

# e ラーニング研究

第 11 号

サイバー大学



# 目 次

Cloud Campus の 10 年 —2012～2022— .....	川原 洋 .....	1
メタバースは多様性と包摂性に資する学びの場となりうるか —サイバネティックス再考— .....	高林 友美 .....	9
プログラミング教育における「写経型演習先行・文法学習後行方式」の 実践報告 .....	松本 幸子 .....	23
WBT 教材の効率的な開発法に関する研究 .....	鈴木 秀男，長島 雄平，田中 健太郎 .....	39



# Cloud Campus の 10 年 ―2012～2022―

川原 洋<sup>1</sup>

## 背 景

2007 年 4 月に開学したサイバー大学（以下、本学）のオンライン教育システムは、全ての授業をオンライン・オンデマンドで実施するという日本初の試みでもあり、機能的にも運用面においても多くの課題を抱えて運用に入った。これらの課題を逐次解決し、オンライン上の運営でありつつ正規教育機関としての責務を果たすべく、自ら開発を進めてきたのが、開学 5 年後の 2012 年 4 月より運用を開始した初代 Cloud Campus（以下 CC）である。

開学直後の授業はすべてオンラインで開講したが、受講者の端末は Windows PC と指定のブラウザに限定されていた。従って、入学者は少なくともパソコンや Web ブラウザの基本的な操作ができていることが前提であり、そのほかにも学内で必要なツールやリソースも Windows 互換の製品であったため、それらも特別な支援なく利用できることが少なからず期待された。

本学が開学した 2007 年は Apple 社が iPhone を世に出した年でもある。いわゆるスマートフォンの登場である。これらの高機能、高画質の情報端末の登場は、タブレット端末も含め、開学後、数年を経た本学の学習環境を大幅に改善することとなった。すなわちオンライン教育のシーズが、オンライン学習環境のユーザインターフェイスの分野で著しい進歩を遂げた時期ともいえる。

本稿では正規教育機関における教育指導要件を満足するために、本学が 2012 年から 2022 年にかけて開発・運用してきた Cloud Campus を通じて、オンライン教育への取り組みを振り返ってみたい。

## 高度マルチメディアによる双方向教育

本学の設置認可要件には、「高度マルチメディア」の活用による「双方向」教育を実施することが掲げられている。すなわち、前者については複数種のデジタル媒体を目的別、あるいは複合的に利用することによる効果的な教育を、インターネットを通信手段として提

---

<sup>1</sup> サイバー大学 学長

供することを目指している。また後者については、講師からの講義による一方的な知識の伝達だけでなく、学生の学習効果の測定のためのテストやレポート課題のみならず、講師と学生間の質疑応答、学生間のコミュニケーションが高度なメディア活用によって行われることも期待されている。

講義科目においては、講師がスライドを用いて解説を行なっている映像と該当するスライドを大小2画面に同時表示して提供されるのが基本となっている。この授業コンテンツを10分以上・20分未満の再生時間の枠組みで「1章」と称し、1回の授業を4章構成とし、さらに8～10問程度の小テストやディベートなどの課題学習が最後に行われて、全体として90分以上の授業時間となっている。講義コンテンツをすべて視聴（再生時間と授業スライドの表示が確認される）しないと、小テストへ進むことができない流れになっている。講義コンテンツはスライドのタイトルによる目次で表示されているので、復習のためにいつでも指定のビデオをスライドと共に再生することができる。

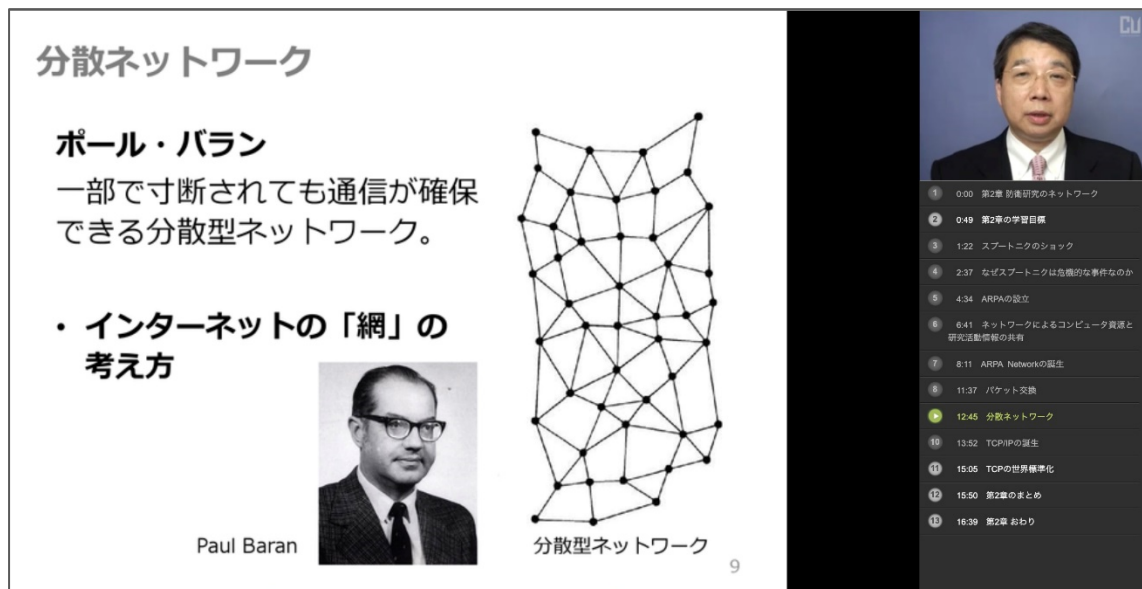


図1 Cloud Campus ビデオとスライドによる講義コンテンツ

このビデオ＋スライド [V＋S] 構成のコンテンツを制作するツールは、ビデオカメラとマイク付きのパソコンさえあれば、誰でも簡単に自作（自撮り）できるように学内で開発した。すなわち、授業コンテンツを制作する教員のみならず、課題やゼミナール等のオンラインでの発表のために、すべての学生が自らのパソコンからコンテンツを制作できるようにした。これによって、メディアの質的な格差なく、講師と学生、あるいは学生間で文字通り、高度なメディアを利用した双方向学習の環境を整えた。

また、制作手順も極力操作性の向上に努めた。PDFに変換されたスライドをCCの「コンテンツ制作サービス」（以下CPS）にアップロードすると順番に画面に表示される。ユー

ずは録画ボタンを押して、画面に現れたスライドを順番に表示しながら、ウェブカメラに向かって自らの録画を始める。終了したらストップボタンを押す。サーバですべてのスライドとビデオの同期処理が行われると、速やかに再生して内容を確認することができる。収録後にスライドとビデオの同期の調整をすることはできるが、実はビデオそのものの編集機能は装備していない。ビデオの長さから（学生制作の場合、通常 5～10 分）、編集に時間をかけるより、再収録した方が時間的に効率が高く、また制作を繰り返すことによって内容の完成度も上がっていくからである。

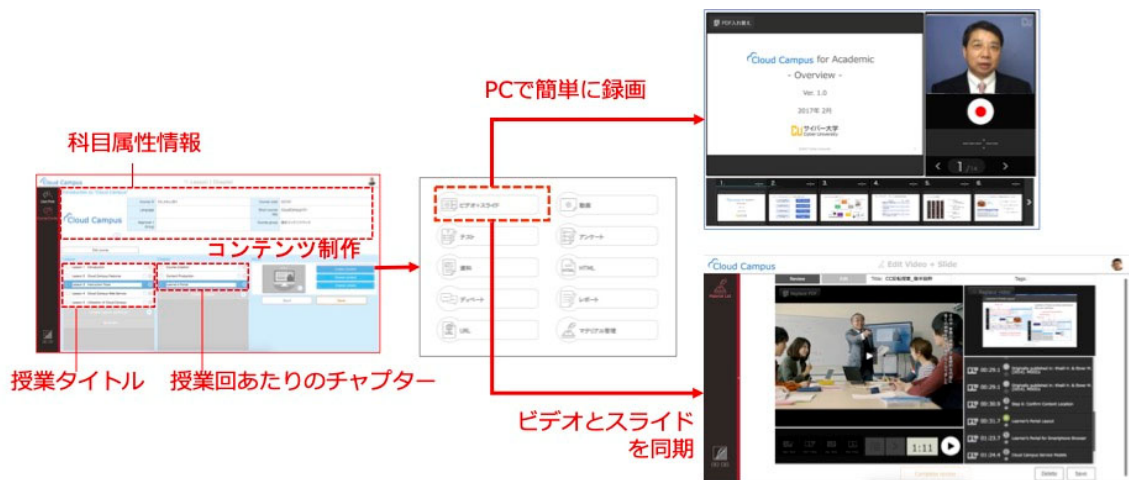


図 2 CPS が提供する [V+S] コンテンツの制作画面

## スマートフォンアプリが学習機会を増大

冒頭で述べたように、いわゆるスマートフォンの登場が、本学のモバイル学習環境を著しく向上させた。しかし、2010 年に国内ではソフトバンクでのみ販売が始まった iPhone 3GS に引き続き、2012 年には iPhone 4 が発売されていたが、当時の iPhone 4 の画面サイズは 3.5 インチ（解像度 640×960）であった。学内での iPhone 向け学習アプリのプロトタイプは完成していたが、それ以前に携帯電話ですでに完成していたモバイル学習アプリ（ケータイキャンパス）を世に出したときも、文部科学省から「画面サイズが小さすぎる」ため、認可対象としていたパソコン画面に準ぜずという理由で、補足的な学習ツールとしてのみ、その利用が認められていた。iPhone の画面サイズは、いわゆるガラケーと比較して格段に向上したとはいえ、一般的なパソコン画面との比較においてはまだ小さい。

このころタブレットと呼ばれる全く新しい種類のモバイル端末が市場に登場した。Apple が iPad を 2010 年に発表し、国内でも同年 5 月には販売が始まった。画面サイズにおいては小型のパソコンに準ずる大きさであったが、残念ながら本学において正規授業端

末として採用するには、後述にある顔認証による本人確認に必要な Web カメラが前面にも装備されている必要があったため、次の iPad2 の出現を待つことになった。



図3 iPad2 の CC アプリ 顔認証による本人確認から小テスト受験

iPad2 向けの iOS アプリのリリースによって、本学初のモバイル学習環境を整備した。モバイルアプリのもう一つの特徴は、授業コンテンツを事前にダウンロードできることである。2012 年当時、パケット通信費用はまだ高価であり、Wi-Fi の利用が推奨されていたが、現在のようにパブリック Wi-Fi がどこでも使える状況でもなく、3G 通信によるビデオの動画配信コストの学生への負担が懸念された。そこで学生が履修登録した科目の授業コンテンツは、Wi-Fi に接続できる環境で、事前に章単位で端末にダウンロードしておき、ネットワークへの接続の有無にかかわらず、いつでもどこでも受講できるようにした。受講後の学習履歴はネットワークに接続されたタイミングでサーバと同期するようにして、学生ごとの学習履歴の一元管理が行えるようにした。

小テストの受験も、本人確認のためにサーバに安定的に接続しておかなければならないが、Web カメラによる本人確認が完了すると試験画面が開くようにした。

その後の iPhone やスマートフォン、そしてタブレット端末の製品機能や画面解像度も著しく向上した。そしてモバイル通信サービスも 3G から 4G へ、そしてさらに 5G と進化してきた。それに伴う通信コストと通信帯域の向上は、パソコンでの学習環境を凌駕するだけでなく、日常のいわゆる隙間時間での学習時間の確保や自宅においても速やかに画面を開くことができるため、学習にとりかかる心理的ハードルを下げる効果を生んだ。

このネットワークの著しい向上は、ネットワークへの常時接続を日常化し、かつ Web 開発手法としてのレスポンスデザインは、パソコンもスマホも Web アプリケーション開



発を一元化することが可能となり、開発効率の向上に貢献し、第一世代で活躍した非同期型のモバイルアプリの利用率を下げる結果となっている。

## 本人確認方法のメディア化

オンライン授業の利便性や可用性はモバイル端末の進化とネットワークの高速化に助けられて学生の学習時間の確保に貢献した。一方で単位認定が必須となる正規教育課程において、都度の学生の本人確認は極めて重要である。

そこで CC の運用時に導入されたのが、サーバベースの顔認証システムである。サーバベースと明記した理由は、これがいまでは汎用的に使われているスマートフォンのログイン時に端末側で行われる顔認証と区別するためである。本人確認のために使われるマスターフォトは、入学後の経年と共に変化する可能性が高い。学生は最新のマスターフォトを随時更新することができるが、すべての撮影履歴は更新後にもそれが本人のものであることを確認するためにサーバで一元的に保管されている。

更新されたマスターフォトは、入学時の写真付き身分証明書や、更新されてきたそれまでのマスターフォトと照合され、CC 上での成績評価や本人確認が重要と思われるテストのオンライン受験や課題提出などの場面において、最新のマスターフォトとして照合されている。

## 定期試験における試験監督

オンライン教育における学習成果の評価方法として、課題の提出や試験が実施されるが、正規教育において本人確認は運用上の最重要事項である。ことにオンライン試験における不正行為対策は、臨場での試験監督にできるだけ近い形で臨まなければならない。本学のオンライン試験は、すべてオーブンブック（テキストや参考文献の持ち込み可能）であるが、本人が自身の力で他者との干渉を一切絶って実施しなければならない。

そのため、試験は隔離した空間を用意し、他者との会話や干渉を一切絶って行うことをルールとしている。試験中は本人確認のために使われた Web カメラが、そのまま監視カメラとして一定の間隔でスナップショットを撮り始める。試験中は第3者がカメラの画角内に映り込んだり、言葉を交わしたりするようなシーンがあれば、これらを記録する。システムから発せられるアラートに対し、改善が見られない場合は、試験中でも強制的に受験を停止することもある。

一方で強制的に受験が停止されるまでに至らずとも、疑義が発生した場合は、他者が試験中に入室したり、話しかけてきたりした場合の状況について詳しくヒヤリングを行い、受験中のログデータの分析結果も含めて、教授会で審議し判定を行っている。

## IT リテラシーとオンライン学習

私たちは一様にスマートフォンなどを携行して、様々な目的で Web へのアクセスを日常的に容易に行なえるようになった。一方で、スマホと比較して格段に大きいパソコンの画面とそのキーボードを駆使して行う情報生成や加工・分析を基本的な行為として行う機会を失いつつある。機会が少なくなれば、そのスキルも低下していく。皮肉なことに情報端末を使いこなせばこなすほど、創造的作業を行うための IT リテラシーが下がってきているのである。IT リテラシーの向上を含めて、IT スキルの習得を目指している学生にとって、オンライン学習に必要な基本的な IT スキルの欠如が問題となり得る。

従って、一般的に入手可能な IT ツールの操作がおぼつかないために初期段階で挫折してしまうことも大いにある。これを根本的に解決するために、各種 IT ツールの操作性を高めるため、簡潔な機能や直感的に操作できる学習システムが必要との結論から、学内の学習ツールはすべてこの視点に立って開発してきたものである。

