

特集ショートペーパー

科目「プログラミング基礎」における対面授業 とオンライン授業の比較

河村 一樹^{1,a)}

受付日 2021年5月25日, 再受付日 2021年9月12日,
採録日 2021年10月30日

概要: COVID-19の影響により, 本科目は, 2020年度春学期は開講せず, 秋学期はZoomによるオンライン授業で実施した. 2019年度まで実施していた対面授業をオンライン授業に移行したことで, 学習効果はどうなったのか, クラスの進捗状況やアンケートによって比較検討した結果について報告する.

キーワード: COVID-19, 対面授業, オンライン授業, Moodle, Zoom

In the Subject “Programming Basics” Comparison of Face-to-face Lessons and Online Lessons

KAZUKI KAWAMURA^{1,a)}

Received: May 25, 2021, Revised: September 12, 2021,
Accepted: October 30, 2021

Abstract: Due to the influence of COVID-19, this course was conducted in the following form. Not open in the spring semester of 2020, and the fall semester was conducted as an online class by Zoom. By shifting the face-to-face lessons that were held until 2019 to online lessons, I report on what happened to the learning effect.

Keywords: COVID-19, face-to-face lessons, online lessons, moodle, Zoom

1. はじめに

Moodleを用いたプログラミング教育は, 各大学で実施されている [1], [2], [3]. ただし, いずれの場合も, 大学のコンピュータ実習室において, 対面授業で行われている. しかし, 2020年に発生したCOVID-19の影響により, 本学では2020年の春学期から全科目をオンライン授業に移行し, 秋学期からは新入生のゼミナールだけ対面授業とし, それ以外はオンライン授業を継続する形で実施した.

オンライン授業を行うためには, 専用のオンラインビデオ通話サービスを導入する必要があるが, 本学ではZoom [4]を採用した. これに合わせて, 教職員には, Zoomの有料アカウントが配布された.

また, 本学では, 2016年度から全学レベルでMoodle [5], [6]の利用を開始するとともに, 実習科目についてはペアコマ制を導入した. ペアコマ制とは, 週2コマ連続での開講 (たとえば, 月曜日の3・4コマ) ではなく, 曜日をずらしての1コマずつの開講 (たとえば, 月曜日の3コマ目と木曜日の3コマ目) である. これによって, それまで実施していた講義1コマの後に実習1コマという一般的な授業形態にとらわれないことになった. そこで, 講義を廃止し, 自学自習ベースによる実習中心の授業に転換した. これによって, 教室では, 講義をせずに個別指導だけを行うとともに, 自学自習を支援するためにMoodleを有効活用することにした [7], [8].

筆者が担当する科目「プログラミング基礎」でも, 2019年度までは通常の対面授業であったが, 2020年度の秋学期 (春学期はすべての実習科目は開講せず) からオンライン授業に切り替わった. 当初は, Zoomの使い方に不慣れだっ

¹ 東京国際大学
Tokyo International University, Saitama 350-1197, Japan
^{a)} kawamurk@tiu.ac.jp

たことや Zoom 自体の機能が限定的であったため多少混乱したが、その後徐々に慣れてきた。そんななかで、Zoom のブレイクアウトルームに着目し、これを個別指導に使うことにした。つまり、学生 1 人ずつのブレイクアウトルームをランダムに設定し、筆者が順番に各ルームを回りながら個別指導を行うという試みである。こうすることで、学生は直接教員に質問できるとともに、Zoom の「画面の共有」機能を使うことで、学生の実習内容（含む、ソースコード）を閲覧でき、教員もきめ細かい助言や指導ができるようになった。

以上の形で、2020 年度の授業を実施したわけだが、その際に以前からクラスの実習進捗状況を Moodle のログデータとして取得していたので、これらを比較して評価することにした。授業の進め方（教材コンテンツや課題の提供と提出、個別指導）は 2019 年度も 2020 年度もまったく同じであり、違いは履修者数（2019 年度：49 人履修し放棄は 7 人、2020 年度：29 人履修し放棄は 6 人）と授業の形態（2019 年度：対面授業、2020 年度：オンライン授業）だけであった。

その結果、2019 年度と 2020 年度のクラス全体の進捗率において、2020 年度の方が若干低いことが明らかになった。これには、アンケートの結果から、オンライン授業における学生同士の教え合いがまったくできなかったことが考えられる。

