+1020\sqrt{6}$ となったら，それは問題 2.3 で x に割り当てられた値が残っているからである．Clear[x] と入力・評価し値をクリアしてみよう（1 章 1.6 節）．文字 x の色が青く x と変化するはずである．その上で，もう一度 [19] をやり直してみよう．

[20] Expand を用いて $(1+x+y)^2$ を**展開**し，結果を s とおく：

```
In[20]:= s = Expand[(1 + x + y)^2]
Out[20]= 1 + 2 x + x^2 + 2 y + 2 x y + y^2
```

[21] Collect を用いて s を y の多項式として**まとめる**：

```
In[21]:= Collect[s, y]
Out[21]= 1 + 2 x + x^2 + (2 + 2 x) y + y^2
```

[22] s の x に値 2 を**代入**[*8]：

[*8] 入力式にある矢印「->」はマイナス - と不等号 > を並べたものである．さらに空白をつけ ->␣ と