## Cloud Campus の今後

CC はその運用が開始されてから 2 回に分けて、システム基盤を更新してきた。第 1 回は 2012 年から 2018 年まで、LMS 機能の他にコンテンツ制作ツールの提供やスマートフォンやタブレット向けアプリケーションの提供を行った。2019 年からは、これらのツール群やオンライン試験監視システムをひとつのシームレスな Web システムに仕上げ、一貫通貫の統合型ラーニングシステムとした。



図 4 Cloud Campus 上の教育活動：授業設計から成績評価まで

CC の性能向上と機能強化のため、学内での開発は絶え間なく続けられているが、その範囲に留まらず、特徴のある外部教育コンテンツとの連携（英語学習コンテンツやプログラミング開発環境など）も行われてきた。また、学生の履修分析には各種 BI ツール（データ分析）を使用したり、Zoom などのウェブ会議システムをライブ授業に活用したりすることもある。

しかし、最も重要なシステム運用は、その可用性すなわちシステムの安定性にある。ダウンタイムを極力なくすため、各種機能の追加や更新のためのシステム保守において、学生の利用時に障害が起きないように、システムのバージョンアップ時のテストには細心の注意を払っている。

CC は、実はサイバー大学だけのために運用されているわけではない。2022 年 12 月現在、230 を超える企業・団体（内 20 以上は大学）において、160 万人を超える学習者が CC を利用している。いうならば、それなりの規模のオンライン教育基盤となっていることから、新たに企業間、あるいは産学間、大学間での人材育成のための教育活動やコンテンツの共有も進みつつある。その責務は一大学だけの教育活動に留まることがない。オンライン学習システムの継続的研究開発やシステムの安定的運用の社会的責務を果たすために一層努力し続けなければならない。



# メタバースは多様性と包摂性に資する 学びの場となりうるか ーサイバネティックス再考ー

高林 友美<sup>1</sup>

## 1. はじめに

2022 年は日本におけるメタバース元年となった。Facebook 社がメタバース事業への本格参入を意識して社名を Meta に変更した 2021 年や、Ball<sup>1)</sup> が今なお多く参照されるメタバースの 7 つの要件をまとめて発表した 2020 年、もしくはそれ以前、2002 年のいわゆる「早すぎたメタバース<sup>2)</sup>」の Second Life のサービス提供開始を黎明期とせずむしろ元年と呼ぶこともありえるかもしれないが、こと日本国内でのメタバースの活用については、やはり 2022 年がそれに相応しい。身近な例を挙げれば、日本経済新聞社の提供する「日経テレコン 21」でメタバースを含む記事件数を年次ごとに見ると、2020 年は 3 件、2021 年は 102 件であるのに対し、2022 年については 1 月 1 日から 12 月 15 日時点での件数でも 803 件となっており、今年に入ってからメタバースが専門領域を超えて多くの人々の話題に挙がるようになったことがうかがえるだろう。

メタバースは急速に広まる中で、それが何であるかのイメージは定着しつつあるものの、明確な定義は現在でも曖昧である。1992 年の SF 小説『スノウ・クラッシュ』内の造語が語源とされているように、“高度なテクノロジーを用いた近未来的な仮想空間”とすることでおおよその合意が得られるだろうが、現在では特に「専用のゴーグルを付けて自身のアバターを操作して、会話や鑑賞・モノの売り買い・遊戯など、現実世界のように没入してやり取りが出来る仮想空間」と定義すると広く理解されている概念に近いだろう。ただし、ゴーグルを付けた操作は 3 次元の空間であることとも同義であるが、一部では PC 上で行う 2 次元空間のアバター操作も広義のメタバースと呼んでいる場合が見られる<sup>3)</sup>。また、西田<sup>4)</sup> は著書の中でメタバースとは『コンピュータとコミュニケーションに関する進化の帰結であり、過去から続くアプローチ』である」と書いており、メタバースは革新的なテクノロジーであるだけでなく、新しいコミュニケーションの形とも考えられていることが分かる。

本稿ではこのメタバースについて、教育の場での活用、特に現代で必要とされる「多

---

<sup>1</sup> サイバー大学 IT 総合学部・講師

多様性と包摂性」の観点から考察する。メタバースの教育活用については、メタバースの活用可能性を多分野の教授陣が共同執筆して 20 の側面からまとめた論文<sup>5)</sup>でも一側面として挙げられている。国内でも教育的活用に関して今後実践研究や効果検証が進んでいくと思われるが、まずどのような取組がなされたのか、その貢献と限界について特定の視点からまとめておくことには一定の意義があるだろう。本稿で全てを網羅することはかなわないが、その分独自の視点として、メタバース元年となった年に同様に国内で話題に挙がることの増えた「多様性と包摂性」の側面の利用例に的を絞る。多様性と包摂性とは、Diversity and Inclusion を日本語に訳した<sup>6)</sup>もので、SDGs (Sustainable Development Goals: 持続可能な開発目標) が目指している「誰一人取り残さない社会」の原則となる概念を指す。次節および3節で取り上げるとおり、メタバースの教育利用の事例数は増加してきているが、多様性と包摂性に資する教育の場としての議論はまだ少なく、今後研究の発展が期待されるもののひとつに挙げられる。本稿では教育利用の面においてどれだけメタバースが「みんなに<sup>7)</sup>」という側面にアプローチしている／すべきなのかを 2022 年時点の状況を踏まえてまとめたい。

第1節では、最新の技術「メタバース」に関係する問題の所在と本稿の目的を述べた。第2節では現代の課題である多様性・包摂性とメタバース上の学びの場の在り方についてまとめ、第3節において本研究のテーマである多様性・包摂性に対するメタバース技術の貢献と限界を事例と共に述べて最新の状況を整理する。第4節では、20 世紀の新しい通信技術の発展が目覚ましい中、人なのか機械かに関わらず横断的に研究することを可能にしたノーバート・ウィーナーの古典『サイバネティックス<sup>8)</sup>』の論を再訪する。これを第5節のまとめにつなげて、メタバースが多様性・包摂性に資する学びの場となりうるか、展望を含めて議論したい。

## 2. メタバースの日本国内における動向

本節では、サイバー空間上の自身のアバターを操ることで活動できる世界である「メタバース」について、特に国内の動向を簡単にまとめて、本稿の議論の土台としたい。

日本国内のメタバースにはまずヘッドマウントディスプレイの流通があった。日本国内で VR 体験が広まったのは、スマートフォンを紙や段ボールで作った部品にはめて簡易ゴーグルとして使用するものであるが、2016 年には「Oculus Rift (米 Oculus)」「Vive (台 HTC)」「PlayStation VR (日ソニー)」といった現在のヘッドマウントディスプレイを牽引する各社の製品が日本で買える状況になり<sup>9)</sup>、特にゲーム市場が豊かな日本では、高価な端末でも多く流通するようになった。現在では、上記の3社のうち Oculus が Meta に買収されているほか、中国の Pico 社提供のものも VR デバイスとして広く流通している<sup>10)</sup>。特に、自前の VR プラットフォームも大きい Meta のスタンドアロン型のヘッドマウントディスプレイが、日本の「メタバース」の象徴となっているだろう。

この土壌をもとに、2019 年以降の新型コロナウイルスの影響を受けメタバースが更に広く使われるようになった。ステイホームと呼び掛けられる期間の影響が大きかった業界を中心にコミュニケーションツールを援用するニーズが高まり、2022 年には多くの分野でメタバースがその候補となった。第 1 節でメタバースの定義に触れて「会話や鑑賞・モノの売り買い・遊戯」といった行為を挙げたが、前述のゲーム市場での興隆もさることながら、VRChat などの VR プラットフォーム上でのアバターと呼ばれる分身同士の会話ややり取りが流行した<sup>11)</sup>だけでなく、遠くに行かなくても間近に鑑賞できる芸術（国宝<sup>12)</sup>、障がいを持つアーティストの作品<sup>13)</sup>、若手アーティストの作品<sup>14)</sup>等）、総合百貨店<sup>15)</sup>から地元の果物の小売り<sup>16)</sup>まで様々な商品の営業と販売などでの利用が話題になった。モノの売り買いの面ではこれまで通りのメタバース内のアイテムが NFT で保証されるだけでなく、実際に手元に商品が届く事例もある。また、地域振興のために観光地の再現鑑賞と特産物の売り買いを同時に実現した自治体<sup>17)</sup>も特筆すべきだろう。さらには内定式<sup>18)</sup>や各種学校の文化祭<sup>19)</sup>・卒業式<sup>20)</sup>など、人生の節目となる行事においても、実際に集合することの障壁なしに一定の人数が集まることの出来る「場」として活用されている。

前掲の西田は、メタバースで今後重要となる 2 大ニーズとして、PC 代わりのビジネスツール（コラボラティブ・コンピューティング）と、教育利用に注目している。西田によれば、教育面では特に高所作業や工場での危険な研修に活用することでメタバースが身近な道具になっていくが、そのような VR 体験だけではなくメタバースらしい活用として、教材の色や大きさ、立体物の関係をリアルに感じられる授業を、他の学習者の存在を感じられる状況と共に提供している角川ドワンゴ学園 N 高等学校（N 高）の先進事例<sup>21)</sup>を紹介している。ここまでのメタバースの国内動向から、2022 年に多くの場面で登場することになったメタバースは、西田の述べるビジネスと教育での活用のニーズを中心に、もはや「ゲームをする一部の人の道具のひとつ」とは言えない、人々が暮らすうえでどこかで出会わざるをえないものとなっていくであろうことが見て取れる。

### 3. 多様性と包摂性をめぐるメタバース上の学びの場

前節でまとめた通り、メタバースは教育活用を含めて様々な展開がなされている。このうち、本稿が目指す「多様性と包摂性」に関わる教育的利用について、メタバースが貢献している例について詳しく述べる。一方で、メタバースを利用したとしても多様性と包摂性に資する学習環境の実現には届きづらい可能性や、むしろ断絶を加速化させる可能性も考えられる。そのような、メタバース活用の限界が読み取れる例についても紹介する。

#### 3.1. メタバースの貢献

「誰も取り残さない」社会を目指すにあたり、特に教育の機会を十分に得られていな

い人々に対するアプローチとしてのメタバース活用は、本邦における初等・中等・高等教育および社会教育において少しずつ始まっている。これらについて、現段階ではどの学校種に最適であるのかその操作性やカリキュラムの自由度、対応する学齢等の検証は不十分である。しかし、本稿では開始されている、または開始が決まっているプログラムをまとめることで、学びの場に来られる人を増やすというメタバースの貢献の評価としたい。

規模としても、日本国内へのインパクトとしても、東京大学がメタバース工学部を開学した<sup>22)</sup>ことは、メタバースによる教育の拡大への貢献としてまず取り上げたい。東京大学は2022年9月23日に、工学の教育プラットフォームとしてメタバース工学部を始めた。東京大学の学部という名称を持っているが、メタバース工学部の対象者は一般的にイメージされる受験に合格して通学する学生とは異なる。23日の式典の説明では「メタバース工学部は中高生や社会人を含む意欲あるすべての人にとっての学びの場だ<sup>23)</sup>」と述べられており、教授を含む東大所属の研究者が中高生向けの講座と社会人向けのリスキリング講座を提供する。リスキリング講座や、メタバース工学部の名のもとに構築されるキャリア形成情報サイトについては、リクルートやソニーグループ、丸井グループ、三菱電機ほか、産業界との提携があることも、本取組の社会的インパクトを高めているだろう。東京大学工学部では以前から高校生らを対象としたサマースクール<sup>24)</sup>などの工学教育を広めるための活動が行われており、動画配信形式の授業など、これまでの活動をなぞる形ではあるものの、メタバース上に再現された東大の赤門、といった象徴的なものを用意したうえでメタバース工学部“開講”とすることで、より一層のアウトリーチを実現しようとしていると言える。この取り組みについて、メタバース工学部長は多様性に資するものであると述べており、特に現在の工学部では1割にとどまっている女性人材育成を強調するほか、新しいテクノロジーを取り入れて、工学のイメージを変えつつ、年齢・地域を問わないアプローチを目指していることを強調している<sup>25)</sup>。メタバース工学部は、メタバースおよびメタバースという言葉を使うことで、これまでの工学教育の裾野を広げる活動を拡充し、より多くの対象に学びの場を提供していると言えるだろう。

また、義務教育である初等・中等教育段階においても、多様なバックグラウンドに対応するためにメタバースを活用しているとする報告がある。東京都教育委員会は「バーチャル・ラーニング・プラットフォーム」を開設し、まず新宿区でデモ版として実施を開始している<sup>26)</sup>。これは不登校の児童生徒や、海外からの転入で日本語が不自由な段階にある児童生徒を対象としており、仮想空間内の“教室”として不登校支援と日本語指導のフロアが用意されている。各フロアには大きめの教室だけでなく、1対1の相談が出来る場所、自由に交流できる場所が設けられているが、垣根を超えた共用スペースとして“グラウンド”“噴水広場”といった場所もあることから、インクルーシブな環境が構成されていると言っているだろう。新宿区のデモ運用では適応指導の対象者のアバターに対してのみ入室キーが与えられているため、支援が必要な子どものみが使用している状態であり、ある意味では一部の子どもを切り分けていると見られる可能性もあるかもしれないが、今まで学校および教育・支援が届きづかった子どもにもアクセスが可能になるという点か



らみても、社会包摂の向上と見てよいのではないだろうか。今後この「バーチャルラーニングプラットフォーム」を誰でも入室可能なものとして公開していくよりもむしろ、これまでの教育学習支援の延長である本プラットフォームを入り口として、より多様な人々が存在するメタバースの広い世界に踏み出すための梯子となることが望まれる。現在構想段階だが、東京都のほかにも複数の自治体で同様の試みが計画されている<sup>27)</sup>。そのほかでは、小規模であるが高いクオリティで実践されているのが、大学院生が計画・運用して広島市が支援する「メタバース不登校学生居場所支援プログラム<sup>28)</sup>」である。これはVRChatをプラットフォームとして使用しているため本格的なメタバースにおける様々な体験が提供されているだけでなく、アバターを引き続き仮想空間で利用できる可能性が高いことや、メタバースで広く活躍する不登校経験者のロールモデルに出会えることなどから、プログラム自体は短期間であっても、長期的に多様性と包摂性に資するものと言えるだろう。

以上について、2022 年現在ではまだメタバースを活用した修了実績の数は限られているものの、前節で紹介された N 高を含む通信制の教育やその他の支援機関・学校がこれまで重ねた数多くの実績が、メタバースの登場によって技術的側面から注目を浴び、前面に出るようになったとも言える。通信制の教育によって保障されていた学びの場が、より広い範囲で注目され、技術的進歩で没入感の高い形で実現してきていると考えられる。

### 3.2. メタバースの限界

夢の場所のように語られることも多かったメタバースの限界についても知られるようになってきている。仮想世界であるとはいえ、現実世界とある意味「地続き」だと分かるとも言えるかもしれない。ここでは3点に絞ってその限界をまとめておく。