2. 科目「プログラミング基礎」の概要

ここでは、2019 年度の対面授業と 2020 年度のオンライン授業において、ともに共通する事項について取り上げる。

2.1 授業の目標

科目名に「プログラミング」というキーワードが入っているが、職業人としてのプログラマの育成を目指しているわけではない。プログラミングを体験する中で、アルゴリズムを用いた論理的な思考力や問題解決力を育むこと、コンピュータをブラックボックス化せずにその可能性と限界を知ること、可能であれば Web サイトの公開や簡単なゲームの開発を目指すことにしている。

2.2 実習環境

実習で使うプログラミング言語は、JavaScript としている。JavaScript を選んだのは、1) 多くの Web サイトで使われているスクリプト言語であり初心者にとっても分かりやすく使いやすいこと、2) 特別な開発環境を必要としないこと、3) デバッガ機能がついていることなどからである。

1) については、Web ページが簡単に作れること、構文規約（変数宣言 var の扱い、ステートメントのセミコロンやフォーマットの扱い）が厳しくないことがあげられる。2) については、別途コンパイラを用意する必要がなく、ブラ

表 1 授業の概要
Table 1 Class outline.

取り上げるテーマ	JavaScriptの構文	課題数	課題番号
自学自習の進め方*1	なし	0	
画面への出力	document.write文	1	問5-1-1
変数の扱い	var, 代入文(=)	2	問5-2-1 ~ 問5-2-2
演算式の扱い	演算子, 代入演算子, ++, --	2	問5-3-1 ~ 問5-3-2
画面からの入力	prompt文, Math.floor関数, eval関数	5	問5-4-1 ~ 問5-4-5
選択文	if文, if else文, else if文, switch文	7	問5-5-1 ~ 問5-5-7
繰り返し文	for, while文, do while文, break文	11	問5-6-1 ~ 問5-6-11
配列の扱い	new Array, 連想配列	11	問5-7-1 ~ 問5-7-11
関数の扱い	function文, 引数/戻り値	6	問5-8-1 ~ 問5-8-6
整列のアルゴリズム	選択/バブル/挿入/マージ/シェル/ヒープソート	6	問6-1-1 ~ 問6-1-6
探索のアルゴリズム	線形/2分探索	4	問6-2-1 ~ 問6-2-4
再帰のアルゴリズム	階乗計算, ハノイの塔	3	問6-3-1 ~ 問6-3-3
JavaScriptの応用	HTML, CSS, ゲームプログラミング	6	問7-3-1 ~ 問7-8-1
合計		64	

*1: Moodleの課題ダウンロード/アップロード, テキストエディタ・ブラウザの操作とデバッグ, 評定

ウザを起動することで実行ができること、Windows PC だけでなく Macintosh でも実習ができることがあげられる。とくに、2020 年度は自宅での実習となり、Macintosh を使いたいという学生もいたので有効であった。3) については、初心者の学生にとっては構文エラーを見つけ出すことが難しいようである。そこで、ブラウザ起動後画面に何も表示されない場合に F12 を押せばエラー箇所とその理由が表示されるという機能は有効である。

なお、実習環境として、エディタは TeraPad（空白の半角か全角を表示可）を、ブラウザは Google Chrome（デバッガ機能あり）を推奨している。

2.3 授業で取り上げる内容

15 週 × 2 コマで実施する授業の概要は、表 1 のようになる。これは、自著の教科書 [9] に準拠した構成になっている。

学生には、科目ガイダンス時に、JavaScript の基本構文について取り上げている問 5-1-1 から問 5-8-6 までの評点が 10 点満点になるように実習を進めること（最低要件）、問 6-1-1 以降の課題提出を評価に加点することも伝えた。

2.4 Moodle の利用

自学自習を推進するためには、教材の提供だけでなく、いくつかの支援策が必要になる。そこで、Moodle の活動モジュールとリソースモジュールを活用することにした。具体的には、次のようになった [10], [11], [12]。

(1) 教材コンテンツの提供

自著のすべての章・節を PDF 化し、それぞれをリソースモジュールの [ファイル] としてアップした。なお、出



図 1 課題の提示と提出

Fig. 1 Presentation and submission of assignments.

