

# 講義動画を利用した Web プログラミング学習時のつまずき要因の分析

高橋圭一<sup>1</sup>

**概要:** 世界的な COVID-19 の感染拡大の影響下において、大学において学びを継続するためオンライン環境での講義が様々な形式で実施されている。筆者が所属する学科においても様々な試みが行われてきた。本稿では、筆者が所属する学科の3年生を対象とした Web プログラミングを学習する科目を対象として、オンライン環境での学習の問題点を明らかにする。具体的には、本科目では演習時間中にあらかじめ用意された講義動画を視聴しながら受講者は学習を進める。受講者のデスクトップ画面を録画して分析することで、受講者がどこでどのようにつまずいたのかを明らかにし、授業資料や演習環境など改善の可能性を検討する。

**キーワード:** ビデオ学習, タイプミス, ファカルティ・デベロップメント

## Analysis of Students' Mistakes in Learning Web Programming Using Lecture Videos

KEIICHI TAKAHASHI<sup>†1</sup>

**Abstract:** Due to the global spread of COVID-19, various types of online lectures have been conducted in order to continue learning at universities. In our department, various attempts have been made to provide online classes. In this paper, we clarify the problems of online learning for a course on Web programming for third-year students in our department. Specifically, in this course, students learn by watching the lecture videos prepared in advance during the exercise time. By recording and analyzing the students' desktop screens, we can identify where and how the students made mistakes, and examine the possibility of improving the class materials and the exercise environment.

**Keywords:** Video-Based Learning, Typing Error, Faculty Development

### 1. はじめに

2020年に新型コロナウイルス感染症が拡大し、世界中の教育機関で安全を確保しつつ学びを継続するための施策が進められてきた[1]。文科省が国内の大学等を対象に行った調査によると、2021年度後期にオンラインで実施する授業があると約83.2%が回答した[2]。また、Takácsらが2020年にハンガリーの学生を対象に行ったオンライン授業に関する意識調査によると、教員や他の学生とのコミュニケーションが難しいなどの課題がある一方、自宅から移動することなく授業を受けられることに利点を感じるとほとんどの学生が回答した[3]。こうした利点を考慮すると感染終息後もオンライン授業が引き続き実施されることが予想される。

筆者が所属する大学においても、2020年初めからオンライン授業の実現方式の調査や実施方法を検討しつつ、学生の受講環境の整備や受講方法の教育などを進めてきた。最終的には、講義資料の配布や課題の提出管理には Google Classroom を採用し、講義映像の配信には Zoom ビデオコミュニケーションズ社が提供する Web 会議サービスである Zoom の採用を決定し、現在でも対面授業と併用しつつそ

れらのツールを活用している。

筆者の担当科目の1つに、3年次を対象とした Ruby による応用プログラミングを学習する科目がある。2019年度以前は講義や演習を対面で実施してきたが、2020年度および2021年度は前述の Google Classroom と Zoom を利用したオンライン授業で実施した。オンライン授業は、オンデマンド型、ライブ配信型、双方向型などがあるが、Zoom は映像配信に加えて受講者が反応する手段が用意されているため双方向型に分類される。演習中につまずいたときに Zoom の「手を挙げる」や「チャット」などの双方向型の機能を利用することで、対面授業と同様に授業内で問題解決が可能と期待した。しかし、実際にオンライン授業を進めてみると、そうした機能を利用する受講者は限られていることがわかった。対面授業であれば、演習中に巡視することでつまずきに対処できるが、オンライン授業では受講者の取り組み状況を把握できないため、受講者から質問がなければつまずきに対応することが難しい状況であった。

本稿では、オンライン授業のうち、特にプログラミング授業において、オンライン授業中に受講者はどのように課題に取り組み、どこでどのようにつまずいたのかを調査す

<sup>1</sup> 近畿大学産業理工学部情報学科  
Kindai University

ることを目的とする。

以下に本稿のリサーチクエスチョンを示す。

RQ 1 受講者はオンライン授業をどのように受講しているのか

RQ 2 受講者はオンライン授業のどこでどのようにつまづいたのか

RQ 3 受講者はつまづきをどのように解決したのか

## 2. 調査方法

### 2.1 対象科目

1 章でも述べたように、筆者が担当する 3 年次対象の Ruby による応用プログラミングを学習する科目の第 9 回と第 10 回を調査対象とする。本科目は講義と演習が連続した 2 コマ (180 分) の授業である。本科目の授業計画を表 1 に示す。