教育に限らず広く知られたメタバースと多様性に関する限界として、メタバース上でアバターを作成し、現実もしくは理想の自分の姿を投影したアイデンティティを表現することが可能であっても、多様性の社会的包摂の真反対にある人種差別や性差別などの枠組みから自由になることが出来ないことが挙げられる。分かりやすい例としては、メタバース上でもセクシャルハラスメントの被害が早くから世界で報告されており<sup>29)</sup> 自身の見た目や空間が変化したとしても、それが即座に「誰もががあるがまま認められる社会」の実現になるのではなく、ユーザーおよびプラットフォーマーの意識・態度そして行動に高い倫理観が必要となる点は注意が必要である。オンライン上のコミュニケーションの匿名性や、現存する差別意識を含んだ教師データの現実・仮想世界への影響など、現代ではむしろ偏見が強化される可能性も大いにある。多様性と包摂性の実現に向けた人々の意識については、引き続き大きな課題となるだろう。ただし、この限界に対してこそ教育的アプローチが盛んに取り組まれていること、特にメタバースに近い技術による実践も始まっていること<sup>30)</sup>を踏まえて、今後、高い没入感を伴って様々な視点を体験することが可能であるメタバースを用いて、まずは人々の意識に呼び掛けていくことが必要である。

メタバースの技術的限界として、着用するヘッドマウントディスプレイの価格及び重さの問題も述べておく。2節で示した通り、VR デバイスはメタバースに欠かせないものであるが、象徴的なものとして広まっていると紹介した **Meta Quest 2** の場合でも 500g 以上の重さと 5 万円以上の価格<sup>31)</sup>は、ひとりひとつを頭に巻き付ける形で使用するものとしては課題である。このデバイスによる制限については、技術的革新を待つことが一つの解決策となる。現在のところ、Meta の研究機関である **Reality Labs** が 2022 年 6 月に公開している多数の試作<sup>32)</sup>からは、重さ、価格のほかにもピント合わせや装着のしやすさについてまで改良が期待される。ただしあくまで試作であり、実際に販売できる予定の見通しは示されていない。安価で軽いヘッドマウントディスプレイが実現するまでの今後長い期間においては、上述の広島市が支援する「メタバース不登校学生居場所支援プログラム」のような、公的支援で現状のデバイスが無償で貸し出す形のフォロー<sup>33)</sup>に留まるだろう。多様性と包摂性のためのメタバースを考えるときには、量だけが重視されるべきではなく、質が保証される形でスタートしていくことに問題はないものの、今後の展開としてはやはり大人数への対応が難しいという課題が残されることになる。

最後に、現在のデバイスでは人間の五感のうち視覚と聴覚に大きく頼る形であることも、多様性と包摂性に資する学びの場としての限界となりうる点も記しておく。視覚と聴覚は人間のコミュニケーションで主たる感覚だと言われやすいが、その2つに大きく頼った状態にあるメタバースは、その特性によって参加できる人を制限してしまう<sup>34)</sup>。また、ヘッドマウントディスプレイの装着に困難がある人々も多く、身体的制限だけでなく発達障がいによる感覚過敏の問題も例外ではない。新型コロナウイルス感染症対策としてのマスク着用においては、感覚過敏の特性を持つ場合の困難<sup>35)</sup>が取り上げられたが、ヘッドマウントディスプレイについても同様の問題が発生しうる。さらに広く言えば、いわゆる「VR 酔い」をしやすい体質についても、メタバースの多様性を制限するものと言えるだろう。この点に関しては、状況にリアルタイムに応じた音声読み上げや文字起こし・字幕の機能など、アクセシビリティへの配慮による解決が期待される。前段落で述べた **Reality Labs** の試作には、一般的な眼鏡と同様の付け心地を想定した高性能ヘッドマウントディスプレイも含まれており、今後の改善が望まれる。その過程では、障がいのある人々がメタバースに自由に参加できるようになるだけでなく、VR デバイスや VR プラットフォームの改善の過程において、より多くの人が視覚や聴覚のない世界を疑似的に体験するようなコンテンツが作られたり、様々な障がいを超えて対話することが出来るようになったりする可能性がある。1点目の限界で述べたことと共通する点として、現在行われている多様性と包摂性の実現にむけた異文化理解教育プログラムに類するものが充実し、現在の限界を乗り越えた人々の学びに繋がる可能性も残されているだろう。

#### 4. サイバネティックスの視点によるメタバース考察

ここで、現状の評価と今後の展望のために「新しい」コミュニケーションに関する古

典である『サイバネティックス』を訪れ、考察を深めたい。サイバネティックスはノーバート・ウィーナー（1894-1964）による造語で、ギリシャ語で「舵手（舵を取る人）」を意味する *kybernētēs* から着想された、制御と通信（Control and Communication）の研究の総称である<sup>36)</sup>。総称とするのは、制御と通信の課題として、変数を調整することで結果を変えることについては、それが生物のものでも機械のものでも本質的には統一されるというウィーナーの考えによるもので、それまでの分野を横断する新しい研究領域となった。彼の提唱したこの新しい分野の研究は、現在の AI の研究を支える基盤にもなっている<sup>37)</sup>。サイバネティックス自体はインターネット通信もまだ発展途中の頃に考えられたものであるが、当時の新しい技術の発展に大きく影響を与えたことに鑑み、現代のメタバースの活用にウィーナーの著述を当てはめて考えてみる。本稿では限定的な側面とはなるものの、サイバネティックスの領域横断的な視点と、テクノロジー活用の目的の点の2つに絞って考察し、現状と展望を論じる。

#### 4.1. 制御と通信という視点：垣根を超えて通信・伝達を見ることへの示唆

まず注目されるべきは、サイバネティックスの生物と機械という垣根を越える考え方である。人がデバイスを装着して仮想世界に飛び込むことになるメタバースでは、生物の要素と機械の要素の両方を、『サイバネティックス』に出てくる表現を用いるならば「ひっくるめて<sup>38)</sup>」考える必要がある。ヘッドマウントディスプレイのアクセシビリティを向上させる過程においては、重量や価格というハード面の問題と、優先される五感や VR 酔いという人間側の問題との両方の内面まで理解し研究されることで、「多様性と包摂性に資する学びの場」という結果に辿り着くことが可能になるだろう。

垣根を越えて考えることによって、人々のイメージという点もまた大きな影響力を持つことを議論に持ってくる事が出来る。実のところ、メタバースという新しくもてはやされる言葉を使うことが人々に与える印象による影響を考えることは、メタバースの技術的影響と同程度に意味を持つ可能性がある。ウィーナーの場合、関係する要素を人体・機械に関わらずすべてをコミュニケーション（通信）成立のために見渡そうとする。現代においてメタバースをサイバネティックスの視点からとらえる場合には、メタバースが持つ、明確な定義が落ち着かないままであっても、新しいものとして急騰して注目されるという力が、これからの学びの場を拡張するために資する重要な点であると論じることが出来るのではないか。前節2項では、多様性を受け入れるための意識が醸成できるかという課題が仮想空間であっても多様性・包摂性実現の限界となりうることを述べたが、メタバースというものがこの元年これだけ広まったことを踏まえて、その印象が人々の生活、特に学びの場に向かおうとする力にポジティブに影響を持ちうることを活用できれば、メタバースが多様性と包摂性に資する学びの場となる可能性は高いと言えるだろう。

人は飽きやすい。サイバネティックスが理解されづらく、その考え方でも生物の要因を含むことに困難が生じがちであるのは、飽きやすさなど、生物が機械よりも制御が難し

いことにも由来するだろう。サイバネティックスの視点に基づいて上記のようなメタバースの力をとらえる場合には、その注目と熱を数値化して制御する方法などを考える必要があるが、その試みが人の関心の変わるスピードに追い付くのは簡単なことではない。ウィーバーの頃とは異なり、現代では SNS を含むオンライン上の言葉を集め数値化に近いことが可能であるが、精度に難が残る。彼の研究への取り組み方を踏まえると、数値化の技術を更新し続けるだけでなく、同時に、ノイズを含むデータを用いてサイバネティックスが対象とするすべてを活用して目的に向けた制御を検討することが望ましい。

#### 4.2. 多様性・包摂性のための技術：技術を向社会的に活用することへの示唆

ウィーナーを現代に再訪するもう一つの意味に、技術活用の目的の向社会的性に関する議論がある。本稿では目的にあたることとして多様性と包摂性の実現を置いて論を進めてきたが、この目的という点もまた彼の著作から示唆を得ることが可能である。彼は生年と没年からも分かる通り、二度の大戦を経験した科学者である。実際に彼が計画していた研究が中断したり、戦時研究計画に参加したりした経験も『サイバネティックス』の序章に語られている。戦争を経験した科学者の多くがそうであるように、彼もまた技術活用の目的について、原子爆弾のもたらした結果についても触れながら、新しい研究分野であるサイバネティックスの平和的活用に心を砕いていたことが見受けられる。サイバネティックスの学会立上げ時に招聘した学者とその専門領域についての記述からも読み取れる通り、彼の考える戦争と搾取とは縁遠い分野に絞ってサイバネティックスの研究成果を臨床に応用することを重視していた。時に思考を数式にして表現することを好むような人物が、技術応用や理論を机上でのみ考えるのではなく“より良い”応用を目指していたことは、現代の新しい技術の利用と研究に関して示唆的である。

具体的には、『サイバネティックス』では義肢の技術向上のための通信技術の応用<sup>39)</sup>が語られており、また、初版サイバネティックスの解説書として数学的な記述を減らして哲学的側面をより多く説明する形で出版された『人間機械論<sup>40)</sup>』では聾<sup>41)</sup>やパーキンソン病<sup>42)</sup>の治療に研究を活用することの言及がある。サイバネティックスという考え方の着想からも見て取れる通り、ウィーナーの専門性は多岐にわたるが、その中でも上記の通りの障がいや病についての言及が複数見つかることは、メタバースの活用を考えるうえでも示唆的だと考えられる。すなわち、少数者も見過ごされることなく最新の研究と技術の恩恵を享受することの重要性が説かれていると言ってもよいだろう。

ウィーナーは戦争の話題と同じ章で産業革命期の搾取についても言及しており、現在のメタバースの主に企業が技術開発を主導する状況は、一部の特権階層のみ潤うような恩恵の分配の問題を憂う彼の望むようなものではないかもしれない。彼の示唆するところを現代において解釈すれば、メタバースの利用目的に関しても公的な支援や研究機関の介入、そして社会・公衆による関心が期待される。彼の時代とはまた異なる形で階層構造がある現代では、企業が先導するからこそ達成されるものもあるだろう。SDGs が叫ばれる以前

から、国内では長らく技術者倫理教育<sup>43)</sup>についても、その意義を含めて産学官で盛んに議論されてきた。企業には利益追求だけでなく高い倫理性も求められる時代だからこそ、サイバネティックスが目指していた新しい技術の活用のあるべき姿として、メタバースは今後も多様性と包摂性等の社会課題に積極的に活用され続ける必要があると考えられる。

## 5. まとめと今後の展望

本稿では、2022 年に日本で興隆したメタバースの教育的活用について、特に多様性と包摂性という視点から議論した。国内における数々の実践から、新しい技術が多様性と包摂性に資する教育的活動に用いられており、その背景と意義はウィーナーのサイバネティックスの考え方で説明できることが分かった。ウィーナーの著述の理解については更に深めていく必要があるものの、社会的課題に対する新しい技術の活用を多面的に考えるにあたっては、現代においてもサイバネティックスが示唆に富むことを確かめたうえで、これからのメタバース活用についても見えてきた。リスクリングを含む生涯学習において、メタバースを含む多くの場所での学びを進める舵を取るのはあくまでも個々人である。しかしより良い舵取りのために、調整できるものと調整できないものを確認したうえで、いかにして方向を決めていくと良いのか、先人に倣ってフィードバック機構を働かせる「モデル」の開発は、そうした個人を支えるものとして広く役に立つ可能性がある。ウィーナーの方法であれば、それは図式化もしくは数式化であり、今後に残された課題となる。これは稿を改めて論じたい。

メタバースのような新しい技術については、言葉自体のインパクトを低く見積もることが出来ないが、それを一つの要因とおいた上でなお、N 高しかり、東大工学部しかり、言葉が流行する以前から地道な取り組みがあったからこそ、メタバースに新しく注目が集まった際に目的に応じた活用が可能となった。もし「多様性と包摂性」や「メタバース」が、いわゆるバズワードとして一瞬の熱の後には関心を集めなくなってしまうとしても、望ましい目的のために地道な働きかけが続けられれば、また次の追い風に乗って新しい技術の活用が可能となるだろう。多様性と包摂性に資する学びの場を築くためには、新しいテクノロジーの持つ影響力と、その時代において良いとされる目的への関心と、そして弛まぬ実践の積み重ねとが必要であり、少なくとも 2022 年のメタバースにはそれらが揃っていたために数多くの貢献を果たすことが出来たと考えられる。

日本で初めてのオンライン大学（学士課程）は、それまであった大学を壊したり飲み込んだりするのではなく、新しい方法を高等教育の世界に提供し続けることでその存在価値を増している。実のところメタバースも同様ではないだろうか。新しい技術の魅力には高い関心が集まるが、現状のものをすっかり置き換えるものにはそうそうならず、しかしオプションとして一部の人々にとってかけがえの無い場所として活用されていくことがこれからのメタバースとして予想される。その過程では、新規性に注目が集まり大きく躍進

する時もあれば、地道な活動を積み上げる時もあるだろう。必要になるのは、サイバネティックスの視点を持って、様々な要素を踏まえながらより良い方向に舵を取ろうとし続けることだと考えられる。メタバースを含むあらゆる技術について、これからも進むべき道の議論とその“フィードバック”としての実践の継続、およびそれらの循環が望まれる。

注および参考文献

- 1) Ball, Matthew “The Metaverse: What It Is, Where to Find It, and Who Will Build It” *MathewBall.vc*, 13 Jan. 2020, <https://www.matthewball.vc/all/themetaverse> (2022 年 12 月 15 日確認) にて、7つの条件 (①リセット・一時停止・エンディング等のない永続するものであること、②現実世界のように同期して体験できること、③同時並行で接続し、その場所に存在感を感じられるユーザーの数に限りがないこと、④完全に機能する経済性を持つこと、⑤デジタルとフィジカル、私的と公的、オープンとクローズドなど対になる世界・ネットワークの両方にまたがること、⑥デジタル世界の様々なサービスとの連携性が高いこと、⑦個人・企業を問わず幅広い貢献者によって成り立つこと) が発表された。なお、Ball は 2 年後の著作においてはメタバースの定義を以下のように述べている。「リアルタイムにレンダリングされた 3D 仮想世界をいくつもつなぎ、相互に連携できるようにした大規模ネットワークで、永続的に同期体験ができるもの。ユーザー数は実質無制限であり、かつ、ユーザーは一人ひとり、個としてそこに存在している感覚を有する。また、アイデンティティ、歴史、各種権利、オブジェクト、コミュニケーション、決済などのデータに連続性がある。(ボール『ザ・メタバース』井口耕二訳、飛鳥新社、2022、p. 53. (Ball, Matthew. *The Metaverse: and How It will Revolutionize Everything*. Liveright Publishing, 2022.))」
- 2) ITmedia 岡田有花「“早すぎたメタバース” Second Life とは何だったのか 再ブームはあり得るか？」ITmedia News、2022 年 1 月 18 日  
<https://www.itmedia.co.jp/news/articles/2201/18/news091.html> (2022 年 12 月 15 日確認) .
- 3) 西田宗千佳『メタバース×ビジネス革命―物質と時間から解放された世界での生存戦略―』SBクリエイティブ、2022.
- 4) 例えば、藤井美沙紀「メタバースで不登校児を支援 その狙いは 埼玉・戸田市」NHK さいたま放送局首都圏ナビ 2022 年 10 月 19 日、<https://www.nhk.or.jp/shutoken/saitama/article/006/49/> (2022 年 12 月 15 日確認) では、ファミコン時代の RPG 世界に似た 2D 空間と Zoom などを繋ぐ Gather というシステムを使うことを「メタバース」と呼んでいる。Gather は一般的に仮想空間であることのみが強調され、メタバースと呼ぶかは定義によって異なる。
- 5) Dwivedi, Yogesh K., et al. "Metaverse beyond the hype: Multidisciplinary perspectives on emerging challenges, opportunities, and agenda for research, practice and policy." *International Journal of Information Management* 66, 2022. <https://doi.org/10.1016/j.ijinfomgt.2022.102542>  
多様な分野・研究機関の研究者 42 名による、職場、ゲーム、健康管理ほか 20 分野でのメタバースの利用可能性の展望論文。教育に関しては、実体験が重要なが危険である教育に関して視聴覚教育よりも有効である可能性や、美術館などの社会教育がより多くの人々に届く可能性等が述べられている。一方注意点として、内容によっては感触・詳細に現実との相違があること、監視とプライバシーの問題、学習内容を現実世界へ応用する能力の必要性が挙げられている。
- 6) より正確な訳として「多様性と包摂」とする場合もあるが、本稿では 2022 年現在より広く使われている語を採用して「多様性と包摂性」とする。