版社にはこの件について事前に承諾を得ている。

これによって、学生はいつでも自学自習ができるようになることから、授業外でも実習が行えることを提示した。

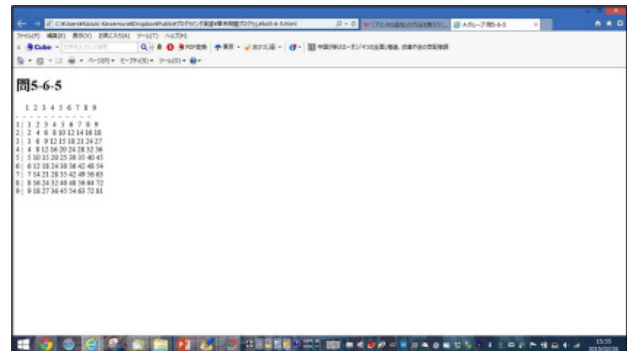
(2) 課題の提示

課題は、章・節ごとに用意してある。出題にあたっては、活動モジュールの [小テスト] の [作文問題] を用いた。図 1 にあるように、上部には課題に関する問題文 (日本語によるプログラム仕様) をコメントとして表し、その下部には、html のソースコードをテンプレートとして用意した。

学生は、教材コンテンツを自学自習後、課題の問題文を読み、html のソースコードをエディタにコピー&ペーストする。その上で、<h1> タグに自分の名前を入力し、<script> タグ内に課題に応じた JavaScript のソースコードを入力してからデバッグを行う。実行結果の画面を確認して OK であれば、完成したソースコードを張り付けるとともに、その html ファイルをドラッグしてアップロードを行うことで課題の提出となる。なお、名前を入力させているのはコピーを防ぐための本人確認であり、html ファイルをアップロードさせているのは教員の方でプログラムの動作確認をするためである。

(3) 課題のヒントと実行結果の提供

この授業では講義はせずに自学自習ベースによる実習が主になる。しかし、以前に教材コンテンツを講読するだけ



画面への表示は、上から下に向かって1行ずつとなる。このことを前提にすると、次のようなfor文の入れ子構造となる。

○上段の数字を表示する。そのために、for文(1から1ずつ9まで繰り返す)を使う。
 ○2段目のハイフンを表示する。そのために、for文を使う。
 ○3段目以降において、9×9の表を表示する。行(上から下方向)の数をi、列(左から右方向)の数をjとすると、for文(iが1から1ずつ9まで)の中に、for文(jが1から1ずつ9まで)を置き、その中でi*jを表示すればよい。ただし、i*jを表示するときに、i*jの値が1桁(1, 2, ..., 9)のときは全角の空白を、i*jの値が2桁(10, 11, 12, ..., 81)のときは半角の空白を、それぞれi*jの前に挿入すると、縦方向がきちんと並ぶ。

```

→→→→→(1から9まで)
1 2 3 4 ... 9
↓ 1 | 1⇨i*j(iが1, jが1)
↓ 2 |
↓ 3 |
↓ 4 |
↓ :
↓ 9 | 81⇨i*j(iが9, jが9)
i(1から9まで)
    
```

図 2 ヒントと実行結果

Fig. 2 Tips and execution results.

では課題ができないという学生の声もあったことから、すべての課題についてヒントと実行結果の画面を提供することにした。これらは、PDF として、リソースモジュールの [ファイル] で章・節ごとにアップした。

ヒントは、文字どおり課題に則したアドバイス (構文の使い方に関する注意点、間違いやすい箇所など) を記述したものであり、できるだけ独力でプログラミングできるように配慮している。実行結果は、課題を実行したときにこういった画面が表示されれば合格であることを示す画面イメージとした (図 2)。

(4) 授業外での質疑応答

後述するが、授業中は教員対学生の 1 対 1 による個別指導を対面あるいはオンラインで行い、授業外は自学自習を推奨している。ただし、授業外において、どうしても分からないが生じる場合もありうる。そのために、授業外でも適宜学生からの質問を受けつける仕組みとして、活動モジュールの [フォーラム] を用いた。その際に学生 1 人と教員だけの [利用制限] をかけることで、1 対 1 のフォーラムで学生との質疑応答が可能になる (図 3)。

また、[フォーラム] では、テキストデータだけでなく添付ファイルもアップすることができるので、どうしてもうまくいかないプログラムについては html ファイルをアップさせた。そして、どこがおかしいかを学生に考えさせるようなアドバイスを返信した。

(5) 課題の評価

アップロードされた課題については、[評点] と [フィー

+ 質疑応答



図3 フォーラムによる質疑応答

Fig. 3 Q & A by forum.

