オンデマンドコンテンツと同時双方向型授業を組み合わせたプログラミング教育の実践

授業実践

オンデマンドコンテンツと同時双方向型授業を組み合わせたプログラミング教育の実践

Practice of programming education by combining on-demand contents and simultaneous interactive classes

岡田龍太郎
武蔵野大学データサイエンス学部

概要

本稿では，プログラミング教育における，オンデマンドコンテンツを用いた学習と同時双方向型の演習を組み合わせたハイブリッド型授業の実践事例について述べる．オンデマンドコンテンツを利用することで効率的な学習が可能となった一方で，授業資料が外部コンテンツに依存性を持ち再利用性が下がるという問題があることを示した．また，授業資料の一般公開を行う際には，そうした依存性を低減させた上で，明確なテーマを持つ題材を定めて作成することが必要であることを示した．

キーワード： オンデマンド型授業，Python，授業資料の一般公開

1. はじめに

近年のプログラミング教育の需要の高まりもあって，現在，Web 上のオンデマンドコンテンツを用いてプログラミング学習が出来るサービスが多数存在しており，またその数も増えてきている．このようなサービスの例としては，例えば本稿で着目する Python 言語を取り扱うものに限定しても，PyQ[1]，Paiza[2]，Progate[3]等の多数のサービスが挙げられる．こうしたサービスは，企業の研修で用いられるのはもちろんのこと，学校教育でも利用されるようになってきていている．こうしたオンデマンドコンテンツは，大学でプログラミング教育を行う際にも活用可能である．

また現在，多くの大学で，授業資料をオンラインで一般公開し，誰でも利用可能にすることが行われつつある．大学の授業の資料や講義の映像を公開するのは大学の社会貢献活動の一環と捉えられており，Python 言語の学習教材に限っても，東京大学[4]，京都大学[5]，中央大学[6]，筑波大学[7]など様々な大学が資料を公開している．

武蔵野大学では情報副専攻コース(AI 活用エキスパートコース) [8]という，AI やデータサイエンスのスキルを身につける授業群を展開しており，その中にはプログラミング教育

も含まれている．筆者は武蔵野大学の全学向けのプログラミング教育を行う授業の一つである「プログラミング発展 A」という，Python 言語を学習する授業を実施した．さらに，授業用に作成した資料に一部手を加え，独習用の教材として一般公開した．

本稿では，プログラミング発展 A において実施した，オンデマンドコンテンツを用いた学習と同時双方向型の演習を組み合わせたハイブリッド型授業の詳細について述べる．またさらに，授業資料を一般公開するにあたって考慮した点について考察を述べる．

2. 授業「プログラミング発展 A」におけるハイブリッド型授業の実践

本節では，筆者が本年度実施した授業である「プログラミング発展 A」という授業について詳細を述べる．なお，以下では「プログラミング発展 A」のことを略して「本授業」あるいは「発展 A」と記す場合がある．

2.1 授業の位置づけとねらい

プログラミング発展 A は本年度初めて開講された授業である．本授業は，「プログラミング基礎」という授業を受講済みであることを前提としている．プログラミング基礎の授業運営における取り組みは文献[9][10][11]に示されている．プログラミング基礎は，プログラミングに初めて触れる人を対象とした授業で，Minecraft: Education Edition[12]をプラットフォームとして，その中で Microsoft MakeCode[13]を用いたプログラミングを教える授業となっている． MakeCode はビジュアルプログラミング言語の一種で，ブロックを組み合わせてプログラミングを行うものである．プログラミング基礎の中では，基本的なプログラミングの文法として，制御構文(if や for にあたる)，変数，配列などを教えている．また，作りたいものをプログラミングによって具体化するための考え方を学ぶために，個人またはグループでミニプロジェクトを実施し，計画から実装までの一連の流れを体験させている．その中で，フローチャートの書き方も教えている．

プログラミング発展 A は Python 言語を学ぶ授業である．プログラミング基礎でプログラミングの初歩を学んだ学生に対して，広くソフトウェア制作の現場でも使われている本格的なプログラミング言語である Python 言語を教えることを目的としている．また，本授業ではただ Python 言語の文法を学ぶだけではなく，プログラミングを用いた問題解決の一手法として，確率的な事象をモデル化し，数値的なシミュレーションを行う方法について解説している．さらに，そのためのプログラミングを行うに当たって，関数を用いた比較的規模の大きいプログラミングの方法について解説している．