- 7) SDGs の 17 の目標のうち直接教育に関わるものは「4. 質の高い教育をみんなに」である。
- 8) ウィーナー『サイバネティックスー動物と機械における制御と通信ー』池原止戈夫・彌永昌吉・室賀三郎・戸田巖訳、岩波書店、2011. (Wiener, Norbert. *Cybernetics or Control and Communication in the Animal and the Machine*. Second Edition. MIT press, 1961.) .
- 9) 日本経済産業新聞「VR 争奪戦 眼前にせまる」日本経済産業新聞、2016 年 3 月 16 日、p. 1.
- 10) 日本経済新聞社『日経業界地図 2023 年版』日本経済新聞出版、2022.
- 11) *ibid.*
- 12) 東京国立博物館「東京国立博物館バーチャル展示 エウレカトーハク! ●89」東京国立博物館 [https://www.tnm.jp/modules/r\\_event/index.php?controller=dtl&cid=5&id=10805](https://www.tnm.jp/modules/r_event/index.php?controller=dtl&cid=5&id=10805) (2022 年 12 月 15 日確認) エウレカトーハクの本展示は 2023 年 1 月からの予定だが、11 月より一部プレオープンし、東京国立博物館初の NFT アート作品も販売される。
- 13) 岩崎貴行「『芸福連携』で心豊かに 首都圏、障害者ら創作に力」日本経済新聞地方経済面東京、2022 年 11 月 17 日、p. 5.
- 14) 東京アニメセンター「昭和百年展 in 渋谷 芸術祭 2022」DNP プラザ渋谷・東京アニメセンター、[https://tokyoanimecenter.jp/event/showa100\\_sas/](https://tokyoanimecenter.jp/event/showa100_sas/) (2022 年 12 月 15 日確認) .
- 15) 日経 MJ「メタバース、小売業も注目、3D モデル、EC 販売へ誘導」日経 MJ (流通新聞)、2022 年 2 月 28 日、p. 11.
- 16) 日本経済新聞「メタバースで果物販売、アバター同士で商談も 山梨の農業法人 販路拡大狙う」日本経済新聞夕刊、2022 年 10 月 8 日、p. 7.
- 17) 泉洗希「メタバースを街づくりに反映 関西鉄道各社、移動需要を喚起」日経産業新聞、2022 年 12 月 1 日、p. 10.
- 18) ANN NEWS「来春新卒の採用内定が解禁 メタバースでアバター参加の式も」テレビ朝日テレ朝 news、2022 年 10 月 3 日、[https://news.tv-asahi.co.jp/news\\_economy/articles/000270484.html](https://news.tv-asahi.co.jp/news_economy/articles/000270484.html) (2022 年 12 月 15 日確認) .
- 19) 桜井芳野「学校行事ぐるり 360 度撮影し配信 船橋の新興がレンタル 共働き世帯の視聴見込む」日本経済新聞地方新聞面神奈川、2022 年 12 月 10 日、p. 26.
- 20) 学校法人角川ドワンゴ学園 N 中部部 N 高等学校 S 高等学校「令和四年度 N 中部部・N 高等学校・S 高等学校メタバース入学式特設サイト」KADOKAWA DWANGO educational institute、[https://nnn.ed.jp/lp/entrance\\_ceremony\\_2022/](https://nnn.ed.jp/lp/entrance_ceremony_2022/) (2022 年 12 月 15 日確認) .
- 21) 西田 op.cit. なお、本書でも紹介されている N 高等学校のメタバース授業は、同校の式典や学内行事とは異なり、オンデマンド型である。生徒たちは自身の好きな時間に受講できるが、向いていた方向や教具オブジェクト (例えば数学の授業中の立体物など) を触っている様子が記録され、他の生徒が受講時にアバターの動きが再生される。非同期オンライン授業で「ゆるやかに」他の生徒の存在を感じることが出来る設計になっている。
- 22) 東京大学工学部・東京大学大学院工学系研究科「メタバース工学部メインサイト」東京大学工学部、<https://www.meta-school.t.u-tokyo.ac.jp/> (2022 年 12 月 15 日確認) 時期にもよるが、ジュニア講座・社会人講座ともに平日夕方からの全 2 ～ 13 回程度の講座が複数用意されてお



## メタバースは多様性と包摂性に資する学びの場となりうるかーサイバネティックス再考ー

り、東大教授陣の授業はオンデマンド配信でも出席できること、ゲストレクチャーの後日配信はないことなどが確認できる。社会人講座の受講は法人申込が必要。

- 23) 日経 MJ「東大、メタバース工学部開講」日経 MJ（流通新聞）2022 年 9 月 28 日、p. 2. 染谷隆夫工学部長のあいさつより。
- 24) 東京大学大学院工学系研究科「サマースクール 2022『宇宙を拓く』」東京大学工学部、<https://sites.google.com/g.ecc.u-tokyo.ac.jp/summerschool2022>（2022 年 12 月 15 日確認）。
- 25) 日経産業新聞「直談 専門家に問う 中高生にキャリア示す 工学分野の女性人材育成」日経産業新聞、2022 年 11 月 21 日、p. 7.
- 26) 浜崎陽介「仮想空間で不登校や外国の子どもたちの学習支援 都教委、新宿区で 12 日に先行して開校」東京新聞朝刊、2022 年 12 月 9 日、<https://sukusuku.tokyo-np.co.jp/education/64031/>（2022 年 12 月 15 日確認）。
- 27) 篠原犀佑「民間と連携し課題解決 行政の DX 待ったなし 戸田・菅原市長に聞く」日本経済新聞地方経済面埼玉、2022 年 9 月 2 日 p. 40.
- 28) メタバース不登校学生居場所支援プログラムゆずあっと <https://www.yuzuatto.com/>（2022 年 12 月 15 日確認）なお、プログラム名は「学生」となっているものの、文科省の 30 日間の欠席定義による不登校を採用しており、初等・中等教育が対象となっている。
- 29) Basu, Tanya, “The metaverse has a groping problem already” MIT Technology Review, 2021 December 16, <https://www.technologyreview.com/2021/12/16/1042516/the-metaverse-has-a-groping-problem/>（2022 年 12 月 15 日確認）。
- 30) 例えば、Shinichi, Sato. “Virtual Reality for Educational Purposes -Trend and Examples in Japan-” The 2nd Virtual Reality Educational Forum, Keynote Speech, 2019 March 14. の佐藤教授の学生による実践では、福祉大学の学生が車椅子に乗った状態で通学路の全地球映像を撮影し、身近な場所にあるバリアに気づく試みが行われている。全地球映像の再生には、メタバースの重要なアイテムであるヘッドマウントディスプレイが以前から使用されている。
- 31) Meta 「Meta Quest 2 オールインワンの VR ヘッドセット」<https://www.meta.com/jp/quest/products/quest-2/>（2022 年 12 月 15 日確認）および実測。なお価格については円安等の影響で 2022 年に改定されたあとのものである。
- 32) 西田宗千佳「初公開されたメタ社の次世代 VR ゴーグル研究に見る本気度…目標は『現実と区別がつかない体験』」Business Insider、2022 年 6 月 21 日、<https://www.businessinsider.jp/post-255670>（2022 年 12 月 15 日確認）。
- 33) メタバース不登校学生居場所支援プログラムゆずあっと op.cit. および、立命館大学「メタバースを用いた不登校学生の居場所支援プログラムを実施」立命館大学 HP、2022 年 10 月 14 日、<https://www.ritsumei.ac.jp/news/detail/?id=2832>（2022 年 12 月 15 日確認）による。本プログラムは広島市社会福祉協議会の助成事業であり、申込受付時点で 5 名分のヘッドマウントディスプレイ無償貸与が用意されており、実際に 3 名の広島市在住の高校生が 2 週間のプログラムを修了した。
- 34) Alexiou, Gus., “Is The Metaverse Likely To Be Accessible And Inclusive Of People With

- Disabilities?" Forbes Online, 31 Mar. 2022, <https://www.forbes.com/sites/gusalexioiu/2022/03/31/is-the-metaverse-likely-to-be-accessible-and-inclusive-of-people-with-disabilities/> (2022 年 12 月 15 日確認) および、Haynes, Alison. and Alsaab, Halifa., "Will the Metaverse be Accessible?" *Inside Equal Access* Oct. 2022, Kent State University, 2022.
- 35) 日本経済新聞「発達障害の 56%マスク『困難』一着用不快・意思疎通の壁」日本経済新聞夕刊、2022 年 9 月 9 日、p. 11.
- 36) ウィーナー op.cit.によると、サイバネティックスという言葉は、調速機と船の操舵機という、結果に応じて変量を調整してより良い結果を導こうとする行為（フィードバック機構の役割）に馴染み深い 2 つのものを由来にしている。舵手を意味するギリシャ語 *kybernētēs* は、そのラテン語訛りから *governor* という言葉が生まれている。ウィーナーは、その後の制御工学の起点となる、蒸気機関の回転速度を調整する「調速機 (*governor*)」の仕組みを微分方程式でモデル化した 1868 年のマクスウェル論文が、フィードバック機構に関する研究全体にとって重要であることを強く主張しており、元々の語源「舵手」以上に「調速機」との縁を重視して造語を着想した可能性が考えられる。なお、ウィーナーは愛読書が百科事典であったという逸話で知られるように幅広い分野に造詣が深かった。その分、彼の書くものは本人にとって話が繋がっても他者には文脈が分かりづらい箇所が見受けられる。また、サイバネティックスの日本語版に寄せた前書きでは、船の操舵機のみを例に挙げて定義の説明をしている。
- 37) 『サイバネティックス』ではヒトの神経系のフィードバック機構の応用について複数個所で述べられており、同内容については、米サイバネティックス学会の立ち上げメンバーでのちに会長も務めたイリノイ大学医学部マカロック (『サイバネティックス』日本語訳においてはマッカロ博士と表記される) が、同じくサイバネティックス学会創設期に活躍し、数学及び生物学の若手研究者だったピッツと共に、ニューラルネットワークの原型となるモデルを発表して (McCulloch, Warren S., and Walter Pitts. "A logical calculus of the ideas immanent in nervous activity." *The bulletin of mathematical biophysics* 5, 1943, pp.115-133.)、現代の AI の発展の礎となっている。
- 38) ウィーナー op.cit., p. 45.
- 39) *ibid.*, pp. 71-74.
- 40) ウィーナー『人間機械論 一人間の人間的な利用』鎮目恭夫・池原止戈夫訳、みすず書房、2014. (Wiener, Norbert. *The Human Use of Human Beings: Cybernetics and Society*. Revised Edition. Anchor, 1954.)
- 41) *ibid.*, pp. 181-182.
- 42) *ibid.*, pp. 173-174. なお『サイバネティックス』pp. 210-211 でもフィードバックの説明に神経の例えを用いる際にパーキンソン病の言及があるが、治療法への応用の話ではない。
- 43) 札野順「技術者倫理教育, その必要性, 目的, 方法, 現状, 課題」工学教育 54-1, 2006, pp. 16-23.

# プログラミング教育における 「写経型演習先行-文法学習後行方式」の実践報告

松本 幸子<sup>1</sup>

## 1. はじめに

プログラミング学習において、初学者がテキスト型プログラミング言語を使用したコード作成技能を習得する過程を考えると、最初にプログラミング言語の文法を学習し、次に教材に示されたプログラムを模倣する段階を経て、自らコードを考えプログラムを作成する段階へと学習を進めていくのが一般的である。こうした過程の中でも、特に、模倣により提示されている実行可能な完全なプログラムのコードを打ち込み、プログラムの実行結果を確認する段階の学習は「写経型学習」と呼ばれている<sup>1) 2) 3)</sup>。

初学者にとって、この写経型学習の過程は、プログラムの動作がコードの記述を反映したものであることを認識し、プログラミング言語の命令や構文の意味を理解するために欠かせない過程である。しかし、形骸化した写経型学習を漫然と行っても、コードの記述とプログラムの動作との対応関係を認識できず、コードの意味についての理解が進まないことも多い。そこで、筆者の担当するサイバー大学（以下「本学」と呼ぶ）におけるC言語の初学者向けのプログラミング教育科目（以下「本科目」と呼ぶ）では、2017年度より、各授業回の文法学習をまとめて行わずに、小さなステップに分けて最小限の知識のみを元にした写経型演習を繰り返し、全ての演習を終えた後に文法学習を行う「写経型演習先行・文法学習後行方式」を導入している。これは、文法などの知識学習よりも、実際の使用場面での具体的な意味を理解する経験の積み重ねにより言語の習得を目指そうという、自然言語習得に関する研究を参考にした取り組みである。そして、この方式は学生からもおおむね好評を得ている。本学は全ての授業をインターネット上で提供する通信制大学であるため、これらの学習活動は全てインターネット上で展開されている。本稿では、この完全eラーニングによる「写経型演習先行・文法学習後行方式」のプログラミング教育実践について報告する。

---

<sup>1</sup> サイバー大学 IT 総合学部・准教授

## 2. 第二言語学習を参考にしたプログラミング言語学習方略

### 2.1. 第二言語学習における問題との類似性

従来、高等教育機関におけるプログラミング教育科目では、授業の中で、①プログラミング言語の文法解説、②その文法事項が用いられた例題プログラムを模倣しながらのコード入力と実行結果の確認を行う演習、③例題プログラムの改変を中心とした基本演習、④学習者が自ら考えてコードを作成する応用演習、という順で学習を進めていくのが一般的な学習順序であった。このうち、④の応用演習は授業の事後学習として課される場合も多かったが、文法学習と例題プログラムの模倣と改変による演習に取り組んだだけでは、自らコードを考えてプログラムを作成するところまでのプログラミング言語運用能力を発達させるのが難しいことが課題であった。