ドバックコメント]をつけて次の授業までに戻した。評点は、合格(仕様どおりの実行結果)は10点、再提出(軽微なエラーあり)は5点あるいは(重度なエラーあり)は0点をそれぞれ入力し、未提出は空欄のままとした。また、10点未満の課題については、必ず[フィードバックコメント]をつけた。これによって、学生に間違いに気づかせ、独力で再提出できるようにした。

Moodleでは、[小テスト]に評点を与えると、自動的に[ナビゲーション]の[評定]に反映されるようになっている。これより、教員の画面では、クラス全体の進捗状況を把握できるようになる。このため個別指導の際には、あらかじめ各人の進捗をチェックするだけでなく、クラス全体の進捗状況をも確認しながら進めた。

3. 個別指導について

個別指導とは、自学自習ベースの授業では学生個々の知識習得レベルやプログラミングのスキルが異なるので、それに合わせた対策として、学生1人ずつと進捗確認と質疑応答を行うものである。2019年度の対面授業と2020年度のオンライン授業において、実施形態が異なる事項としては個別指導があげられる。具体的に2019年度と2020年度を比較すると、表2のようになる。

1)については、オンライン授業では場所を固定することなく通信環境さえ揃っていれば受講できることになる。このため、学生によっては自宅ではなく、移動途中でスマートフォンで個別指導を受ける学生もいた。また、自宅などで受講できることで通学の必要がないことは、学生にとって利点にもなっていたようである。

2)については、オンライン授業では学生(あるいは、家族)が所有しているデバイス(パソコン、タブレット端末)を使うことにしたので、学生にとっても操作に慣れており使いやすいという利点があったといえる。また、大学ではWindowsパソコンに限定されるが、学生が自宅で使っているMacintoshやiPadでも可とした。

3)については、対面授業では学生を1人ずつ呼び出し、教卓の横に座らせ、筆者と1対1の形で個別指導を行っ

表2 個別指導の比較

Table 2 Comparison of tutoring.

	比較事項	対面授業 (2019年度)	オンライン授業 (2020年度)
1)	学生の受講場所	大学の実習室	学生の自宅等
2)	使用するデバイス	大学のWindowsパソコン	学生個人のパソコンかタブレット端末 ※学生の希望によりMacintoshも可
3)	指導の場所	教卓の横	Zoomのブレイクアウトルームで実施 ※学生一人だけの設定
4)	指導の順番	学籍番号順	ブレイクアウトルームを「自動で割り当て」
5)	進め方	印刷したソースコードリストでの確認か、教卓のパソコンでの閲覧	ブレイクアウトルームの「画面の共有」での閲覧

たが、三十数名の履修者数では90分間で一巡しかできなかった。オンライン授業ではZoomのブレイクアウトルーム(時間制限なし)を用いた。その際に、ルームのメンバは学生1人だけとし、教員はホスト権限で各ルームに入室した。これによって、教員と学生1対1の対話ができようになった。

4)については、対面授業では学籍番号順に実施したが、授業アンケートにより「いつも授業の最後の方になってしまい不公平である」という意見があった。そこで、オンライン授業では、ブレイクアウトルームを「自動で割り当てる」にすることにした。これによって、ルーム1からアットランダムに学生が割り当てられるようになる。また、学生の通信環境が悪く途中で接続が切れてしまっても、再度ルームに「追加割り当て」ができるようになっており、中断することなく指導を続けることができた。

さらに、教員がすべてのルームを回った後に、質問したい学生にはブレイクアウトルームで「ヘルプを求める」をクリックするように指示した。これによって、教員(ホスト)に通知がくるので、その学生に対して個別指導が可能になる。ただし、教員が別のルームに入室している間、待ちが生じる場合がある。このように、オンライン授業では教員に対する質問にも制約が生じてしまうことになる。

5)については、学生のソースコードを確認する必要が生じる。そのため、対面授業では、学生に自分のソースコードを印刷したリストをチェックするか、教卓のパソコンで学生のID/PWでログインさせてプログラムをチェックしなければならず、手間がかかった。一方、Zoomのブレイクアウトルームでは「画面の共有」機能が使え、学生の画面を表示することで、直接指導することができた。