本授業のねらいは以下の 3 つに整理される．

- Python 言語を利用可能にする
- 関数を用いた大規模なプログラムを書けるようにする
- プログラミングを用いた数値シミュレーションを可能にする

2.2　授業の環境

本授業は，オンデマンド教材による自主学習とオンラインによる同時双方向型授業のハイブリッド授業として行われる．以下，2.2.1 節では利用したオンデマンド教材について述べる．2.2.2 節では Python 言語のプログラミング環境について述べる．2.2.3 節では，それらの授業の実施形態について述べる．

2.2.1 オンデマンド教材

Python 言語を学ぶためのオンデマンド教材としては Progate[3]の Python コースを用いた．Progate はブラウザ上でプログラミング言語を学べる Web サイトであり，Python 言語以外にも様々なプログラミング言語を学ぶことができる．Python 用のコースは 5 つのレッスンが用意されている．レッスンの中身は細かい単位に分割されていて，一つのレッスンを終えると次に進める形となっている．プログラムを入力し実行できる環境が用意されており，解説の後に実際にコードを書いて実行し，正しい出力が得られていれば合格となる．レッスンの初めに完成形が示され，一つの演習では数行ずつ追加・変更していき，一つのレッスンを終えると一つの意味のあるプログラムが完成する形式になっている．

2.2.2 プログラミング環境

同時双方向型の授業では，Python 言語を記述・実行する環境として，Google Colaboratory(以下，Colab)を利用した．Colab は，Jupyter Notebook という形式のファイルをクラウド環境で実行できるようにしたものである．Colab は Google が開発しており，自身の Google アカウントから利用でき，Google Drive と連携してファイルを操作することも可能である．Jupyter Notebook 形式のドキュメントとは，Python を含むいくつかの言語をインタラクティブに利用可能にしたもので，プログラムコードの他に，Markdown テキスト，数式，図式等を含むことができる．Jupyter Notebook 形式のファイルはセルという単位の連なりとして構成されており，各セル単位でプログラムコードや Markdown テキストを実行することができる．Jupyter Notebook 形式は，複数のプログラムコードを一つのファイルにまとめることができる点や，Makrdown テキストによる解説とプログラムの実行を織り交ぜて記述することができる点が優れており，授業にも使いやすいものとなっている．

2.2.3 授業の実施形態

本授業は，まず Progate によるオンデマンド授業を行い，その後に同時双方型授業でその詳細の解説を行うとともに，発展的な課題に取り組むという形態で行った．

オンデマンド授業では，Progate での課題をこなした後に，習熟度を測るテストを受けてもらった．テストは Progate 内の課題とは別に，選択式のクイズ形式のものを作成した．

同時双方向型授業はオンラインで行った．同時双方向型授業は，Remo Conference[14](以下，Remo)というオンライン会議を行うことを主目的としたツールを用いて行った．Remo では，8 人までが同じテーブルに参加することができ，参加しているメンバーはそれぞれビ

デオ通話ができるうえに，各自が同時に画面共有をすることができる．受講生は常に作業中の画面を画面共有しておくことによって，自分から助けを求めなくて講師の指導を受けることができる．

授業の講師は，メイン講師である著者に加えてサブ講師 1 名の計 2 名であった．メイン講師は用意した Jupyter Notebook 形式によるに教材を順に解説する．教材には途中に練習問題が含まれており，最後には演習課題がある．演習課題は授業外の時間も使って回答する．サブ講師は同時双方向授業の中で，学生の様子を確認し，指導及び質問への対応を行った．

授業の受講生との授業に関する連絡や，オンデマンド授業のテストは Microsoft Teams を用いて行った．

2.3 授業内容の構成

本授業の構成を表 1 に示す．

表 1　プログラミング発展 A の構成.

Table 1 Overview of Programming Development A.

実施回	オンデマンド	同時双方向
1	(初回授業以前の課題はなく，第 2 回授業までの課題とまとめる)	イントロダクション，Python 基本文法，Python ライブラリの紹介
2	Progate レッスン 1 (基本文法，print，変数，if)	print，数値計算，比較演算，Markdown
3	Progate レッスン 2 (for, while,配列，辞書)	リスト，for, while，グラフ表示(Matplotlib)，地図上へのプロット
4	Progate レッスン 3 (関数)	モデル化とシミュレーション(1)，関数
5	Progate レッスン 4 (クラス)	モデル化とシミュレーション(2)，ファイルの入出力, 基本的なライブラリ(NumPy, Pandas, Seaborn)
6	いくつかの Python ライブラリの体験(Jupyter Notebook の配布による)	モデル化とシミュレーション(3)
7	(ミニプロジェクトの課題)	ミニプロジェクト

