$1+1$ はなぜ 2 になるのか
（知っておきたい集合、論理、公理の基礎知識）

1　集合

旅の準備として、数を理解することから始めよう。数の理解といっても、自然数や無理数がどうのという話ではない。この旅は、数とは何かという場所から出発する。したがって、旅の前に数の知識はないのである。もしいま、数を $1, 2, 3, \ldots$ なんて理解していたとしたら、それは忘れてほしい。今回は、そんなことすら準備せずに旅に出るのである。

まず、数は**集合**である。集合とはものの集まりを指すので、数はそれ自体で集合だ。集合に含まれるひとつひつのものは**要素**と呼ばれる。とくに、私たちが普通に使う数—何を普通と考えるかは人それぞれかもしれない—は**実数**と呼ばれる数だ。つまり、1 は実数の集合の要素である。円周率 π も実数の集合の要素である。しかし、a は実数の集合の要素ではない。単に文字としか見えないからだ。でも $a=10$ ということなら、a は実数の集合の要素である。ちょっと、ややこしいね。

大雑把に考えて、大きな集合は小さな集合に分けることができる。実数の集合は大きな集合である。大きいという表現も漠然としたものだが、実数には無数の数があるので、日常的な感覚ではこれ以上大きなものはないだろう。そこで、たとえば実数の集合を「正の数の集合」と「負の数の集合」に分けてみよう。数学の記号を使えば

$$\text{正の数の集合：}\{\ 1,\quad 0.333\cdots,\quad \frac{5}{8},\quad \sqrt{2},\quad \pi,\quad \ldots\}$$
$$\text{負の数の集合：}\{\ -273,\quad -0.0001,\quad -\frac{1}{3},\quad 1-\sqrt{3},\quad \ldots\}$$

のように、要素を集めたグループで記述できる。集合は { と } で囲んで表すのが一般的である。この記述により、実数の集合を 2 分することができた...のではない。なぜって、0 が入る場所がないから。一般に、0 は正でも負でもないとしている。よって、実数の集合は「正の数の集合」と「負の数の集合」と「正でも負でもない数の集合」に分けられることになる。そこで

$$\text{正でも負でもない数の集合：}\{0\}$$

を追加しておく。

さあ、いきなり旅の肝に出会ってしまった。$\{0\}$ って 0 と違うの？　そう、$\{0\}$ と 0 はまったく別ものである。説明すること自体難しいのだけれど、$\{0\}$ は要素が 0 だけの “集合” を意味し、0 は “数 0” を意味する。

要するに集合であるものと集合でないものという違いがある。その視点で眺めれば { } は何もない状態を示しているのではなく、要素がひとつもない集合を意味する。そのような集合は**空集合**と呼び、記号 $\emptyset$（斜線入りの 0）で表す。要素が何もないのに “ものの集まり” とみなす感覚はどうかと思うだろうが、首尾一貫をよしとする数学では自然な感覚なのだ。余談になるが、空集合の記号がフォントにないときは、ギリシア文字 ϕ で代用されることが多い。

集合には様々なものがあり、集合に含まれる要素もまた様々である。習慣として、集合をひとことで表す場合は大文字、任意の要素を表す場合は小文字を使うことが多い。たとえば、任意の負の数 x は実数の集合 R に**属する**が自然数の集合 N には**属さない**、などと表現する。当然、記号を使うほうが簡略化できるので、これらは $x \in R$ や $x \notin N$ と書いて、要素と集合の含有関係を示す。一方で、自然数の集合 N は完全に実数の集合 R に**含まれる**。集合と集合の包含関係を表す記号には $N \subset R$ を使う。

□ **集合で用いる記号**

- 集合を具体的に列挙する場合は { と } で囲む
- 要素をひとつも含まない集合は空集合と呼び $\emptyset$ または { } で表す
- 要素 x が集合 A に属する場合は $x \in A$ と書く
- 集合 A が集合 B に含まれる場合は $A \subset B$ と書く

旅には相変わらず **PowerShell** を持ち歩いているだろうか。**PowerShell** で厳密に集合は扱えないが、集合に近いことをやってもらうことはできる。たとえば $A = \{1, 2, 3, 4, 5\}$ は、ひとつの変数に代入することで実現できる。

[ps script]

```
PS C:\Users\Yours > $a = 1,2,3,4,5
PS C:\Users\Yours > "$a"
1 2 3 4 5
```

プロンプトに対して`"$a"`でなく`$a` と打ち込むと、値がひとつずつ改行されて表示される。本質は同じだけれど、$\{1, 2, 3, 4, 5\}$ に近い表示のほうが雰囲気がでるだろう。

さて、この旅の重要な鍵となることを話しておこう。それは、集合の要素は重複して記述しない、ということである。実際の例として、1, 2, 2, 3, 3, 3 を集合で考えた場合、それは $\{1, 2, 3\}$ と書くのである。なぜなら集合とは、“どれだけのもの” が集まっているかを表すのではなく、“どんなもの” が集まっているかを表すからである。そのため、重複に目を向ける必要はない。よって、$\{1, 2, 3\}$ も $\{3, 1, 2\}$ も同じものを意味する。その点から言えば、先に述べた $\{0\}$ と 0 の違いも理解しやすくなるかもしれない。0 はただひとつの 0 であるが、$\{0\}$ はいくつかの 0 を含んでいる場合があるからだ。

PowerShell はちゃんと処理できるだろうか。

[ps script]
```
PS C:\Users\Yours > $a = 1,2,2,3,3,3
PS C:\Users\Yours > "$a"
1 2 2 3 3 3
```

残念ながら要素が重複して表示されてしまった。しかし **PowerShell** はシェルだけに、様々の処理を行うためのコマンドがあらかじめ用意されている。続けて以下の操作を加えてみよう。

[ps script]
```
PS C:\Users\Yours > $a = $a | sort -uniq
PS C:\Users\Yours > "$a"
1 2 3
```

これは、-uniq パラメータを使って$a を sort し、結果を$a へ代入している。-uniq パラメータを使った sort は、データを並べ替えた後、重複するデータを削除するものである。この程度のことをやってくれれば、この先安心して旅が続けられる。なかなかやるじゃないか。