一方、第二言語（母語以外の自然言語）を学習する場合にも、文法学習、模倣による学習、改変による学習、応用学習、というのは一般的な学習順序ではあるが、長らく日本でも学校での英語教育に対して指摘されてきたように、精読（Intensive Reading）に取り組みながら語彙や構文などの文法を学習し、例文の模倣や改変による練習を行うだけでは、自ら作文や発話を行う応用段階に進んでいくのが難しく、なかなか実用的な言語運用能力が身につかないことが課題であった。こうしたプログラミング言語学習とも類似した問題をかかえる中、近年ではこれを解消する学習法として、文法や語彙などを明示的な知識として学習するよりも、その言語での表現に触れて理解する経験の積み重ねによって習得するという、言語に接する経験量を重視した「多読」(Extensive Reading)「多聴」(Extensive Listening) といった学習法が注目されている<sup>4) 5) 6) 7)</sup>。

### 2.2. 第二言語学習におけるインプット重視の学習法

第二言語習得研究においては、「読む」「聞く」という学習者が言語を用いた表現に接する経験、すなわち「インプット」が重要とする研究が数多く存在する。Krashen は、第二言語の能力を発達させるには、意識的な「学習」(learning)と無意識的な「習得」(acquisition)という区別された2つの方法があるとした。そして、意識的な「学習」で身につけた能力では、学習者が発する表現すなわち「アウトプット」をモニターし修正を加えることしかできず、「アウトプット」自体は母語習得過程と類似の無意識の「習得」で得た能力によってのみ産み出されるもの、「習得」は「インプット」の内容を理解する時に起こるもの、などとした仮説を唱えた<sup>8)</sup>。

この、文法学習やアウトプットの練習などといった意識的「学習」は、言語習得に寄与しないとする Krashen の仮説に対しては異論も多い。しかし、Krashen はアメリカの学校における第一言語指導の研究から、語彙、語法、綴りなどの技能を意識的に学習させ誤りを修正させるような直接的指導ではリテラシーを発達させられず、自由読書だけが効果

をあげていることを示した。そして、「言語を覚えること、つまり習得は、情報のインプット（入力）によるもので、情報のアウトプット（出力）によるものではありません。また、理解によるものであり、表現によるものではありません」<sup>9)</sup>と述べるなど、言語の知識を学習するよりも言語により表現された内容の理解に主眼を置くべきとして、インプットの重要性を強く主張した。

インプット重視の学習法としては、Day & Bamford は学習者が辞書などに頼らずに自ら理解できるレベルの本を自由に選んで楽しみながら大量に読む「多読」(Intensive Reading)を提唱した<sup>4)</sup>。日本では、酒井らが英語学習法として100万語という読書量の目標を掲げ、多読三原則（辞書を引かない、わからないところは飛ばす、つまらなければ途中でやめ別の本へ）に則り、英語学習者向けに語彙、文法、文長が制限された読本(Graded Readers)を読んでいく学習法を提唱した<sup>5) 6)</sup>。さらに、読書の際に読本の朗読を聴く「多聴」(Intensive Listening)を組み合わせる手法も提案されている<sup>7)</sup>。多読多聴学習は、学校教育へ取り入れるにあたっては、多くの読本を用意する必要があるなど導入ハードルが高いため、学校教育における広がりには限られる。しかし、多読多聴について多くの学習効果が報告されており<sup>10) 11) 12)</sup>、インプットの重要性については広く認知されるようになっていく。

### 2.3. プログラミング言語学習におけるインプット重視の学習方略

プログラミング言語は、自然言語に比べれば、圧倒的に文法は単純で例外もなく、覚えるべき語数も少ない。しかし、これらの知識を学びさえすれば、プログラミング言語が習得できコードが書けるようになるというわけではなく、具体的な用例を通じた習熟が必要となる。そればかりか、文法学習、模倣による学習、改変による学習、という一般的な学習段階を踏んでもなお、自らコードを作成する応用段階に至るのは難しい。それならば、プログラミング言語学習においても、第二言語学習において効果を上げているインプット重視の学習方略が効果的なのではなかろうか。プログラミング言語も「言語」というからには、プログラムを作成するというアウトプットの練習や、それに先立つ知識としての文法学習よりも、インプットを増やすことを重視した学習方略、つまり、多くのプログラムのコードに触れ、その意味を理解する経験を増やし、経験の積み重ねによってプログラミング言語を習得しようという学習方略には効果があるはずである。

ただし、プログラミング言語によって記述されている内容は、実際にそのプログラムを実行してはじめて、プログラムの動作として認知されるものである。自然言語で表現されている内容であれば、前後の文脈や挿絵、話者の身振り手振りなど、読む・聴くというインプットと共に認知される情報から類推し、内容を理解することも可能であろう。しかし、プログラミング言語で書かれているプログラムの場合、その内容を理解するにはプログラムの動作を観察することが不可欠となる。つまり、提示されたプログラムのコードと実行時の動作とをあわせて観察する演習を通して、コードの記述の意味を理解する経験を増や

すことが重要となる。

こうした演習では、必ずしも学習者が自らタイピングしコード入力を行う必要はない。しかし、完全にタイピングによるコード入力を省いてしまうと、学習者にはコードが長い呪文のように見えてしまい、構造や意味を持った記述として認識できなくなってしまう懸念がある。また、プログラミング言語は自然言語と違い、僅かな綴りや記号の違いでも意味をなさなくなる、あるいは、意味が変わってしまうという点も、学ぶべき事項の一つである。したがって、コードの理解に影響を及ぼさない範囲でコード入力の手間を省く工夫は施しつつも、基本的には、学習者が自らタイピングすることにより提示されたプログラムのコードを丸写しして入力し、プログラムの動作を確認するという「写経型」の演習を学習の中心とすべきであろう。

結局、第二言語習得研究を参考にすれば、文法学習や学習者自身がコードを考えてプログラムを作成する演習よりも、まずは写経型の演習によってコードの意味を理解する経験を積み、経験を通してある程度まで言語が理解できるようになった段階で、その経験を振り返り整理する形で文法学習を行い、その後に自ら考えてコードを作成する応用演習へと学習段階を進める、という学習方略が有効であろうと考えられる。

### 3. プログラミング言語学習に特有の課題とその対策

プログラミング言語学習には、第二言語学習の場合と類似した問題が存在することは前述の通りであるが、さらにプログラミング言語学習に特有の事情というものも存在する。第一に、そもそもプログラミング言語で表現されている内容は、日常生活では使われていない、プログラミング特有の概念に基づいたものである。第二に、プログラミング初学者は、プログラミング特有の概念の表現が可能な既知の言語を持たず、比較対象となる言語の知識を持ち合わせていない。第三に、内容を伝える相手は人間ではなくコンピュータであるため、些細なタイプミスによってもプログラミング言語を介したコンピュータへの内容伝達に失敗するなど、プログラム実行までには困難が伴う。第四に、写経型学習においてはコードの意味理解という本来の目的が見失われやすい。本章では、これらの課題に対する方策を検討する。

#### 3.1. プログラミング特有の概念に対する理解の重要性

第二言語学習の場合、学習している例文の内容は、母語である第一言語では表現することができていた内容であり、学習者にとっては既知の概念である場合がほとんどである。そのため、インプットを重視した多読学習のみならず従来の学習順序であっても、模倣や改変によるアウトプットの練習を通して、第二言語による表現とそれが示す概念との対応関係を認識することが可能であった。これに対し、プログラミング言語の初学者は、そも

そもプログラミング言語で表現されている内容そのものに対し、概念イメージを持っていない場合がある。

プログラミング言語は、変数などに対する操作として処理を記述し、さらにその処理の流れを作る制御構造を記述する言語である。そのため、プログラムのコードを理解するには変数、条件分岐、繰り返しといった抽象概念の理解が欠かせない。しかし、これらの概念は日常では接することのない概念であるため、学習者が類似の概念を学習したことがない場合には、これらの概念のイメージが形成されていないものと考えられる。こうした概念イメージを持たない初学者にとっては、例題プログラムのコードを入力しプログラムの実行結果を確認するという写経型学習に取り組むだけでは、コードとそのコードが表現している内容との対応関係がなかなか認識できない。つまり、写経型学習が、コードの意味を理解し、プログラミング言語の命令や文法を習得していくことにつながらない場合がある。

これに対する方策としては、ソースコードという具体例を題材とした学習を始める前に、あらかじめ抽象的な概念のイメージを形成しておくことが有効である。例えば、プログラミング言語の学習を始める前に、制御構造を表す図を用いて処理の流れに対する明確なイメージを形成しておくことは、プログラミングの理解度向上に効果があったと、長谷川、山住により報告されている<sup>13) 14)</sup>。また、近年、小学校教育へも広く導入の進んでいる、ビジュアル型プログラミング言語を用いたプログラミング教育も、概念の獲得に大きく寄与するものと考えられる。本学では、ビジュアル型プログラミング言語を用いたプログラミング教育科目も開講しており、プログラミング初学者に対しては本科目受講前の履修を推奨している。また、本科目では、毎回の授業冒頭で、動画視聴による概念イメージ形成のための学習を実施している。

### 3.2. 既知のプログラミング言語を持たない初学者の学習方略

プログラミング言語の文法学習は、初学者にとって、自然言語である第二言語学習の場合の文法学習とは全く状況が異なる。第二言語の文法学習では、獲得済みの第一言語の文法との比較が可能であるため、具体的な表現に接した経験がわずかであっても、文法学習が漠然とであれ文法について何かしらの予備知識を得る機会となり、第二言語による具体的表現に接する際の理解促進につながるだろう。しかし、プログラミング言語の学習経験を持たない初学者の場合、比較対象となりうるプログラミング特有の概念を表現可能なプログラミング言語の知識を、一切持ち合わせていない。そのため、自然言語の文法と比べ圧倒的に単純であることを考慮しても、それを理解するのは第二言語学習の場合より困難となり得る。

実際、従来の学習順序を採用していた過去の学期には、プログラミング演習に先立つ文法学習に対し、「文法を説明されてもピンとこない」という感想が寄せられていた。この感想に代表されるように、プログラムのコード作成や実行結果の確認などに取り組む前の文法学習は、既知のプログラミング言語を持たない学習者にとっては理解し難く、効果が低

いものと考えられる。初学者にとって、プログラムのコードは、括弧をはじめとした種々の記号が使用されているなど、日常の文章では見慣れない形式も相まって、意味や構造を持った記述として認識するのが難しいものである。そのような状況では、より抽象的な文法学習は効果が上がらないのも当然である。

このような既知のプログラミング言語を持たない状況、そして、学習している言語で表現される概念自体も十分には把握できていない状況を鑑みるに、初学者におけるプログラミング言語の習得過程は、第二言語の習得過程よりも、むしろ幼児が母語である第一言語を習得する過程と類似しているというべきであろう。第一言語は、その言語にさらされる状況の中で、表現が示す概念と共に獲得されていくものである。したがって、より第一言語習得過程に近い、文法などの知識学習よりもインプットの意味理解に重点をおく学習法は、第二言語習得の場合以上にプログラミング言語習得には有効であろうと考えられる。つまり、プログラミング言語学習においては、文法学習よりも写経型学習を先行させる学習方略が合理的といえよう。

### 3.3. プログラム実行までの困難軽減のための取り組み

自然言語を学習している場合、学習者が書く・話すといったその言語を用いたアウトプットによって内容を伝えたい相手は人間である。単語の綴りや発音、あるいは構文や語句の使い方などに多少の間違いがあっても、学習者は相手におおよその内容を伝えることができる。つまり、たとえ言語が十分に習得されていない状態であっても、例文の模倣ではなく学習者が自ら考えた多少の間違いを含む表現であっても、アウトプットを試みれば、それがただちに一定程度の成功をおさめる。従って、精読と語彙や文法などの知識学習を行った後にアウトプット練習を繰り返すということによっても、語彙や文法の定着をはかることは可能であろう。

一方、プログラミング言語の場合は、内容を伝えたい相手はコンピュータである。そのため、コンパイルや実行などの操作の誤りやコード上のわずかなスペルミスによっても、コードの内容はコンピュータに伝わらなくなる。つまり、エラーが出てプログラムが実行できない、あるいは実行はできても正しく動作しないといった状態になり、往々にして、模倣によりプログラムを入力し実行するという、写経型学習段階における試みさえも失敗してしまう。さらに、プログラミング言語の理解が不十分なまま、例示されたプログラムを改変する段階、獲得した知識を活用してプログラムを作成する段階へと進もうとすれば、構文や命令の誤用などでより多くのエラーに遭遇することになる。学習者にとって、デバッグ作業は非常に負荷の高い作業であるため、たとえ写経型学習の段階であっても、プログラムの実行結果が容易に得られるとは限らない。

こうしたプログラム実行までの困難を軽減し、学習効率を向上させるための取り組みとしては、教師の手助けや学習者同士の協力が期待できる対面での授業時間を、躓きの発生しやすい後半段階の学習へと振り向けるために、前半段階の学習を事前学習として課して



おく反転授業の取り組みが挙げられる。動画教材により文法学習を事前学習として課す手法（高井，水谷<sup>15)</sup>）、文法学習だけでなく例題プログラムの模倣による写経型学習までを事前学習とする試み（喜多，岡本<sup>3)</sup>）、動画教材による学習と確認問題への取り組みまでを事前学習としておき授業にはグループによる協調学習を導入する取り組み（林ら<sup>16) 17)</sup>）などが報告されている。

いずれも演習の際の躓きを早期に解決し、多くの演習課題に取り組む機会をつくる効果があるものと考えられる。しかし、学習者のタイピングスキルなどによっても進捗や理解度に大きな差が生じるという、プログラミング学習の特性による課題は依然として残る。そこで、学習者自身のペースで、周囲の助けを待たずに自立的に進められる学習方法として、躓き要因の比較的少ない写経型学習を質、量ともに充実させていくことが肝要となる。特に、対面の授業時間を持たない完全 e ラーニングで授業を進める本学においては、学生の演習状況を確認できる演習環境を利用し、学生の質問には複数の教員とティーチングアシスタント（以下「TA」と呼ぶ）により迅速に対応する体制を整えてはいるものの、さらに写経型学習を通じて最大限の学びを引き出すことが求められている。そのための方策として、一般に写経型学習と呼ばれる模倣による学習の段階だけでなく、改変による学習の段階までを、コードと実行結果を明示した写経型の演習により進めている。

### 3.4. 写経型学習が陥りやすい学習目的忘失

写経型学習の本来の目的は、ソースコードを読み、模倣してコード入力し、実行結果を確認する過程を通して、プログラムの動作がコードの記述を反映したものであることを認識し、写経対象のプログラムの持つ基本構造と個々の命令や構文の意味を理解すること、さらには、コードという具体例から文法を抽出し習得することである。しかし、写経型学習は、提示されたコードの丸写しによって演習を進めていくため、盲目的な作業として漫然と進めることも可能である。また、この単なる模倣の段階でさえ、実行までには一文字でもタイプミスがあるとプログラムが実行できなくなるなどの困難が存在するため、プログラムの実行自体が目的化してしまい、本来の学習目的を見失ってしまうことも多い。すなわち、タイプミスなどによるコンパイルエラーを無くし実行にこぎつけただけで満足してしまい、肝心のコードに着目することができず、プログラムの実行結果がコードの記述を反映したものと認識してコードの意味を理解するに至らなくなるのである。