本授業の最大の特色となるのは，4 週目から 6 週目にかけて行われた「モデル化とシミュレーション」である．この課題では，確率的な事象に対して，モデル化を行い，数値シミュレーションを行うことで予想を行う一連のプロセスを体験する．こうした手法は一般的にはモンテカルロ法と呼ばれる．具体的には，「百発百中の砲一門」と「百発一中の砲百門」が打ち合った時にどちらが勝利するか，大砲の数や命中率を変化させた時にどういう風に

結果が変化するのかを，プログラミングを用いて計算する．この課題では，現象をモデル化しプログラミングで表現することの他に，大規模なプログラミングを書く体験を提供することを目的としている．

ミニプロジェクトでは，自分でテーマを決定し，データ分析あるいはシミュレーションを行うこととした．

2.4 授業の実施と振り返り

本授業の受講生は 4 名と，想定より非常に少なかった．各受講生と教員 2 名は Remo の一つのテーブルに収まったので，学生が画面共有している画面を教員はいつでも確認することができ，きめ細やかな指導をすることができた．一部の局面ではサブ講師が Remo の別のテーブルに学生を連れ出して質問に対応することもあった．Remo を利用した指導は上手く機能していたと言えるが，受講生が少なかったためサブ講師と分担して多くの学生を見るという実施形態を試すことは出来なかった．アンケート結果では，画面を常時共有しておいて細かく指導をもらう形態について，質問しにくい場面でも指導がもらえる点について好評の声があった．

モデル化とシミュレーションの課題は，受講生からは好評であったことがアンケートから伺えた．ただし，本授業はプログラミング基礎を受講し，プログラミングに特に興味を持った学生が受講してきていていると考えられ，また受講生の中には高校までにプログラミングを経験してきた人もいるなど，もともとプログラミングが得意な学生が主であったと考えられる．そうした学生が集っていたにも関わらず，講師が付きっきりで指導をしてようやく課題を完成させることが出来たという状況であった．そのため，受講生が増えた場合には，指導が細やかに行えないこと，そこまでプログラミングが得意でない学生の割合が増えることが想定されるため，現状よりは難度を下げることを検討すべきであると考えられる．

ミニプロジェクトは一週のみの取り組みとなっていたため期間が短く，学生の取り組んだ課題も小規模なデータ分析にとどまっていた．モデル化とシミュレーションについての課題に取り組もうとした学生もいたが，計画段階で相談を受け，授業期間内に作成するのは難しいと判断し，別の課題に変更させた．モデル化とシミュレーションの課題をミニプロジェクトとして行うには，小規模な課題をこちらで候補として用意するなどして学生の負担を軽減する必要があると考えられる．

その他，アンケートでは，実践的な知識が得られたといった意見や，もっと他の学生と交流したかった，といった意見が挙げられていた．

2.5 オンデマンドコンテンツと同時双方向型授業の組み合わせに関する考察

本授業では，授業内容をまずオンデマンドコンテンツを用いて学習し，同時双方向型授業でそれを確認しさらに発展的な課題に取り組むという形態を採用していた．プログラミング教育において，初歩の段階では基本的な記法を学ぶことが必要であり，その内容はプログ

ラミング言語ごとにある程度共通しているため，オンデマンドコンテンツを利用してその段階の学習をすることは有効であると考えられ，本授業においても同時双方向型授業での解説の時間を大幅に短縮することが出来た．

学生の立場から考えると，オンデマンドコンテンツを用いることは，自由な時間に学習することが出来るメリットがある一方，モチベーションが低下することや，分からないことがあった時に相談しにくいといったデメリットがあると考えられる．学生へのアンケートでは，「オンデマンド学習の確認テストを利用して，Progate で学んだことの理解度をチェックできた」という項目では「5，4，5，5」と 4 人中 3 人が最高評価の 5 を付けており，独習してその理解度を把握するという点においては上手く機能していたことが伺える．「オンデマンドではなく，プログラミング基礎と同じように 2 コマとも同時双方向（ライブ）で授業を受けたかった」という項目では「3，3，1，2」となっており，オンデマンドコンテンツを用いた授業形態に特に不満はなく，むしろ望んでいる人もいることが伺える．ただし，「オンデマンドではなく，プログラミング基礎の「オンライン演習室」のような教員のサポートを受けたかった」という項目では，「4，4，3，1」となっており，分からないことがあった時に相談したいという欲求はあることが伺える．

また，今回用いた Progate ではブラウザ上で作成したコードを実行することが出来るため，学生は環境構築の手間なく学習を行うことが出来た．初学者にとってプログラミングの環境構築はハードルが高いため，この点でも負担を軽減することが出来たと言える．同時双方向型授業においても環境構築の必要ない Colab を利用することで，授業全体としても環境構築を必要としない授業となった．このことは負担軽減という意味では良いが，学生に環境構築の方法を教示する機会を失っているとも言える．しかしながら，現在では Colab だけを用いたプログラミングでもデータ分析等の問題解決が行えるケースは多くなっており，そのデメリットは相対的に減少していると考えられる．