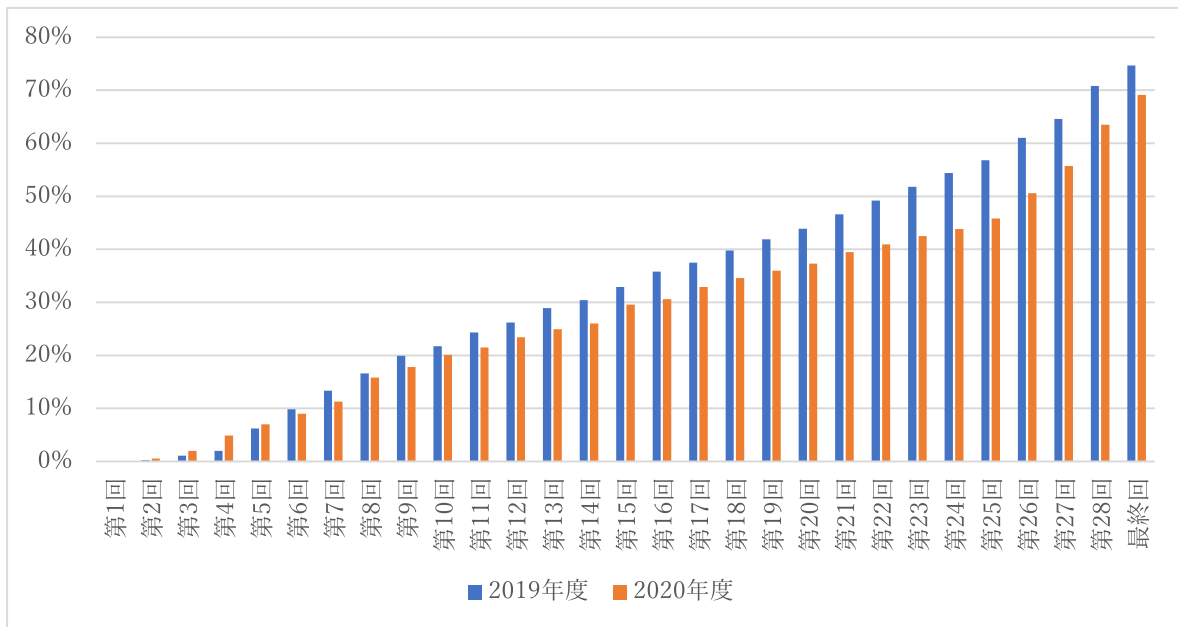


図 4 授業回ごとの進捗状況
Fig. 4 Progress for each lesson.

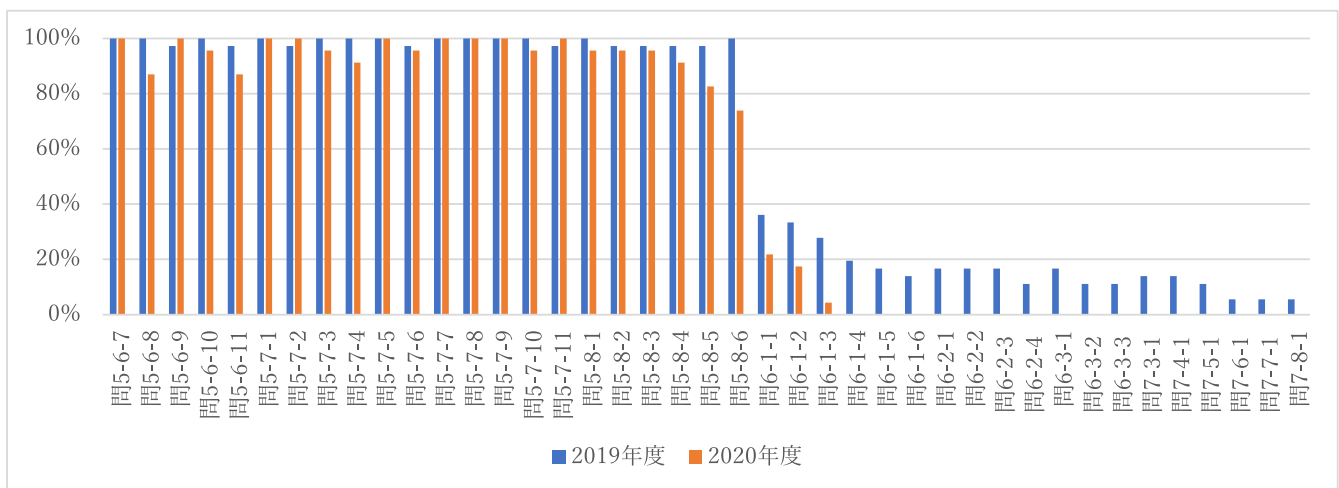


図 5 クラス全体の課題達成状況
Fig. 5 Achievement status of the entire class.