受講者は 1 年次より Ruby の基礎を学習しており、さらに本科目の第 1 回から第 8 回までに様々な応用プログラミングを学ぶことで Ruby プログラミングの練習を十分に積んだ状態である。

調査対象である第 9 回と第 10 回は Ruby on Rails (以降、Rails と略す) プログラミングの基礎を学ぶことを目的としている。Rails は Ruby を用いた Web アプリケーション開発フレームワークの 1 つである。現代的な Web アプリケーション開発ではフレームワークを利用することを前提としており、他言語でも類似するフレームワークが多数あることから、本調査で得られた知見が少なからず活かせると考えてこの講義回を選定した。

ブックマークを管理する素朴な Web アプリケーションを作りながら、Rails の基本的な機能や操作を体験的に学ぶ。操作手順はすべて講義資料に記載されているため、受講者は講義中に講師による説明を聞きながらプログラムを入力すれば課題を完成できる。

表 1 調査対象科目の授業計画

講義回	授業内容
1	導入講義、Rubyの基礎 (復習)
2	クラスモジュール
3	ファイルによる永続化
4	データベース基礎 (復習)
5	データベースによる永続化
6	Webアプリケーション基礎
7	蔵書管理アプリのWeb化
8	Webアプリケーション開発
9	Ruby on Rails : ルーティングとMVC
10	Ruby on Rails : scaffoldを使わないアプリ開発
11	Ruby on Rails : 画像共有アプリ
12	Ruby on Rails : バリデーションとテスト
13	Ruby on Rails : 総合演習
14	Ruby on Rails : 総合演習
15	定期試験

### 2.2 講義資料の操作手順

第 9 回および第 10 回の講義資料の例を図 1 に示す。本資料は 2018 年度に、1 コマ (90 分) で対面で講義する前提で作成した資料である。この講義資料から抜き出した操作手順を表 3 及び表 4 に示す。

第 9 回の操作手順は 16 ステップであり、第 10 回は 21 ステップある。Rails には生産性を高めるためさまざまなコマンドが用意されている。表中では、操作コマンドに下線を付記している。また、Rails は MVC (Model-View-Controller) アーキテクチャを採用しているため、Web アプリケーションを構築するためには、MVC に対応した、それぞれのファイルにプログラムを記述する必要がある。表中には、受講者が各ファイルに記述するプログラム量の目安として文字数と行数を示している。第 9 回ではコマンド操作が 7 回で、修正するファイルは 5 つある。第 10 回ではコマンド操作はなく、修正するファイルは 6 種類ある。



図 1 講義資料の例

### 2.3 演習環境

オンライン授業であるため、受講者のパソコン上に Rails プログラミングの開発環境を構築する必要がある。OS によらず動作し、環境構築の手間を省くため、Amazon 社が提供する AWS Cloud9 を使用する。AWS Cloud9 は Web ブラウザ上で動作するため、パソコンにソフトウェアを導入したり設定する必要はない。図 2 に AWS Cloud9 の操作画面の例を示す。

Rails アプリケーションは、階層構造で配置されたさまざまなファイルからなるが、①で必要なファイルを探して選択することで、②のテキストエディタで編集できる。また、③のターミナルウィンドウで Linux コマンドや Rails の各種コマンドを実行できる。

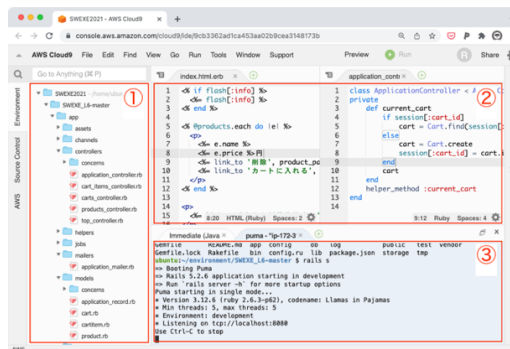


図 2 AWS Cloud9 の操作画面例

## 2.4 被験者および被験者による実験手順

2021年度の本科目の受講者から男女2名ずつ4名を被験者とする。被験者は講義開始時に他の受講者と同様にZoomに接続し、各自のパソコンのデスクトップ全体をZoomの画面共有機能で共有してから画面録画を開始する。その後、Google Classroomにアクセスして講義動画を視聴し、必要に応じて講義資料を参照する。

受講者は、講義動画を視聴しながら操作を進め、操作につまづいたらZoomのチャット機能で講師にヘルプを求められることができる。講義動画の視聴と課題の作成を終えた時点で録画を停止し、Zoomアプリケーションが生成した動画ファイルを講師に提出する。