こうした写経型学習が陥りやすい演習の形骸化を防ぎ、コードの意味理解という本来の目的に対する意識を維持するためには、躓き予防策を講じてプログラム実行までの困難を軽減するとともに、コードの内容に着目させる工夫が重要となる。これまでには岡本・喜多により、認知的負荷軽減のためプログラムの動作を視覚的に顕在化させるような写経用教材の開発、実行可能な完全なプログラムと実行例の提示、学習者がエラーに際し対処法を逐次参照できるような教材の準備、エラーの明示的経験、記号の識別力強化やコミュニケーション円滑化のためのコード音読・訳読への配慮などの取り組みが報告されている<sup>2) 3)</sup>。

本科目では、さらなる工夫として次のような対策を講じている。

### 1) 模倣から改変までのスモールステップの積み重ねによる写経型演習

本科目では、基本的なコードの模倣だけでなく、改変の過程までをコードを提示しての写経型の演習によって扱っている（以下、模倣だけでなく改変までの過程を含めて「写経型演習」と呼ぶ）。演習は、ごく短いプログラムのコード入力と実行結果の確認から始め、さらに、そのコードに対し少量ずつ改変や追記を行っては、実行結果の変化を確認するという演習を、手本を参照しながらの写経により繰り返す。実行結果の確認にあたる際の、その度毎のコード入力量を減らすことにより、コード中の着目すべき箇所を明確化して意味理解を促進するとともに、タイプミスによるエラー発生時に原因箇所の特定を容易にし、デバッグの労力軽減をはかっている。これらの工夫により、タイピングの遅い学習者であっても、できるだけ多くのコードと実行結果の対応を観察する経験を積めるよう、そしてコードの意味理解に至ることができるよう配慮している。

### 2) 写経型演習先行・文法学習後行という学習順序の採用

文法学習には、演習前に取り組むのではなく、テーマとなる概念と最低限の知識のみを与えられた状態で写経型演習を行った後に取り組むこととした。この「写経型演習先行・文法学習後行」という学習順を採用したのは、前述のような、既知のプログラミング言語を持たない初学者にとって演習前の文法学習はあまり効果が見込めない、という消極的な理由からだけではない。この学習順の採用は、演習に際しコードの意味理解に対する目的意識を維持しようという積極的な理由からでもある。コードの意味をあらかじめ詳しく解説してしまうことなく、実行結果からコードの意味を類推する余地を残しておくことにより、演習の際に謎解き感覚でコードの意味を考えさせ、演習目的を見失わせないようにすることを狙っている。つまり、演習の形骸化防止効果を期待した取り組みでもある。

### 3) 課題でのエラー発生からデバッグまでの明示的体験

写経型演習においても、エラー発生に伴うデバッグ作業は避けては通れない。慣れない初学者にとって、原因の究明とプログラムの修正は難しい作業であり躓き要因となる。コンパイルエラーが発生してプログラムが実行できない場合は、エラーメッセージが原因特定の助けにはなるだろうが、デバッグが負荷の高い作業であることに変わりはない。また、実行できるが正しく動作しない場合は、動作が正しくないことに気付かない場合や、気付いたとしても、コードに誤りがあるという発想にさえ至らない場合も多い。エラーの発生箇所は学生によって異なり一定しないため、原因究明やバグ修正についての一斉指導には限界があり個別指導が必要となる。しかし、どれだけ学生からの質問に対応し個別指導する体制を整えたとしても、学生自身が教員や TA に質問するなどの問題の解決に向けた行動を取らねばならず、個別指導に至るまでには一定のハードルが存在する。つまり、躓いた時に解決のための質問をすることなく、投げ出してしまいう学生も多い。そこで、本科目

## プログラミング教育における「写経型演習先行-文法学習後行方式」の実践報告

では、バグを含むコードを示し、そのバグの症状や修正方法についての選択式問題に答えながら、演習環境で実際にデバッグを行ってプログラムを完成させる課題を課し、エラーへの対応力強化をはかっている。

### 4. 「写経型演習先行・文法学習後行方式」の実践と評価

本科目において、2022 年度春学期に実施した授業の具体的な進め方と、学期末に行ったアンケート結果について以下に報告する。

#### 4.1. 本科目の構成

本科目は全 15 回で構成されており、学習内容は表 1 の通りである。本学はインターネット上で授業を提供している通信制大学であるため、学生は、動画コンテンツ視聴、プログラミング演習、テキスト主体のコンテンツ学習、小テスト、レポート課題、期末試験など全ての学習活動に e ラーニングにより取り組み、疑問点についてはメール等で教員や TA に質問する。演習には、goorm という Docker コンテナで提供されるクラウド開発環境を利用しており、学生からの問い合わせに応じて、教員・TA が各学生の演習環境内に入って演習状況を確認できるような体制を整えている。

各回は、表 2 に示すように、第 1 章～第 3 章として提供される 3 種の授業コンテンツ（解説動画、演習用テキスト、文法学習用テキスト）と課題（小テストあるいは Web レポート）で構成されている。課題としては、第 6 回、第 9 回、第 13 回では、5～6 個のプログラムを、コードの穴埋めやバグ修正方法についての選択式問題に答えながら、演習環境で完成させる Web レポート課題を課し、その他の回では、比較的負荷の少ない全 8 問の小テストを課している。学生は、回ごとに決められている受講期限までに、これらの学習活動に取り組むことで各回の出席が認定される。そして、学期末には選択式問題からなる期末試験も課される。この他、第 14 回には、決められた基本仕様に各自で機能を追加するなどの工夫を加えてプログラムを完成させる、任意提出の課題も用意している。

表1 本科目の構成 (全15回)

回	学習内容	課題
第1回	ビルド・実行の方法	小テスト
第2回	コード入力・標準出力	小テスト
第3回	データ型・変数・標準入力	小テスト
第4回	算術演算子・代入演算子	小テスト
第5回	条件演算子・if文・if～else文	小テスト
第6回	論理演算子・if文・switch文	Webレポート
第7回	while文	小テスト
第8回	for文	小テスト
第9回	配列	Webレポート
第10回	関数	小テスト
第11回	ポインタ	小テスト
第12回	文字列	小テスト
第13回	構造体	Webレポート
第14回	ファイル入出力	小テスト 任意提出課題
第15回	複数ファイルでの開発	小テスト
期末試験	-----	期末試験

表2 各回の授業コンテンツの構成

第1章	動画視聴 (10～15分) ・ 前回の復習 ・ 新出概念の説明とキーワードの紹介
第2章	プログラミング演習 ・ テキストを参照しながらの写経型学習 ・ 連載形式で示される指針に従い 各自のプログラム改良に取り組む応用学習
第3章	テキスト学習 ・ 文法のまとめ ・ チェック問題
小テスト または Webレポート	小テスト (全8問) ・ 文法確認と演習実施確認 Webレポート (全25問程度) ・ 第6, 9, 13回で実施 ・ 穴埋め、バグ修正などについての選択式問題

## 4.2. 各回の構成

各回の第1章は、前回の復習と、その回で新たに登場する概念や、演習に必要な最低限の知識についての説明を行った 10～15 分程度の動画である。前回の復習を行っているのは、これから新たに学習する内容をこれまでに学習した文法事項と組み合わせて利用するという意識を持ってもらうためである。また、新たな概念については、例えば変数の宣言なら「花見の場所取り」を想起させるイラストを示すなど、図やイラストによるイメージ形成に主眼を置き、文法の詳細には立ち入らずに、キーワードとなる予約語や関数名を動画内で紹介することにより読み方を示す程度にとどめている。

続く第2章は、演習用テキストとなっている。学生はこのテキストを参照しながら、提示されているコードを模倣しタイピングして入力する「写経」を行い、まずごく短いコードのプログラムを作成する。そして、テキストの指示に従い、このプログラムのコードに少量の変更あるいは追記を加えては、実行結果の変化を確認する。こうしたスモールステップの演習を繰り返し、このプログラムを次第に長いプログラムへと発展させていく。演習用テキストには、変更や追記を行う部分だけでなく、プログラム全体のコードと実行結果を示しており、特に変更・追記部分のコードは色分けにより明示し、紛らわしい記号などとともに注釈を付記している。通常1回の授業で完成させるプログラムは、変更や追記を繰り返して作成するプログラム2個程度とし、新たなプログラムの作成作業に移る際には、再び短いプログラムの作成から始めるか、あるいはコード入力の手間を省けるよう、コピー可能な形式で提供されたプログラムから始める。

また、写経型演習によって得た知識を活用し、自らコードを考える段階へと学習を進めるための応用演習も用意している。具体的には、テーマとなっているプログラム（「データ管理プログラム」と「数当てゲーム」）に、複数回にわたり、新たに学習した知識を踏まえて、機能を追加するなどの改良を施していき、次第に複雑で大きなプログラムを構築していく演習である。テキストには、第3回から第14回までの連載形式で、改良の方針とそれを実現するためのヒントを載せ、次の回でコーディング例を示している。

第3章は、文法のまとめと文法の理解を確認するための解答付きチェック問題からなる、テキスト主体のコンテンツである。第2章に従って演習を行った後に、この第3章に取り組むことで、演習で学んだことを文法知識として整理し定着をはかっている。

これら第1章から第3章までの授業コンテンツによる学習の後に、通常（Web レポートを課している回以外）は、必須課題として小テストを課す。小テストは、文法理解を確認するための選択式問題7問の他に、第2章の演習用テキストで伏せておいた実行結果を答える記述式問題1問を加えて計8問とし、毎回の演習を実施したかどうかの確認としている。

第6回、第9回、第13回では、小テストは課さずに Web レポートを課す。これは、数多くのプログラムを読む経験をさせるとともに、エラー体験によりバグへの対応力を強化することを目的とした課題である。学生は、提示されている5～6個の短い不完全なプログラム（空欄やバグを含む）を、設問に回答しながら、実際に演習環境を利用して完成さ

せる。設問は、空欄に当てはまるコード、不具合症状、バグ修正方法などを選択肢から選ばせる形式を主体とし、難しくなりすぎずに多くのプログラムのデバッグ作業を経験させる課題としている。提示する不完全なプログラムはコピー可能な状態で提供し、コード入力負担を軽減する工夫をしている。また、コンパイルエラーの発生するバグだけでなく、実行は出来るが意図と違う動作をするバグも経験させ、写経に夢中になるあまり、プログラムが実行できただけで満足しコードの意味を考えないという事態に陥らぬよう配慮している。

#### 4.3. アンケート結果

以上のような2022年度春学期に実施した本科目の授業方式について、学期末にアンケートを実施したところ、受講した学生408名のうち117名から有効な回答が寄せられた。

##### a. プログラミング学習経験について

アンケートに回答した学生の内訳は、プログラミング学習経験についての質問への回答によると、「テキスト型プログラミング言語の学習経験あり」37名、「主にビジュアル型プログラミング言語の学習経験のみ」35名、「プログラミング学習経験なし」45名であり、図1のような構成であった。本学にもビジュアル型プログラミング言語を用いたプログラミング入門科目が用意されているが、全くプログラミング学習経験のない者も多く、プログラミング特有の概念についてイメージが形成されていない者も、相当数存在するであろうことは想像に難くない。全体としても、テキスト型プログラミング言語の学習経験の無い者が7割近くにのぼっていて、テキスト型プログラミング言語に関しては初学者が大半を占める状況となっている。

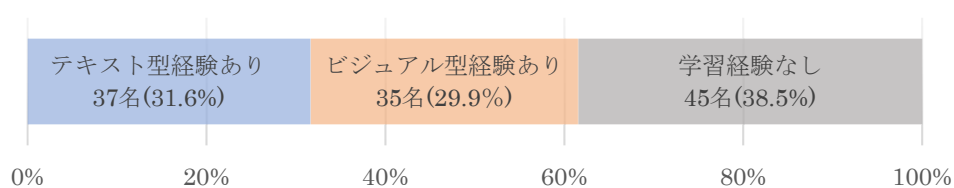


図1 プログラミング学習経験

##### b. 学習方式について

従来の「文法学習後に課題プログラムのコードを自分で考える演習を行う方式」と比べ、本科目で採用していた「提示されたコードを入力し実行結果を確認することを中心とした演習を行い、演習後に振り返りとして文法学習を行う方式」は効果的だったと思うかを尋ねた質問に対する回答を、学習経験別に集計した結果を表3に示す。それぞれの割合は図2の通りであった。

テキスト型プログラミング言語の学習経験を持つ者は、既知のプログラミング言語の文

## プログラミング教育における「写経型演習先行-文法学習後行方式」の実践報告

法知識を有しているため、文法学習を先に行った方が効率的と考える者も多いのではないかと、とも予想されるところであった。しかし、実際には、テキスト型プログラミング言語の学習経験を持つ者についても「従来方式の方が良い」と答えた者は8.1%に過ぎず、「非常に効果あり」と「ある程度 効果あり」を合わせれば、実に91.9%という多くの者が、文法学習に先立って写経型演習を行う学習方式を支持した（図2）。

テキスト型プログラミング言語の学習経験を持たない者については、当然、従来方式による学習経験も持っていないため、二つの方式を比較することは不可能である。しかし、「従来方式による学習経験が無い方も、どの程度、効果的だったかをお答えください」と付記したところ、「どちらとも言えない」を選んだ回答者は14%前後にとどまり、ビジュアル型プログラミング言語の学習経験の有無にかかわらず、8割以上が本科目の学習方式について効果を認める回答を選択した（図2）。

表1 学習方式について

人数（名）	非常に 効果あり	ある程度 効果あり	従来方式 の方が良い	どちらとも 言えない	計
テキスト型 学習経験あり	13	21	3	0	37
ビジュアル型 学習経験あり	12	17	1	5	35
学習経験なし	12	26	1	6	45
計	37	64	5	11	117

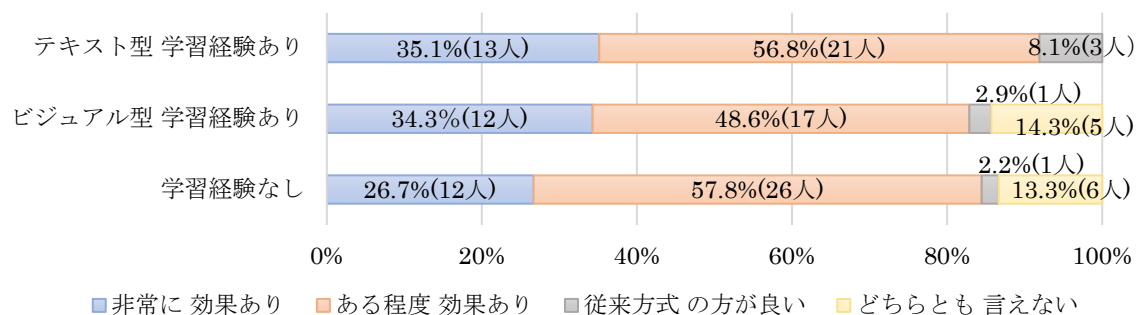


図2 学習方式について

## 5. まとめと今後の課題

テキスト型プログラミング言語の学習経験を持つ者については、従来の「文法学習後に課題プログラムのコードを自分で考える演習を行う方式」による学習を経験した者も多いと考えられる。したがって、こうしたテキスト型プログラミング言語の学習経験を持つ学習者の中で本科目の学習方式を支持する意見が多かったということは、テキスト型プログラミング言語の学習経験が無い者たちの意見にもまして、信頼度の高い結果だと考えられる。このように「写経型演習先行・文法学習後行方式」が、筆者の予想以上に支持を集める結果となったことから、文法などの知識についての学習よりも言語に触れる量を重視した学習は、自然言語習得だけでなくプログラミング言語習得にも有効だということが裏付けられたと考えている。