本授業で独自に用意したコンテンツは，Progate での学習を前提としているため，教える内容の順番は Progate に準拠したものとなり，また内容的には Progate の内容と相補的になるように構成することとなった．このことは，効率的な学習を実現した一方で，作成した資料が外部コンテンツに依存する形になってしまったことをも意味している．そのため，資料の再利用性が低くなるという問題を抱えることとなった．この点については改善の余地があり，例えば単体として成立する完全な資料を構成した上で，外部コンテンツに合わせて省略する部分を決定するといった運用を行うことが考えられる．

3. 独習用教材としての展開

本授業で作成した資料を，筆者が所属するデータサイエンス学科向けの教材として展開し，さらに一般に広く公開する教材として展開した．

3.1 データサイエンス学科の勉強会での利用

武蔵野大学データサイエンス学部データサイエンス学科では Python 言語の使い方を授業

で教えており，発展 A の授業で用意した教材が復習や発展課題の教材として使えるのではないかと考えた．そこで，2021 年の 8 月に，データサイエンス学科の学生向けに，本授業の教材を用いた勉強会を 3 回に分けて開催した．勉強会は Zoom を用いてオンラインで行った．参加者は日によって少し変動はあったが計 15 名程度であり，1 年生の割合が高かった．かかった時間は，3 日間でそれぞれ 3 時間，3 時間，5 時間の計 11 時間ほどであった．

勉強会を行うに当たって利用したのは Colab で動く Jupyter Notebook のファイルだけであり，開催にあたっては資料を見直して，解説や課題をより充実させた．また，データサイエンス学科の中でもプログラミングが得意な学生数名に事前に資料を確認してもらい，改善点について意見をもらって反映させた．さらに，実施にあたっては学生一名にアシスタントとして協力してもらい，解説の補足などをしてもらい，さらにそれを事後に報告してもらい，資料の修正を行った．

データサイエンス学科の学生は普段から Python 言語によるプログラミングの例には触れているため，授業で用いた資料からは導入のための例などを除き，「Python 基礎講座」という名称の資料としてまとめ，その名の通り基礎固めに用いるための資料とした．

参加した学生からは好評の声が聞かれたが，任意参加の勉強会であるため，参加者数自体は多くなかった．資料は再利用できるので，資料を学生に独習用教材として公開しいつでも利用可能とした．今後は再度勉強会を開くことを計画している．

3.2. 独習用資料の一般公開

勉強会のために資料を精査したことで完成度を高めることが出来たので，筆者の researchmap 上で独習教材として一般公開した[15]．公開にあたってはさらに内容を精査し修正するとともに，解答のファイルを追加した．

公開の告知は著者の SNS アカウントで個人的に行った．本学学生の他に，他大学の学生や，その他一般の人から利用の報告を受けた．

4. プログラミング学習における教材作成についての考察

本稿ではこれまで，授業用に Python 言語を習得するための教材を作成したこと，それを独習用教材として展開したことを述べてきた．これらの経験から，大学の授業でプログラミングの教材を作成する上での考慮点について考察していく．

まず，今回の発展 A の授業では，Progate をオンデマンド教材として利用し，それと相補的になるように Colab 上で動く Jupyter Notebook 形式の資料を作成した．このことにより，作成した資料は次のような特徴を持つこととなった．一つは，完全な初心者向けの資料ではなく，最低限の記法は理解している人を対象とした資料となっていることである．もう一つは，説明の順番が Progate のコンテンツに依存しているということである．これらの特徴は，授業にはフィットするというメリットがあるが，独立の教材として用いる場合には少なからぬデメリットを生んでいる．また，Progate では繰り返し処理より先に条件分岐を解説

しているが，筆者は順番は逆の方がデモを示しやすくて良いのではないかと考えていた．今回のケースでは，順番を変更することはなかったが，初回授業にはただ実行するだけで動くプログラムの形でデモを用意し，その中で繰り返し処理についてのデモを見せることで対応した．

発展 A の資料を作成するにあたって一番難航したのは演習課題の作成であった．プログラミング基礎では Minecraft を利用していたので，プログラミングをした結果を Minecraft の世界に反映させることで視覚的にも分かりやすいフィードバックを得ることができた．しかしこの点において Python 言語は，ライブラリ等を使わない場合は文字を出力するだけであり，分かりやすいフィードバックを得ることが難しかった．モデル化とシミュレーションの課題ではシミュレーションの結果が出力されるのでそうした問題はなかったが，基礎的な部分の説明においては単調な印象を避ける工夫をしていく必要がある．