## 3. 実験結果

### 3.1 被験者ごとの作業時間

4名の被験者から受け取った動画ファイルに記録された作業時間を表2に示す。講義動画の長さはそれぞれ、第9回は99分、第10回は77分であった。表中の比率は、講義動画の長さに対する各受講者の作業時間の割合を示している。

受講者は、講義動画を視聴しながら、途中で動画を止めたり巻き戻したりしながら操作する必要がある。したがって、比率が大きいほど、プログラムの入力やデバッグに時間がかかったことを表している。なお、作業時間の下線は授業時間中に講義動画の視聴が完了しなかったことを表している。理由については後述する。

作業時間は被験者によって個人差があるものの、平均値を見ると、第9回は164分で第10回は106分であることから60分減少している。第9回はRails開発の初回であったため作業時間を必要としたが、第10回では比率が1に近くなっていることから、講義動画を見ながら順調に作業できたようである。

表2 被験者ごとの作業時間と講義動画に対する比率

被験者	第9回		第10回	
	作業時間(分)	比率	作業時間(分)	比率
1	<u>176</u>	1.77	<u>170</u>	2.20
2	<u>181</u>	1.83	77	0.99
3	151	1.52	81	1.04
4	148	1.50	96	1.24
平均	164	1.65	106	1.37

### 3.2 被験者ごとのつまづき状況

4名の被験者から受け取った動画ファイルを再生して作業の様子を目視で分析した結果を表3及び表4に示す。表中では、4名の被験者のつまづきの種別とそのつまづきを取り除くまでの修正時間(単位は分)を表している。

つまづきを3つの種別に分類する。Tはタイプミス、Sは保存忘れ、Pは場所間違いである。「保存忘れ」は、講義動画を視聴して同じようにテキストエディタに入力したもの

の保存せずに次の操作をしたため、講義動画とは異なる結果が表示されるつまづきである。「場所間違い」は、コマンドを実行するディレクトリが操作手順とは異なるため、講義動画とは異なる結果になるつまづきである。

3.1節で述べたとおり、第9回には被験者1と被験者2が、第10回には被験者1が講義動画の視聴を完了できなかった。表中の灰色部はそのことを表している。特に第10回の被験者1は表4の3番目の操作手順でつまづいて授業終了まで自己解決できなかったため、修正時間を算出できなかったことを※で表している。

#### 3.2.1 つまづき種別

表3及び表4をもとに、種別ごとのつまづき回数と修正時間の合計を図3及び図4に示す。

つまづき回数では、約8割のつまづきがタイプミスであることが分かる。さらに修正時間を見ると、第10回のタイプミスは全体の9割以上を占めていることが分かる。

第9回では保存忘れの発生回数が2回である。それらの修正時間は11分を要しているが、第10回では1分で修正できている。いずれも発生回数はわずかであるが、保存を忘れたファイルの種類によって修正の困難さに差があることが示唆された。

第9回では、場所間違いの発生回数が多かった。Rails開発を初めて学習した回であったため、ターミナル操作に不慣れであることが原因と考えられる。

表 3 第 9 回の操作内容と被験者によるつまずき状況

操作 手順	操作コマンド/操作ファイル名	文字数	行数	被験者 1		被験者 2		被験者 3		被験者 4	
				種別	時間	種別	時間	種別	時間	種別	時間
1	<u>rails generate controller bookmarks index</u>	41	1			P	0.5				
2	routes.rb	21	1								
3	bookmarks_controller.rb	13	2								
4	routes.rb	67	2	T	4.0						
5	<u>rails generate model bookmark title:string url:string</u>	53	1			T	1.9				
6	<u>rails db:migrate</u>	16	1					T	4.8		
7	application.rb	106	3							T	9.7
8	Gemfile	16	1	S	10.4			S	1.1		
9	<u>bundle install</u>	14	1			P	0.4				
10	<u>rails console</u>	13	1	P	1.7						
11	<u>book = Bookmark.new( title:'a', url:'http://a.com')</u>	54	1					T	2.2	T	0.6
12	<u>book.save</u>	9	1								
13	bookmarks_controller.rb	39	3								
14	index.html.erb	76	4								
15	index.html.erb	98	3								
16	index.html.erb	84	3					T	0.8	T	2.6
合計		720	29								
						16	2.9	8.8			13