写経型演習が形骸化した作業に陥りやすいという懸念については、コードのごく一部の改変による実行結果の変化を観察させる、文法学習を後行させるなど、コードの意味理解に対する目的意識を維持させるような対策を講じたことにより、一定程度、払拭できたと考えている。一方、Web レポートとして課していたデバッグ体験課題については、演習を最後まで行わずに途中でやめてしまう学生を減らし、エラー発生に際しても投げ出さずにデバッグに取り組む姿勢を養うことに寄与できたと考えているが、エラーの修正方法を選択式問題で答えさせていることから、コードが誤っていたことやバグの修正方法については理解したものの、どのようにして誤ったコードによりおかしい動作が生じたのか、という新たな疑問をもつようになった学生も少なくなかった。今後の授業改善の取り組みとしては、このような、学生がコードについて深く考え疑問を持つという絶好のタイミングをとらえ、より充実した詳しい説明を与えていくことが、理解度を高めることにつながると期待している。

## 参考文献

- 1) 喜多 一, 岡本 雅子, 藤岡 健史, 吉川 直人, 「写経型学習による C プログラミングワークブック」, 共立出版, 2012.
- 2) 岡本 雅子, 喜多 一, 「プログラミングの「写経型学習」における初学者のつまずきの類型化とその考察」, 滋賀大学教育学部附属教育実践総合センター紀要, 22 巻, pp. 49-53, 2014.
- 3) 喜多 一, 岡本 雅子, 「写経型プログラミング学習と反転授業」 第 60 回システム制御情報学会研究発表講演論文集, 2016.
- 4) R. Day, J. Bamford (著), 榊井 幹生(監訳), 「多読で学ぶ英語 楽しいリーディングへの招待」, 松柏社, 2006. (原著 “Extensive Reading in the Second Language Classroom”, Cambridge:



## プログラミング教育における「写経型演習先行-文法学習後行方式」の実践報告

Cambridge University Press, 1998. )

- 5) 酒井 邦秀, 「快読 100 万語! ペーパーバックへの道」, ちくま学芸文庫, 2002.
- 6) 古川 昭夫, 伊藤 晶子, 酒井 邦秀, 「100 万語多読入門—辞書を捨てれば英語が読める」 コスモピア, 2005.
- 7) 高瀬 敦子, 「英語多読・多聴指導マニュアル」, 大修館書店, 2010.
- 8) S. D. Krashen, “Principles and Practice in Second Language Acquisition”, 1982.
- 9) S. D. Krashen (著), 長倉 美恵子, 塚原 博, 黒沢 浩 (共訳), 「読書はパワー」(原著 “The Power of Reading : Insights from the Research”, 1993) , 金の星社, 1996, p. 107.
- 10) 西澤 一, 吉岡 貴芳, 伊藤 和晃, 「英文多読による工学系学生の英語運用能力改善」 電気学会論文誌 A (基礎・材料・共通部門誌) , 126 巻, 7 号, pp. 556-562, 2006.
- 11) 熊田 道子, 「Extensive Reading (多読) による読み手の変化」 日本語教育方法研究会誌, 22 巻, 3 号, pp. 90-91, 2016.
- 12) 林 幸代, 丸尾 加奈子, 川瀬 義清, 長 加奈子, 「多読学習が英語読解に与える影響」 LET Kyushu-Okinawa BULLETIN, 20 巻, pp. 27-38, 2020.
- 13) 長谷川 聡, 山住 富也, 「プログラミング教育と学習者のイメージ形成」, 名古屋文理短期大学紀要, 22 巻, pp. 9-14, 1997.
- 14) 長谷川 聡, 山住 富也, 「プログラミング教育と学習者のイメージ形成 (その 2)」 名古屋文理短期大学紀要, 23 巻, pp. 9-14, 1998.
- 15) 水谷 晃三, 高井 久美子, 「プログラミング初学者を対象にした動画教材による反転授業の実践と評価」, 研究報告コンピュータと教育, 132 巻, 34 号, pp. 1-8, 2015.
- 16) 林 康弘, 深町 賢一, 小松川 浩, 「e ラーニング利用による反転授業を取り入れたプログラミング教育の実践」, ICT 活用教育方法研究, 16 巻, 1 号, pp. 19-23, 2013.
- 17) 林 康弘, 深町 賢一, 小松川 浩, 「プログラミング教育における反転授業の実践と評価」, 教育システム情報学会 第 40 回全国大会論文集, pp. 97-98, 2015.



# WBT 教材の効率的な開発法に関する研究

鈴木 秀男<sup>1</sup>, 長島 雄平<sup>2</sup>, 田中 健太郎<sup>2</sup>

キーワード: WBT (Web Based Training)、教材開発、eXeLearning、MathJax、CortexJS、KaTeX

## 要 旨

本学にて使用されている WBT 教材の開発ツールである eXe は、2017 年に最終版をリリースして以来メンテナンスが行われていない。リリース以降すでに 5 年が経過し、eXe のバージョンも 1.04 から 2.7 へと進化を遂げており、新機能の追加やバグフィックスも行われている。本論文では、最新版の eXe の使用方法から本学での利用を前提とした検証を行った。その結果、現状の eXe からの必要機能の移植も可能であり、本学での WBT 教材の開発に十分利用できることが確認された。

## 1. はじめに

サイバー大学（以下、「本学」）の Cloud Campus（以下、CC）上での WBT (Web Based Training) 教材の開発には、フリーソフトウェアである eXe<sup>1)</sup>を利用している。しかし、本学で利用している eXe のバージョンは 1.04 と古いのが現状である（以下、「現状の eXe」）。また、教育メディア開発部より提供されている eXe ソフトウェアの種類も、数学系科目で使用する数式の表現に対応している数式表現版、プログラム開発に向いている通常版の二つがあり、同時にインストールすることができない。同時にインストールできないことから、両バージョンともに、インストール不要版も提供されているが、管理者権限がないと利用できないという制約がある。以上のように、現状の eXe には、多くの問題点があり、決して使いやすいものとなっていない。

本研究では、現状の eXe の問題点を、これまでの利用手順を追いながらまとめることから始める。はじめに、教職員へのアンケートを行い、その結果をもとに問題点の抽出を試みる。その後、現状のバージョンでの解決策を検討し、今後の eXe 利用の方策について検討する。また、最新版の eXe（2022 年 12 月時点では 2.7<sup>1)</sup>）では、数式も含めて統一的に処理できることから、CC で利用する教材開発用に利用できるかどうかを検討する。この際、現状バージョンからの開発環境の移行が可能であるかどうかを併せて検討し、最新版 eXe の利用可能性について検討する。

---

<sup>1</sup> サイバー大学 IT 総合学部・教授

<sup>2</sup> サイバー大学 教育メディア開発部

## 2. 現状の eXe に関するアンケート結果

本学では、これまで WBT 教材を開発するために eXe を利用してきた。はじめに、現状の eXe を使った教材開発の手順についてまとめておく。教材の開発から CC 上で使用できる教材の形式に変換するまでの手順は以下の通りである。

1. eXe を使ってテキストベースの教材を作成
2. 作成した教材を eXe 上で一旦 ZIP 形式として出力する
3. ローカル側の Windows マシンで ZIP 形式のファイルを解凍する
4. TextSS.net で本学用のレイアウトに変換をする
5. 変換されたフォルダを再度 ZIP 形式として出力する
6. 出力された ZIP 形式のファイルを CC 上へアップする

本研究を進めるにあたり、WBT 形式の教材を担当科目内で扱っている教職員を対象に、現状の eXe の操作性についてアンケートを行った。アンケートの項目は、現状の eXe の使い勝手に関するものであり、率直な意見を収集するために記述式を採用している。表 1 はアンケートの設問一覧であり、調査は 2022 年 6 月に行い対象者 15 名中 14 名から回答があった。

表 1 アンケートの設問一覧

<p>① 教材の作成や改修などの際に eXe を使ったことがありますか？ (ここで「いいえ」の方は、⑧のみご回答ください。)</p> <p>② 満足している操作性、機能などをお書きください。 (エディタの機能や編集の際の機能など)</p> <p>③ 満足されていない操作性、機能などをお書きください。 (エディタの機能や編集の際の機能など)</p> <p>④ 不要であると思われる機能があればお書きください。</p> <p>⑤ 追加が必要であると思われる機能があればお書きください。</p> <p>⑥ 全体的な評価として、現状の eXe の操作性について満足していますか？</p> <p>⑦ 現在は、eXe で作成したのち、ZIP ファイル作成後の変換作業 (TextSS) を経て、最終版の納品となります。こちらの変換作業を行ったことはありますか？</p> <p>⑧ その他、eXe について、ご意見があればご記入ください。</p>
---

実際に教材を開発した経験のある教職員は、設問①より 11 名であった。以下、設問①に「はい」と回答した 11 名の回答について考察を行う。設問⑥では、現状の eXe の操作性について、45%が「不満」または「やや不満」と回答している (図 1)。これは、上記の手順を見ても分かるように eXe だけで完結できるものではなく、TextSS.net などの外部処理が必要となるため、ZIP ファイルの解凍と再作成の手間がかかることから、その煩雑さがマイナスの評価となっているものと考えられる。

⑥現状のeXeの操作性について満足していますか？

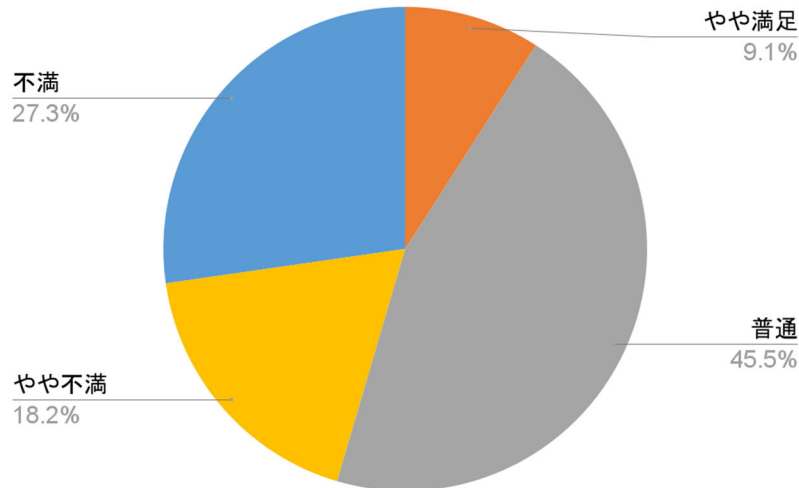


図 1 eXe の操作性に関するアンケート結果

「不満」や「やや不満」の要因は、記述式での回答から、主に三つの項目に分類できる（図 2）。一つ目は、eXe の編集画面の使い勝手が悪いことである。eXe のバージョンが古くインターフェースが直感的ではないことと、ワープロやプレゼンテーションソフトのように完成形を視認しながら直接の編集ができないことが挙げられている。もともと、eXe は HTML を生成するエディタであるため、仕様上仕方のない面もある。二つ目は、変換作業が面倒であり変換後の完成形の確認が CC にアップロードするまで分からないことである。手順を見ても分かるように、手間であると同時に完成形が HTML をまとめた ZIP のため、CC にアップしないと確認ができない。そのため確認後に再度修正をする場合は、eXe での編集操作に戻りあらためて修正するという手間がかかるからだと思われる。本アンケートでは、11 名中 4 名がレイアウトの変換作業まで対応を行っており（アンケートの設問⑦）、その内 3 名が不満な点として回答している。三つ目は、eXe に備わっている機能に満足できないことである。これは eXe のバージョンが古く、もともと十分な機能が備わっていないことも要因である。最新版では多くの有用な機能が実装されている。

このように現状の eXe では、バージョンの古さにも起因して、十分な機能が実装されていない。次章では、最新版の eXe の機能について紹介する。

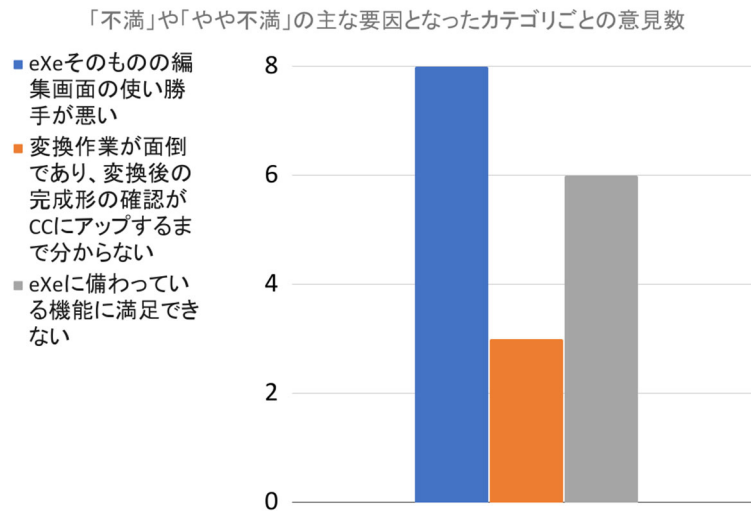


図 2 「不満」や「やや不満」の主な要因

### 3. 最新版 eXe の機能

#### 3.1. 動作環境

最新版の eXe は、OS 環境として Windows だけでなく Linux や Mac にも対応している。OS 環境ごとのダウンロードからインストールまでを簡単に記載しておく。

##### 3.1.1. Linux 版

Linux 環境では、ディストリビューションごとにダウンロードファイルが分けられている。例えば、Debian/Ubuntu 系のディストリビューションであれば、deb 形式のファイルが用意されているため、それをダウンロードしてインストールを行うことができる。Fedora/RedHat 系のディストリビューションについても、rpm 形式のファイルが用意されているため、簡単にインストールが可能である。また、Debian/Ubuntu 系及び Fedora/RedHat 系ともに、Snap パッケージに対応しており、snap コマンドで簡単にインストールが可能である。本研究では、eXe の動作や機能に改変を加えることから、ソースファイルが提供されている Portable 版を使用した。Portable 版は tar.gz 形式で提供されており、適当なディレクトリ配下に展開をすることでターミナルのコマンドラインから eXe を直接起動できる。

ダウンロードは、eXeLearning のサイトのトップページから、「DOWNLOADS」をクリックしダウンロードのページへ移行する。GNU/Linux の項目の一番下にある「Portable (Linux)」をクリックすればダウンロードが開始される。ダウンロードされるファイル名は、「portable-intef-exe-2.7-linux.tar.gz」で、サイズは約 90MB ある。適当なディレクト