以上のように、オンライン授業で個別指導をどのように実施すべきかについては、Zoomのブレイクアウトルームを活用できることが分かった。

4. 実施結果と考察

Moodleの[小テスト]において、教員は「受験をレビューする」により学生からアップロードされた課題ごとに張り付けられたソースコードリストとhtmlファイルを参照することができる。

ソースコードを確認後、htmlファイルをブラウザで起動して実行する。「コメントを追加または評点を更新する」において、[コメント]と[評点]を入力する。これによって、[ダッシュボード]の[評定]に自動的に反映される。

以上の形で、次回の授業開始時前までに、すべての学生の評価を行った上でエクスポートしたcsvファイルをMoodleの[フォーラム]に公開した。

このcsvファイルをもとに、授業回ごとの進捗状況(図4)とクラス全体の課題達成状況(図5)を比較した。いずれも、10点満点となった課題数を累計している。なお、被験者数が年度ごとに数十名となっているのは、筆者だけの担当であること、実習室を使うので履修抽選による人数制限によるためである。

図4では、履修者数×64問(全課題数)を100%として、次回の授業までに提出された課題数を比率として算出しプロットした。これより、2019年度の方が2020年度よりも毎回の課題提出状況が高いことが明らかになった。な

お、第5回目までは2019年度よりも2020年度の方が、提出数が高かったが、その後逆転した。これは、第6回あたりから、課題のレベルがあがってきたことでオンライン授業では学生が1人で苦戦していたことが想定される。

図5では、問5-1-1から問5-6-6までは2019年度も2020年度も100%の提出状況であり、問5-6-7以降に差が生じてきたので、その部分をプロットした。これより、2019年度ではほぼ全員が問5-8-6まで終わるとともに、数名の(プログラミングの適性がある)学生がすべての課題を終えた。一方、2020年度は数名の学生が、多くても問6-1-3までしか達成できなかったことが明らかになった。

以上より、2019年度よりも2020年度の方が、クラス全体の課題達成状況が低いことが明らかになった。

次に、両年度の学期末に実施したアンケート(一部抜粋)について取り上げる。アンケートは、最終回の授業時に、Moodleの[フィードバック]モジュールを用いて行った。回答数は2019年度が33人(回収率:91%)、2020年度が12人(回収率:52%)となった。

Q1の「実習」には、Moodleにあるテキストの自学自習および課題のプログラミングとデバッグが含まれる。このため、授業中に講義をすることはなく、個別指導中心で進めていくことになる。

対面授業の場合、授業中に実習を行っている割合がほとんどであった。これは、個別指導を受けるまでの間に実習を続け、受けた後も実習を続けていたことによると思われる。一方、オンライン授業では3割程度の学生が授業中だけでなく授業外にも実習を行っていた。自宅などに実習環境があることから、時間(時間割枠)や場所(実習室)に限定されることなく実習がやりやすいからかもしれない。

Q2については、「授業中・授業外に友人に聞く」が、オンライン授業よりも対面授業の方が圧倒的に多かった。オンライン授業では、ほとんどの学生が自宅で実習を行っており、ブレイクアウトルームも1人だけに制限されていることから、友人とコミュニケーションを取ることができない状況にあった。一方、対面授業では、実習室で(座席は自由)隣同士に座り、お互いに分からないことを教えあう場面が見られた。スペルミスなどの軽微なエラーも、友人に見てもらおうことですぐに判明するからであろう。

Q3については、対面授業では3割の学生が質疑応答フォーラムを使わず、オンライン授業では全員が使っていた。オンライン授業では、1人で課題をこなしていかなければならず、分からないことを解決するためには授業中にブレイクアウトルームで質問するか授業外のときはフォーラムで質問するしかないことになる。また、対面授業で「しなかった」理由について聞いたところ、「友人からヒントをもらった」「授業内の個別指導の時間のみで解決したため」「家で実習ができないため、使うことがなかった」という回答になった。

表3 アンケート結果

Table 3 Questionnaire results.