注1 種別は、つまずきの種別を表す。それぞれ、Tはタイプミス、Sは保存忘れ、Pは場所間違いである。

注2 時間は、つまずきから復帰するまでのデバッグ時間を表す。単位は分である。

表 4 第 10 回の操作内容と被験者によるつまずき状況

操作 手順	操作コマンド/操作ファイル名	文字数	行数	被験者 1		被験者 2		被験者 3		被験者 4	
				種別	時間	種別	時間	種別	時間	種別	時間
1	index.html.erb	47	1			T	5.8				
2	routes.rb	40	1								
3	bookmarks_controller.rb	37	3	P	※						
4	new.html.erb	200	7			T	1.8		T	33.6	
5	routes.rb	40	1								
6	bookmarks_controller.rb	48	3						T	0.7	
7	bookmarks_controller.rb	133	5								
8	index.html.erb	196	7			T	0.8				
9	routes.rb	47	1			T	6.0		T	6.3	
10	bookmarks_controller.rb	41	3			T	6		T	4.1	
11	bookmarks_controller.rb	86	4								
12	index.html.erb	44	1								
13	routes.rb	41	1								
14	bookmarks_controller.rb	52	1								
15	show.html.erb	136	9			T	3.3	T	0.4		
16	index.html.erb	49	1								
17	routes.rb	46	1						T	8.6	
18	bookmarks_controller.rb	52	3								
19	edit.html.erb	199	7					S	1.5		
20	routes.rb	45	1								
21	bookmarks_controller.rb	149	5					T	4.7		
合計		1364	53		※						
						16	6.6	20			

注1 種別は、つまずきの種別を表す。それぞれ、Tはタイプミス、Sは保存忘れ、Pは場所間違いである。

注2 時間は、つまずきから復帰するまでのデバッグ時間を表す。単位は分である。

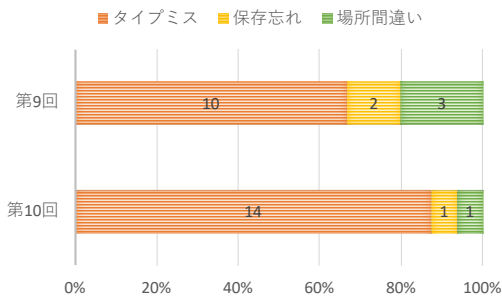


図 3 種別ごとつまずき回数合計

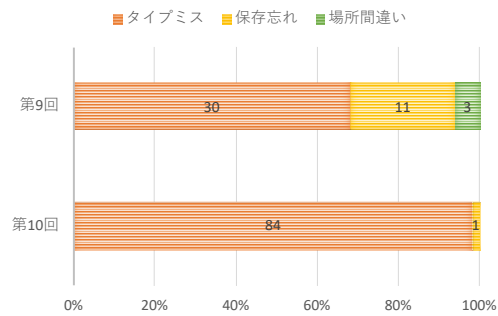


図 4 種別ごと修正時間(分)合計

### 3.2.2 タイプミスのつまずきの具体例

本節では、発生回数の多かったタイプミスについて、被験者のつまずきの具体例を述べる。今回の実験のつまずきを観察すると、タイプミスは次の3種類に分けられる。

**Rails の規約に起因する誤り**：例えば、routes.rb というファイルに以下を追記したときに、bookmarks を bookmark と単数形とした誤りである(表 3 の被験者 1 の 4 番目)。

```
get 'bookmarks/index', to: 'bookmarks#index'
```

Rails ではコントローラのクラス名を基本的に複数形とする規約がある。そもそもこれは識別子の誤りなので、1文字でも誤るとエラーになるが、Rails のプログラムではこの例のような名詞の複数形が頻出する。慣れないうちは名詞を正確に入力することに気を取られて s を入力忘れしやすくなる可能性が考えられる。

**音声とつづりの不一致による誤り**：以下のプログラムを routes.rb に追記したときに、被験者 2 と被験者 4 がともに destroy とタイプミスした(表 4 の 9 番目)。

```
delete 'bookmarks/:id', to: 'bookmarks#destroy'
```

本人は正しく入力したと思っているため、つづり間違いに気づきにくく修正に時間を要したようである。

**Ruby 文法の理解不足による誤り**：Rails ではメソッドの引数として Hash を多用する傾向にある。Ruby で Hash を記述する方法はいくつかあるが、最も簡潔な記法としてキー名の後ろにコロンをつける表記を用いる場合が多い。受講者自身が Hash であると認識してプログラムを入力していればコロン忘れは発生しにくいはずであるが、title や url の後ろに付記すべきコロンを忘れる誤りがあった(表 3 の被

