

# 「Excelの次」としてのノーコード

角南 北斗<sup>\*1</sup>

Email: hello@shokuto.com

\*1: フリーランス (ウェブデザイナー)

◎Key Words Excel, データベース, ノーコード

---

## 1. はじめに

発表者は、大学で初学者向けの情報の授業を10年以上担当している。その授業でExcelを扱う際、データベース形式のデータ処理の方法も紹介しているが、その有用性にピンとこない受講生が多い、という印象が担当としてはあった。実用的な例を示そうと、Wordの差し込み印刷と連携した宛名印刷を例にして説明してきたが、最近は印刷物の作成自体が身近ではない受講生が大半になっていて、例にするには限界を感じていた。

そこで2022年度は、ノーコード (NoCode) によるアプリ開発を例に、受講生にも身近なアプリとデータベースの関係性をイメージしてもらう試みを行った。本発表は、その教育実践を紹介し、課題も含めて知見を共有するものである。

## 2. 実践の概要

本実践を行った授業は3つで、主な受講生はそれぞれ1年生、2年生、4年生である。いずれも、授業の前半ないしは別の授業でExcelの基本的な部分は学習済みで、Excelのデータベース活用の代表例であるテーブル機能もそれに含まれている。

授業全体の学習目標は、授業によって多少異なるものの、Excelを使ってデータ処理の基礎的な考え方を身につけるという範囲は超えない。また、アプリ開発やプログラムに関する特段の知識・関心を受講生は持ってはいない、という前提で授業を設計している。

限られた時間でアプリ開発を経験してもらうためのツールとして、本授業ではGoogleのAppSheet (appsheet.com) を採用した。AppSheetを扱ったのは15週のうち4週程度で、スマートフォンの標準アプリとして用意されているようなベーシックなメモ帳アプリを作ってもらった。最初に最低限の構成のデータベースをGoogleスプレッドシート

(以下スプレッドシート) で作成してもらい、AppSheetにそれを読み込ませてアプリを自動で組み上げてもらう。データを追加したり、データベースのカラムを増やしたりしながら、機能の追加と表示の調整をしてもらうといった手順で進め、最後に各自のスマートフォンで (AppSheetアプリを使って) 作ったアプリを動作させるというものである。

受講生は多いクラスで30名を超えていたため、担当教員一人で対応できるか不安も大きかった。しかしながら、個別のトラブルシューティングは一部必要だったものの、大半の受講生は発表者の用意した配布資料に沿って進めてもらう形で、アプリを完成させることができた。

## 3. AppSheetを採用した背景

アプリ開発の手段としてノーコードと呼ばれるサービスを用いたのは、受講生はプログラムに対して知識や関心がない、開発のための特別な環境を用意できない、時間が限られる中でも一定の達成感を持たせたい、といった背景がある。

担当教員である発表者自身も高度なプログラムスキルがあるわけではないが、さまざまなサービスを活用すれば「プログラムが書けなければできないような高度なこと」を実現できる、自分の可能性が広がる、ということを学生時代から体験してきた。受講生にも、そうした体験を今のうちにしてもらいたいという思いが発表者にはある。

ノーコード系のサービスはAppSheet以外にも数多く存在する。その中でAppSheetを採用した主な理由としては、受講生にとって身近なスプレッドシートをデータベースとして使えること、データベースからのアプリ化をほぼ自動でやってくれること、日本語の書かれた解説が (他のノーコード系

サービスと比較して) サイトや書籍などで多く出回っていること、などが挙げられる。

もっとも、ノーコードを使えば専門知識ゼロでもすんなりアプリが作れるのかというと、実際はそう単純な話ではない。AppSheetは確かに「コードを書かずに」アプリを開発できるツールとして設計されているが、データベースのカラムに対するデータ型などの設定は必要であるし、データの抽出には関数を含む数式を記述する場合もある。エラーメッセージを読み解くにはプログラムに対する理解も不可欠である。プログラムそのものは記述しないで済むとしても、画面上での設定操作は基本的な開発の知識を求められる。また、これはAppSheetに限ったことではないが、いざ自分でアプリを作るとなると、インターフェースや画面遷移の役割やパターンを認識し直す必要があり、これもなかなか難しい。

それでも短時間で「自分が作成したデータベースと連動したアプリを作る」という経験がしやすいツールとして、発表者はAppSheetを選んだ。

#### 4. 教材としてのAppSheetの難しさ

AppSheetがツールとして優れているのは、適切なデータベースを設計すれば自動でアプリを生成してくれ、しかもその完成度が高いことである。そのため教師が適切なスプレッドシートを受講生に配布すれば、開発工程を短縮でき、受講生がトラブルに陥るケースも回避しやすい。表示のカスタマイズ要素もさほど多くないうえ、いずれを選んでもそれなりに整ったものになるため、受講生が試行錯誤でアプリを壊してしまう危険性も抑えられている。こうした長所は、時間が限られる授業での利用にも向いているといえるだろう。

ただし同時に、受講生が「手順通り作業してアプリを完成させること」を目標にしてしまいやすい、という面もあると感じた。データベース形式のデータ処理に対する理解を深めるのが授業の目的であるので、例えば「ビジュアル面を好みに変える」といった作業ではなく「データが思ったように表示されないことから原因を考えて改良していく」といった経験を受講生にはしてもらいたい。それを意図的に授業で起こす場合は、AppSheetは適切にアプリ化してくれないようなスプレッドシートを受講生に配布したり、スプレッドシートをゼ

ロから作っていくような授業設計が必要だろう。つまり皮肉なことに、ツール自体の目指すことと逆行するような仕掛けが教師には求められる。

世の中にあるAppSheetの開発工程の解説資料のほとんどは、目指すアプリが具体的にイメージできる・提示されていることや、アプリ化したいデータが手元にあることが前提となっている。そのため当然ながら、アプリを完成させる方法の解説に力点が置かれている。一方で本授業の受講生は、作りたいアプリや活用したいデータが具体的にあるわけではない。アプリを完成させることよりもデータ処理の概念を理解することが実践の目的であるため、授業でそうした資料をそのまま使ってしまうと「作ること」だけに注意が向きやすい。

いくつかアプリを作っていくことで概念に対する理解も深まっていくことは考えられるので、そのようなアプローチも必要だとは思うものの、授業全体の内容のバランスを考えると、アプリ開発に大幅に時間を割くというのは難しい現実がある。

#### 5. 次年度に向けて

前年度に引き続き、2023年度の後半でも、Excelの基本的な学習を行なった後のステップとして、ノーコードを使ったアプリ開発を授業に組み込む予定である。現状は(上述のように課題もあるものの) AppSheetを採用するつもりだが、この種のサービスはアップデートによって設計や機能が大幅に変わることもよくあるので、他のサービスを試しながら準備をしたい。

一例を挙げると、同じくノーコード系のサービスであるGlide (glideapps.com) は、AppSheetよりも画面設計に柔軟性があり、作れるアプリも幅広いという印象を持っている。2022年度は、4年生が主体のクラスでAppSheetのあとにGlideを取り上げたが、受講生からは「ツールを理解するまでが難しいが、Glideの方が自由度が高く面白い」というコメントも多く寄せられた。もっとも、こちらのサービスもこの1~2年で大きく機能が変更されているため、同じ懸念材料はある。

発表では、プログラム教育などに詳しい参加者から、データベースを使ったデータ処理の理解を促せるような初学者向けの切り口、ツールなどを教えていただき、場で共有できればと考えている。