授業資料をデータサイエンス学科での勉強会の資料として活用した点については，大きな問題は無かったと考えている．もともとデータサイエンス学科の学生は Python 言語でのプログラミングに触れたことがあり，発展 A の授業で Progate を用いて基本的な文法を理解した状態で受講するという状況と近かったことで，スムーズに教えることが出来た．

この勉強会で用いた授業資料を独習用教材として一般に公開した件については，社会的インパクトとしては軽微なものになってしまったと考えている．その要因としてはいくつかあり，まず，東京大学が同様に Colab を用いた Python 言語プログラミング学習のための資料を既に公開している[4]ことにより，一般の人からすればそちらを選択すれば良いことが挙げられる．また，一般向けの独習用教材としては初心者向けでない点もハードルを上げている上に，モデル化とシミュレーションに特化した教材としてのアピールもできていなかった．さらに，Progate のレッスンの順番に準拠していることも，単体の教材としての流れを阻害してしまっている．

これらのことから，今後プログラミング学習の教材を作る際には以下の点に注意すると良いと考えられる．

- 初心者向けの教材は商用のものも大学が無償公開するものも充実しており，学ばなければならないことも大部分が共通しているので，既存のコンテンツを積極的に活用する
- 新たに資料を作成する際には，明確なテーマを持つ題材を定め，内容はもちろん広報としても差別化できるようにする

5. おわりに

本稿では，プログラミング教育における，オンデマンドコンテンツを用いた学習と同時双方向型の演習を組み合わせたハイブリッド型授業の詳細について述べた．またさらに，授業資料を一般公開するにあたって考慮した点について考察を述べた．オンデマンドコンテン

ツを利用することで効率的な学習が可能となった一方で，授業資料が外部コンテンツに依存性を持ち再利用性が下がるという問題があることを示した．また，授業資料の一般公開を行う際には，そうした依存性を低減させた上で，明確なテーマを持つ題材を定めて作成することが必要であることを示した．

謝辞　プログラミング発展 A の責任教員として授業の構成や運営について様々なご助言をいただいた渡邊紀文准教授に感謝する．

参考文献

[1] PyQ | Python で一歩踏み出すあなたのための，独学プラットフォーム https://pyq.jp (参照 2022-2-22)
[2] 環境構築不要！初心者でも楽しく学習できるプログラミング入門サービス【paiza ラーニング】https://paiza.jp/works/ (参照 2022-2-22)
[3] Progate | プログラミングの入門なら基礎から学べる Progate[プロゲート] https://prog-8.com (参照 2022-2-22)
[4] Python プログラミング入門 https://sites.google.com/view/ut-python/ (参照 2022-2-22)
[5] Kyoto University Research Information Repository: プログラミング演習 Python 2021 http://hdl.handle.net/2433/265459 (参照 2022-2-22)
[6] ベーシック講座 1｜m.PIME 本物の STEM 教育 https://lecture.m-pime.jp/category/basic01/ (参照 2022-2-22)
[7] Python ゼロからはじめるプログラミング」サポートページ https://mitani.cs.tsukuba.ac.jp/book_support/python/ (参照 2022-2-22)
[8] 情報副専攻（AI 活用エキスパートコース）https://www.musashino-u.ac.jp/guide/facility/MUSIC_center/submajor_aiexpert.html (参照 2022-2-22)
[9] 圓崎祐貴(2021): ”演習形式授業のオンライン化についての考察と取り組み”，Musashino University Smart Intelligence Center 紀要，Vol.2, pp. 77−82
[10] 渡邊紀文(2021) ”問題解決を重視したプログラミング教育とオンラインでの実践”，Musashino University Smart Intelligence Center 紀要，Vol.2, pp. 60-67
[11] 岡田真穂(2021) ” プログラミング科目におけるオンライン演習室の取り組みと考察”，Musashino University Smart Intelligence Center 紀要，Vol.2, pp. 68-76
[12] ホームページ | Minecraft Education Edition https://education.minecraft.net/ (参照 2022-2-22)
[13] Microsoft MakeCode https://www.microsoft.com/ja-jp/makecode (参照 2022-2-22)
[14] Remo conference: https://remo.co/conference/（参照 2022-2-20）
[15] 岡田 龍太郎 (ryotaro okada) - マイポータル - researchmap https://researchmap.jp/blogs/blog_entries/view/568385/73435bc281f7282a4db5092725bde00c （参照 2022-2-20）