験者 3 と被験者 4 の 11 番目)。

```
book = Bookmark.new( title: 'a', url: 'http://a.com' )
```

### 3.2.3 受講者自身で修正が困難なつまずき

特殊な例ではあるが、他の受講者も発生した可能性があるため、被験者 1 と被験者 2 が講義動画を視聴完了できなかった 3 つの状況について述べる。

**被験者 1 の第 9 回**：つまずいた回数や修正時間は他の被験者と大きな差はなかった。特に大きなトラブルも見られなかったことから単にプログラムの入力が遅かったためと考えられる。

**被験者 2 の第 9 回**：テスト用の Web サーバを起動するコマンドを実行した直後に誤ってターミナルウィンドウを閉じる操作をした。ウィンドウが閉じて Web サーバが起動した状態となりポートを専有していたため、Web サーバを起動するコマンドが実行できず、授業終了まで原因を調査し続けたため操作手順が進まなかった。

**被験者 1 の第 10 回**：コントローラである bookmarks\_controller.rb を編集しようとして、受講者が勘違いしてテスト用の bookmarks\_controller\_test.rb を編集したため、プログラムをいくら追記しても Web ページに結果が表示されなかったものの、そのまま操作手順のとおり作業を進めたため、結果的に全く機能が動作しなかった。

## 4. 考察

### 4.1 RQ1 受講者はオンライン授業をどのように受講しているのか

講義動画は動画ファイル (mp4 形式) で提供しているため、再生速度を上げて視聴することも可能なはずだが、被験者の録画ファイルを見る限り、被験者は全員、講義動画を等倍速で視聴していた。講義動画は単一ファイルであり、解説の中のどこで操作説明になるかわからないためスキップすることが難しかったためと考える。

一方、講義動画が単一ファイルであるため、講義動画中の操作デモンストレーションの区切りで受講者が一時停止し、プログラムを入力する様子がたびたび見られた。講義動画と同じ結果にならない場合は、動画を巻き戻して入力したプログラムを確認の様子も見られた。動画ファイルの再生時間はおよそ 90 分くらいあるため、短時間の巻き戻し操作は非常に難しい様子であった。こうした操作に気が取られてしまい、タイプミスしやすい状況になっているように見えた。

2020 年に本科目をオンライン授業として初めて実施したときには、Zoom でリアルタイム配信した。しかし、操作トラブル等で講義に追従できなかった学生から講義動画を見たいという要望があったため、あらかじめ講義動画を講師が作成し、受講者は講義時間中にその動画を視聴し、講

師はつまずいた受講者の対応をするようになった。

対策としては、講義動画の視聴が終わってからプログラムを入力できるように、操作の区切りごとに個別の動画ファイルを用意し、それぞれの作業に集中できるようにする方法が考えられる。さらに、プログラムが正しく入力できたか確認する方法も講義資料に明示することで着実に演習を進められるだろう。

#### 4.2 RQ2 受講者はオンライン授業のどこでどのようにつまずいたのか

表 3 及び表 4 を見ると、つまずきの発生箇所には顕著な偏りはなく、操作手順の至るところでつまずきが発生している。つまずきは、操作手順よりはむしろ被験者の課題の取り組み方に依存しているようである。例えば、被験者 2 と被験者 4 はタイプミスが 6 回以上発生しているが、被験者 3 は 2 回であった (表 4)。被験者 2 は短時間にプログラムを入力し、誤りがあれば素早く修正することを好み、被験者 4 は着実にプログラムを入力することを好む様子が見られた。結果的につまずきを解決できるのであれば、つまずき回数的大小は問題ではないと考えている。

3 つのつまずき種別のうちタイプミスが最も多かったが、その要因は 3.2.2 節で述べたとおりである。偶発的に誤入力する場合もあれば、development や rails db:migrate のように先入観や曖昧な知識によるタイプミスもある。このような偶発的な誤りはもちろん、個人の前知識から生じる系統的なタイプミスは予見が困難である。

一方、3.2.2 節でも述べたとおり、コロンの付記を忘れてつまずく誤りがあったが、コロン以外でも「`<%= link_to book.title, book.url %>`」のようなプログラムを入力するときにピリオドやカンマなどのタイプミスや記入忘れがあった。この誤りの発生の要因の 1 つとして、講義動画の見にくさに関係しているようである。

