

## Introduction

さて、ちょっと数学に浸ってみようか。もちろん学校で学ぶ数学を基本にするけど、それにほんの少し調味料を効かせてね。数学に浸ると言っても庭をうろうろして土いじりする程度だし、調味料と言っても大層なものじゃない。庭の隅にばらばらと種を蒔いて何か収穫できれば儲けものって感じなんだ。

でも、種を蒔いたら水や肥料をやらなくちゃいけないね。そんなときはサンダル程度の履物で庭に出て、ちょっと土いじりをすればよい。もちろんジョウロとスコップは手にしているだろうが、手元にはコンピュータソフトウェアの **Haskell** を置いてみたい。庭いじりには大げさかもしれないけれど、数学とは相性が良いと思う。必ずしも必要な装備ではないが、庭いじりの最中に **Haskell** で書かれたスクリプトに出会うので、できればコンピュータ上で実行してほしい。見慣れた庭の風景が違って見えるはずだから。

庭いじりの前に言っておきたいことがある。それは、この作業で数学やプログラミングが“習得”できるわけではない、ということだ。数学について何かを習得しようと思ったら、それなりの数学の書物をそれなりに読まなくてはならない。プログラミングの習得には、それなりの経験が必要だ。と、こんな風にえらそうに書き出してはいるが、私は数学の専門家ではないしプログラミングの専門家でもない。そんな者がこんな物を書くのはお門違いかもしれないが、少なくとも何も知らない人より、多少の経験はあると思う（思い過ごしかもしれないが）。しかし、専門家でないことは事実なので、随所に専門家の目に耐えかねる部分が存在するのも確かだが、あからさまな間違いを書いているつもりはないので安心してほしい。

今回は家の前に足を伸ばすだけなので、面倒な数式が登場しても厳密な議論をしていない。厳密な議論がないので、数学という庭をいじりはするが、決して本格的なガーデニングなどするわけではない。海外旅行に出かけても、観光はするが現地の生活に溶け込むことがないのと同じだ。また **Haskell** のスクリプトについては、単純なことしか処理しないものになっている。庭は猫の額ほどしかないし、庭いじりは仕事ではないので、ときに変なスクリプトに出会うかもしれないが勘弁してほしい。

早い話、私は上っ面だけをなめているに過ぎないので、数学や **Haskell** について詳しく学びたいければ、それ相応の書物を手元に置いておくことを勧める。細かいことや厳密なことや正統な作法は、是非そのような書物から学んでほしい。私は単にきっかけを与えることができれば満足なのである。



**Haskell** はコンピュータにインストールして利用する。インストールのプログラムは Internet から無料でダウンロードできる。**Haskell Platform** をダウンロードするのが手っ取り早いだろう。インストールについて詳しく述べることはしないが、特別難しい操作があることはないので大丈夫だろう。ただし、テキストエディタは必要である。でも、Windows や Macintosh に標準で付属しているもので十分だ。

**Haskell** は巷（ちまた）で広く使われているプログラミング言語に比べると少々異なるところが多い。でも、数学的な考え方をそのままスクリプトに置き換えられるような自然な言語である（巷で広く使われている言語から見たら不自然かもしれないけれどね）。しかし、スクリプトを書くには多少の知識がいる。また、Unix（もしくは MS-DOS）のコマンドの基本的使い方は知っていてほしい。詳しいことは、**Haskell** について書かれた書物を読んでもらうか、WEB サイトから情報を得てもらうしかないが。

プログラミングには関心がないという人には、無意味な庭いじりを強いているように感じるかもしれない。だが、そうではない。プログラミングというのは、ソフトウェアを開発する一部の人だけに必要な素養ではないのだ。もちろん、誰もが大規模なソフトウェアの開発などしない。しかし、コンピュータソフトは誰もが使うはずだ。特に、表計算ソフトやデータベースソフトでは、たとえプログラミングをしなくとも、プログラミング的思考ができるのとそうでないのとでは、できる仕事や作業に雲泥の差がつくのだ。電子レンジを使うだけでも食事はできるが、フライパンや鍋を自在に使えてこそ質のよい食事ができるのと似ている。電子レンジを使うだけのレベルがソフトウェアを使うだけのレベル、プロの料理人のレベルがソフトウェア開発を職業とするレベルとするなら、家庭で料理を作るレベルがプログラミングができるレベルと言えるだろう。

今回は限定された場所を行き来するだけなので、ほんの短いスクリプトしか書かない。そして、スクリプトで使われる数値や入力される数値は限定的である。だから何も知らずに勝手な数を入力すると、簡単にエラーを引き起こしてしまうかもしれない。間違った入力に対しても堅牢な動作を保つには、もうちょっと **Haskell** に詳しくならなければいけないのだが、そうはしていない。また、**Haskell** のバージョンによっては、動作や計算結果が違ってしまうことがあるかもしれないが、書いたスクリプトは自分の責任で楽しんでもらいたい<sup>1</sup>。つまり、物好きが道楽でする気ままな庭いじりである。それでは早速、庭に下りてみよう。

---

<sup>1</sup> 本文で示したコードと実行結果は、GHCi, version 8.6.5 / macOS Catalina, version 10.15 による。