Q&A	対面授業 (2019年度)	オンライン授業 (2020年度)
Q1. 実習はおもにどこでやっていますか?		
1) 授業中だけ	21%	8%
2) 授業中および時々授業外	76%	58%
3) 時々授業中おもには授業外	0%	34%
4) すべて授業外	3%	0%
Q2. 自学自習していて、わからない場合どうしていますか? (複数回答)		
1) 授業中先生に聞く	85%	75%
2) 授業中に友人に聞く	58%	0%
3) 授業外に友人に聞く	21%	17%
4) ヒントと実行結果を見る	67%	83%
5) 質疑応答フォーラムで質問する	55%	83%
6) インターネットで調べる	27%	75%
7) ほっておく	0%	0%
Q3. 質疑応答フォーラムで質問しましたか?		
1) した	70%	100%
2) しなかった	30%	0%
Q4. 実習の課題についてどうやっていますか? (複数回答)		
1) すべて独力でやっている	3%	25%
2) 時々友人か先生に教えてもらう	64%	50%
3) ほとんど友人か先生に教えてもらう	30%	25%
4) 時々、友人のプログラムをコピーして	3%	0%
5) ほとんど友人のプログラムをコピーして	0%	0%
6) すべて友人のプログラムをコピーして	0%	0%

Q4については、「すべて独力でやっている」が対面授業よりもオンライン授業の方が多くなった。また、「友人か先生に教えてもらう」が、いずれも対面授業の方がオンライン授業よりも多くなっている。(自己申告ではあるが)コピーについては、対面授業で1人だけときどきしていると回答した以外はまったくなかった。科目ガイダンス時には、個別指導やフォーラムを使うことで分からないことを明らかにできることを強調するとともに、ソースコードには必ず自分の名前を入力させるように指導している。

さらに、「授業について自由に書いてください」と質問したところ、相互コミュニケーションについて言及しているものを抜粋した結果、以下ようになった。

対面授業 (2019年度)

- 自分の力だけではできなくて、ときどき友達に聞くことがあった。
- 後半になるにつれて自分だけではできなくなり、友人や先生に聞くことが多くなった。
- 自分で解いた問題もあったが、友人や先生にヒントをもらう回数が多かった気がする。
- 先生や友達に教えてもらいながら頑張って進めた。
- 自分の力でやり、分からない箇所は友人に聞いたりした。
- 自分自身で解くこともするが、ケアレスや打ち間違え

が多いので友人に見てもらった。

これより、実習中の友人とのコミュニケーションについて言及している学生がいることが分かった。

オンライン授業（2020年度）

- 実習系の授業をオンラインで受けるのは難しく感じた。
- 講義がほとんどなく、実習だけで、分からない箇所は何をあてはめるのか検索しても出てこないところは先生に聞いたりして進めてきましたがだいぶ苦戦しました。ウェブアプリ論もプログラミング基礎も友達と受けていなかったのが分からない箇所はだいぶ先生に質問してしまってお迷惑おかけしてすみません。オンライン授業ではなく対面授業で受けた授業でした。
- オンラインだと自分でやる時間が増えるので、自分だけになると授業によって怠けてしまうことが起きてしまいがちなので、私は対面のほうがいいと思います。
- 対面の方が勉強しやすいと感じた。
- 人が密集している場所での授業ではないので心地良い環境ではあるが、なかなか質問できないのが少し不便だと感じた。

これより、このような実習科目ではオンライン授業よりも対面授業の方が良いという意見も散見された。

以上より、オンライン授業においてブレイクアウトルームを使うことで対面授業とほぼ同様に教員との質疑応答が実現できたといえる。ただし、学生1人だけのブレイクアウトルームとしたことによって、教員と1対1で質疑応答をできるようにしたが、常時できるわけではなく教員がルームに入ったときだけとなる。

さらに、学生にとっては、教員に聞くよりも友人同士で気軽に教え合うことも実習を進めるうえで有益といえる。しかし、オンライン授業では、学生同士の教え合いの機会を与えることができなかった。

以上のことから、対面授業では友人とのコミュニケーション（教え合い）がとれる一方でオンライン授業ではできないことが、2019年度よりも2020年度の方がクラス全体の課題の進捗状況や達成状況が低くなった要因として考えられる。

これを回避するための方策としては、ブレイクアウトルームを学生1人だけに限定しないやり方があげられる。現在のZoomの仕様では、参加者が自由にブレイクアウトルームを移動できる機能が追加されている[13]。そこで、個別指導の時間中に、友人となる学生同士がお互いのルームに入室してコミュニケーションをとれるようにする。あるいは、学生の中で進捗の早い者（毎年数名は存在）に共同ホストの権限を与えるやり方もある。これによって、ヘルプを求めた学生に対して、共同ホストの学生がそのルームに入室して教え合うようにする。これらについては、2021年度の授業で取り入れることを考えている。