講師は自身のパソコンでスライドを全画面化して説明し、その画面を録画して講義動画を作成するが、受講者はその動画を見ながらプログラムを入力する必要があるため、講義動画のウィンドウと AWS Cloud9 のウィンドウをデスクトップ上に左右に並べて視聴していた。したがって、講義動画は画面の半分以下に縮小して表示されるため、特にピリオドとカンマは区別しにくいような状況が見られた。この環境がタイプミスを誘発していた可能性は考えられる。この点については、受講者のパソコンの画面で縮小表示されることを想定して動画を作成する、あるいは、タイプミスしやすい記号などは、講義資料で注意を促すなどで回避できるかも知れない。

#### 4.3 RQ3 受講者はつまずきをどのように解決したのか

表 3 及び表 4 に挙げたつまずきのほとんどは被験者が自己解決していた。自己解決するためには、被験者は、講義動画で示された手順を漏れなく操作しているかを確認する、あるいは、表示されたエラーメッセージをインターネ

ットで検索して対処方法を探す様子が見られた。

表 3 と表 4 を見ると、プログラムコードを入力するコントローラ (bookmarks\_controller.rb) やビュー (\*.erb) の修正時間はほぼ 5 分以内であるため比較的短時間に修正できる。これは操作手順で直前に入力したファイルに誤りがある場合が多いためと考えられる。一方、第 9 回の application.rb や Gemfile、第 10 回の routes.rb のような設定ファイルにおける誤りは、誤り位置のわずかな差によってエラーの症状が大きく異なるため修正は困難である。自己解決できない場合には正解ファイルを早めに提示するなどの対応が必要と考えられる。

今回の実験で最も修正に時間を要したのは、第 10 回の被験者 4 の 4 番目の操作であった。4 番目の手順では、200 文字 7 行のコードを new.html.erb というビューのファイルにプログラムを入力する必要があるが、その先頭行を以下に示す。

```
<%= form_for @bookmark , url: {action: :create} do |f| %>
```

被験者 4 はこのプログラムの「=」を忘れるタイプ忘れの誤りをした。「=」を入れることで HTML を出力できるのだが、「=」を忘れると出力されないため、Web ページを開いても空のページが表示されない。この誤りは本科目ではよく見られる誤りであることから、講義資料に注意を促す記述を記載することでつまずきを低減できるかも知れない。

このように、つまずきを発見して講義資料などを充実することでつまずきを抑制できる可能性がある。一方、3.2.3 節で述べたように、経験が少ない状況では自己解決が困難なつまずきがある。例えば、講義開始前に「同じ箇所のつまずきが 10 分以上解決できない場合は必ず相談すること」などのように、質問する条件についてルール化することで、過度のつまずきの低減が期待できる。

## 5. まとめ

本稿では、オンライン授業中の受講者の取り組み状況を調査するため、筆者が所属する学科の 3 年生を対象として Web プログラミングの科目において、講義動画を視聴して課題に取り組む様子を録画した動画ファイルを分析した。

分析の結果、つまずき要因の約 8 割はタイプミスによる誤りであった。タイプミスの発生要因には、偶発的に発生する誤りや個人の前知識から生じる誤りであった。対策としては、講義資料のうち、つまずきが頻出する箇所にタイプミスの事例を掲載するなど、注意を促す方法が考えられる。また、経験が少ない状況では自己解決が非常に困難なつまずきがあった。この対策としては、講義開始前に「同じ箇所のつまずきが 10 分以上解決できない場合は必ず相談すること」などのように、質問する条件についてルール化することで、過度のつまずきの低減が期待できる。

今後は、本稿の調査で得られた知見をもとに講義資料や講義動画を改善していく。

## 参考文献

- [1] Vincent-Lancrin, S., C. Cobo Romani and F. Reimers (eds.) (2022), How Learning Continued during the COVID-19 Pandemic: Global Lessons from Initiatives to Support Learners and Teachers, OECD Publishing, Paris, <https://doi.org/10.1787/bbeca162-en>.
- [2] “令和3年度後期の大学等における授業の実施方針等に関する調査の結果について”, [https://www.mext.go.jp/content/20211118-mxt\\_kouhou01-000004520\\_1.pdf](https://www.mext.go.jp/content/20211118-mxt_kouhou01-000004520_1.pdf), (参照 2022-2-8)
- [3] J. M. Takács and M. Pogatsnik, "The online learning from the students' perspective," 2021 IEEE 19th World Symposium on Applied Machine Intelligence and Informatics (SAMI), 2021, pp. 000027-000032, doi: 10.1109/SAMI50585.2021.9378665.