5. おわりに

科目「プログラミング基礎」を、2019年度と2020年度において、対面授業とオンライン授業とで比較した。授業で取り上げた教材と課題は同じであったが、クラス全体の課題進捗状況と課題達成状況に差が生じた。2020年度のオンライン授業の方が、2019年度の対面授業よりもそれぞれの比率が若干低くなったことが明らかになった。この要因には、友人とのコミュニケーション（教え合い）の有無があげられる。また、質問のしやすさという点では、オンライン授業（ブレイクアウトルームやヘルプを活用）よりも、対面授業（挙手）の方が良いといえる。

2021年度については、文科省の指示により本学でも対面授業が主で、一部（履修者数の多い科目）だけオンデマンド授業になっている。「プログラミング基礎」は実習科目でもあるので、対面授業での実施となっている。

実習室では、換気、消毒、座席指定、PCごとにアクリル板設置といった感染対策を行っている。そこで、さらにディスタンスを徹底するために、教室での個別指導においてもZoomを継続して使うことにした。

今後については、2021年度も同じ形で授業を実施するので、オンライン授業でのZoom利用と対面授業でのZoom利用を比較した上で、より効果的な授業モデルの在り方についても検討したい。

参考文献

- [1] 岩崎日出夫：デジタル教科書とMoodleを用いた授業に関する検討—高等教育における初心者向けプログラミング教育の実践，日本デジタル教科書学会発表予稿集，Vol.6 (2017).
- [2] 五月女仁子：Moodleを利用したプログラミング教育の報告，神奈川大学経済学会商経論叢，Vol.47, No.19, pp.45-47 (2011).
- [3] 平塚紘一郎：プログラミング教育におけるMoodleの活用，仁愛女子短期大学研究紀要，Vol.43, pp.13-16 (2011).
- [4] Zoomミーティングとチャット，入手先 (<https://zoom.us/jp-jp/meetings.html>) (参照 2021-05-07).
- [5] Moodle ホームページ，入手先 (<https://moodle.org/>) (参照 2021-05-07).
- [6] 井上博樹，奥村晴彦，中田 平：Moodle入門，KAIBUNDO (2006).
- [7] 河村一樹：講義レスによるプログラミング実習教育の試み，情報処理学会研究報告，Vol.2015-CE-128, No.21 (2015).
- [8] 河村一樹：開講コマの違いによる学習進捗の相違について—自学自習ベースのプログラミング教育の場合，情報処理学会研究報告，Vol.2017-CE-139, No.8 (2017).
- [9] 河村一樹：JavaScriptによる情報教育入門，大学教育出版 (2011).
- [10] 河村一樹：Moodleを用いたプログラミング教育の事例，TIU学内報FD Newsletter “SEED” (2018).
- [11] 河村一樹：Moodleを用いたプログラミング教育における評価方法について—「正確性」に「迅速性」を加味した結果，e-Learning 教育研究，Vol.14, pp.34-42 (2020).
- [12] 河村一樹：Moodleを用いた自学自習ベースのプログラミ

ング教育, 東京国際大学人間科学・複合境域研究, No.6, ISSN 2423-9771 (2021).

- [13] Zoom ヘルプセンター: ブレイクアウトルームの管理, 入手先 (<https://support.zoom.us/hc/ja/articles/206476313>) (参照 2021-05-16).

推薦文

本論文は, 大学でのプログラミング基礎授業において, 新型コロナウイルスの対応でオンライン授業を行った際の, 従来の対面形式の授業とオンライン形式の授業を比較した報告である. 実践された授業について, 授業ごとの進捗や課題提出の状況, 受講生の学習中の行動などが対面授業とオンライン授業で対比されており, その中で「先生への質問は大幅に減ったわけではないが, 友人同士の質問は大幅に減っている」などの違いが明らかになった. 受講生の感想も紹介されており, 今後もオンライン授業が継続すると思われるプログラミングの基礎教育において, 一定の貢献が期待できることから, 速報として推薦する.

(論文誌「教育とコンピュータ」編集副委員長 兼宗 進)



河村 一樹 (正会員)

1955 年生まれ. 立教大学理学部卒業, 日本大学大学院理工学研究科博士前期課程修了, 博士 (工学). 東京国際大学商学部教授. 情報教育工学に関する調査研究に従事. 日本教育工学会, e-Learning 教育学会各会員.