リへ保存して解凍すれば、「exelearning27」というディレクトリが作成される。このディレクトリ内にある「exelearning.sh」を実行すれば eXe が起動する。なお、実行する際に、いくつかのパッケージが不足しているとのメッセージが表示される場合は、メッセージに従い不足しているパッケージをインストールする必要がある。

### 3.1.2. Windows 版

Windows 版では、現在本学で使用している eXe と同様に、インストール版とインストール不要版が用意されている。インストール版は、ダウンロードしたファイルをダブルクリックすることでインストールが始まり、インストールが終われば「exelearning」というアプリケーションが作成されるので、これをダブルクリックすることで eXe が起動する。インストール不要版は、ダウンロードしたファイルをダブルクリックすれば eXe が起動する。なお、現状の eXe がインストールされている状態で、最新版の eXe をインストールすると、一部の設定ファイルが上書きされるため、現状の eXe を起動することができなくなるので注意が必要である。

Windows 版においてもソースファイルの改変を行うため、本研究では ZIP 形式で提供されている Portable 版を使用する。ZIP 形式のファイルを適当なフォルダに展開することで、すべてのソースファイルが同じフォルダ内に格納されることになり、「exelearning」というアプリケーションをダブルクリックすることで eXe が起動する。

ダウンロードは、Linux 版と同様に、ダウンロードページから Microsoft Windows の項目の一番下にある「Portable (Windows)」をクリックすればダウンロードが開始される。ダウンロードされるファイル名は、「portable-INTEF-exe-2.7-win.zip」で、サイズは約 85MB である。適当なフォルダへ保存して解凍すれば、「portable-INTEF-exe-2.7-win」というフォルダが作成される。このフォルダ内にある「exelearning」をダブルクリックすれば eXe が起動する。

### 3.1.3. Mac 版

Mac 版に関しては、Intel 版と ARM 版が用意されているため、インストールするコンピュータの CPU スペックに合わせたファイルをダウンロードする必要がある。インストールは、ダブルクリックで完了し、インストール完了後に eXe のアイコンをクリックすることで eXe が起動する。

## 3.2. 最新版 eXe に搭載されている主な機能

最新版 eXe での WBT 教材の作成方法は、現状の eXe と同様であるが、搭載されている機能において各種の拡張機能が追加されている。本節では、WBT 教材の作成において、役に立つと思われる機能を説明する。なお、本節では数式以外の機能について説明し、次節で数式について詳しく説明する。

### 3.2.1. 「Insert HTML code as text」機能

「Insert HTML code as text」機能は、HTMLのタグやプログラミング言語のキーワードを色付けして分かりやすく表示する機能である。現状のeXeにも搭載されているが、最新版eXeでは機能的にも拡張され、より多くの構文をサポートできるようになっている。この機能を使うには、テキスト編集画面のメニューから、赤丸で囲まれたアイコンをクリックする(図3)。

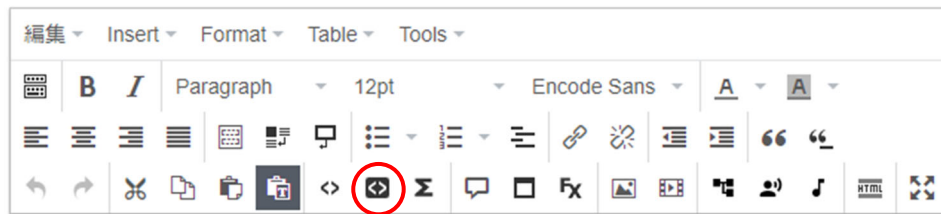


図3 テキスト編集画面のメニュー (Insert HTML code as text)

アイコンをクリックすると、ポップアップウィンドウが表示され、構文表示を可能にする場合は、「Include Styles」にチェックを入れる。チェックを入れることで、「Highlight syntax」が表示されるので、構文を色分けする場合には、こちらにもチェックを入れる(図4)。チェックを入れると、「Programming language」が表示され、プルダウンで、言語を指定できる。図4のように、C言語、Java言語、LaTeX、PHPなど、様々な言語に対応している。

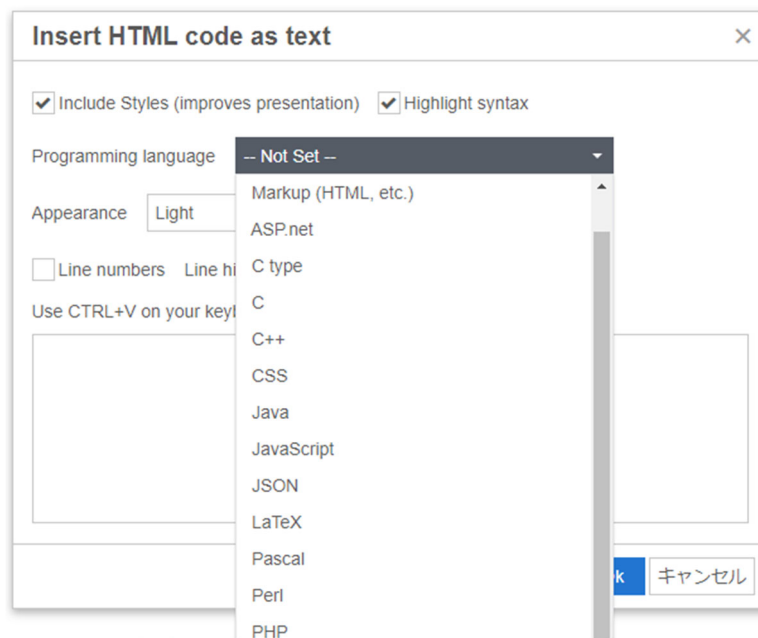


図4 Insert HTML code as text 画面



図4のプルダウンから、C言語を選択し、ソースコードを入力またはコピー&ペーストし、OKボタンを押すことで構文に色が付いた状態で表示される(図5)。なお、必要に応じて、「Line numbers」の機能や、「Line highlight」の機能を使用することができる。図5では、この機能を使い、行番号の表示および途中の5行目から8行目までをハイライトしている。

```

1  #include <stdio.h>
2  int main(){
3      int i,s;
4      printf("1 から10 までの和は、");
5      s=0;
6      for(i=1; i<=10; i=i+1){
7          s=s+i;
8      }
9      printf("%d です。 \n",s);
10 }
```

図5 C言語のプログラムの表示例

### 3.2.2. 「Paste HTML fragment(embed code)」機能

「Paste HTML fragment(embed code)」機能は、HTMLのコードをeXeに直接埋め込むことで、HTMLの動作をeXeの画面上で確認できるといったもので、とてもユニークな機能である。WBTコンテンツの中で、実際の動作の流れを示す際に、通常は画像を使い、画面の遷移や動作の流れを説明する。本機能はHTMLのみであるが、実際のコードを入力することで、HTMLの動作をeXe上で確認できるものである。

この機能を使うには、テキスト編集画面のメニューから、赤丸で囲まれたアイコンをクリックする(図6)。クリックすると、ポップアップウィンドウが表示されるので、HTMLのコードを入力またはコピー&ペーストすることで、入力されたコードがeXeに埋め込まれる。埋め込まれたHTMLのコードは、eXe上で動作を確認することができる。例えば、図7は、HTMLを使ったフォームを表示してeXe上から入力を行っている状態である。利用者は、eXeの教材を使いつつ、実際の入力を体験しながら学習を進めることができる。埋め込まれたHTMLを修正する場合には、HTMLエディタを使うことで可能となっている。

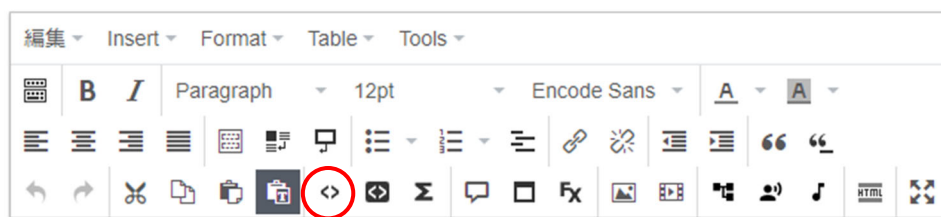


図6 テキスト編集画面のメニュー (Paste HTML fragment(embed code))



図 7 埋め込まれた HTML に eXe から直接入力

### 3.2.3. 「タイトル画像」機能と「テキスト」機能

「タイトル画像」機能は、タイトルの前に画像を挿入できる機能であり、現状の eXe には実装されていない機能である。通常のタイトルは文字だけで表記することが多いが、文字のみ、画像のみ、画像と文字の組み合わせでタイトルを表記できる。タイトルにイメージ画像を組み合わせることで、文字だけのタイトルに比べて視覚的にもアピールすることができ分かりやすいタイトル表示となる。

現状の eXe では、複数の段落を使って WBT 教材を作成しても、それらの段落を区別する機能がなくすべてが同じページ内にベタで連続して表示されるだけであった。最新版の eXe では、「テキスト」機能を使うことで、複数の段落を一つにまとめて表示単位とすることができる。この表示単位のことを eXe では「テキスト」と呼び、「テキスト」毎にタイトルを付けて、複数の「テキスト」を同一ページ内に配置することができる。「テキスト」機能を使うことで、学習項目を細分化しやすくなり、教材の見栄えが良くなっている。ただし、「テキスト」にタイトルとして図を使用した場合、配布資料用に PDF へ変換すると図とタイトル文字が被るというバグが発生している。本学のように、配布資料に PDF を使っている場合は、タイトル画像の機能において図の挿入は使わない方が良い。

図 8 は最新版の eXe でタイトルとして文字と画像をともに使用した場合の例である。ページ内の表示としては、「テキスト」ごとに背景に色が付き、タイトルの画像も豊富に用意されているため、ページ内の見栄えが従来よりも大幅に改善されている。図 8 では、タイトルとして、「タイトルのサンプル」という文字列を使い、その前に講師がレクチャーしているような画像を表示させている。



図 8 タイトル画像の例

### 3.2.4. 「折り畳み」機能と「Menu」機能

「折り畳み」機能も現状の eXe には実装されていないが、スマホやタブレットなどの表示画面が制限された環境で利用する際に有効な機能として最新版の eXe には実装されている。最新版の eXe では、ページよりも小さな「テキスト」の単位、すなわち複数の段落を一つのまとまりとして扱うことができるようになった。そして、その「テキスト」を表示画面で折り畳むことで、スクロール移動を最小限に設定することができる。

図 9 は、1 ページ内に 2 個のタイトルテキストを収めた例である。右側の「+」マークは、現在の項目が折り畳まれていることを意味している。この「+」マークをクリックし、項目の内容を展開すると図 10 のようになり、「+」マークの部分が「-」マークへと変化する。

通常は、現状の eXe と同様に左側にページの情報が表示されており（図 11）、それをクリックすることで、任意のページへ遷移することができる。最新版の eXe でも同様な操作は可能であるが、右上の「Menu」をクリックすると、左側のメニューを消去し、画面を大きく表示することができる（図 12）。限られたウインドウサイズでの視認性が向上する。

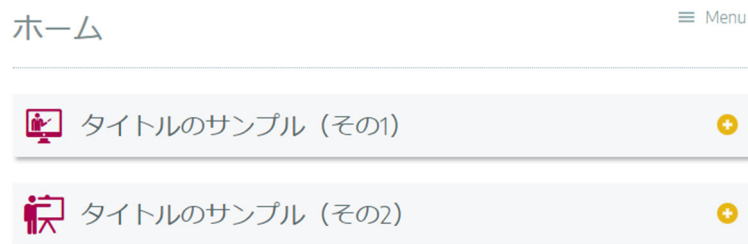


図 9 折り畳まれた表示の例

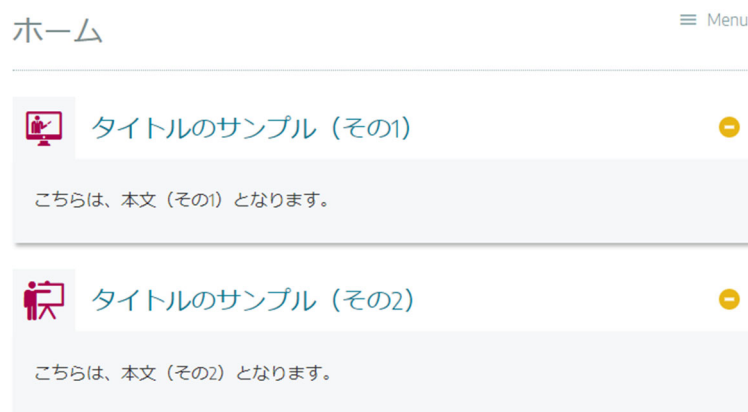


図 10 展開された表示の例



図 11 左側にページメニューが表示

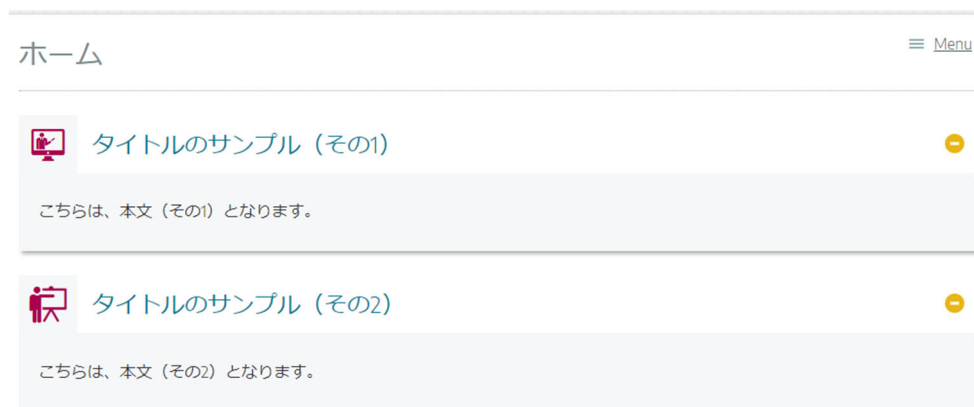


図 12 ページメニューを隠した表示

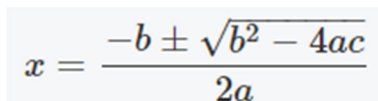
### 3.3. 最新版 eXe に搭載されている数式表示機能

最新版 eXe では、標準で数式表示の機能が備わっており、TeX ライクな数式入力ができる。実際の数式の入力は、編集画面のメニューから「Σ」マークの「Paste mathematical markup」をクリックして表示されるポップアップウィンドウの入力部分に数式を TeX 形式で入力し、「OK」ボタンを押すことで完了する。

このように、最新版の eXe では、メニュー画面から GUI で数式を作成することができ、数式の表現には TeX での表記が使える。GUI からの数式は、独立した行として数式が行内でセンタリングされ表示される。一方、エディタを使って直接数式を入力することもでき、この場合は、インラインでの数式や独立行としての数式を記述することができる。

例えば、「¥(」と「¥)」で数式を囲むとインラインでの数式が表示され、「¥[」と「¥]」または「\$\$」と「\$\$」で数式を囲むと独立した行に数式が表示される。TeX に慣れた人は、インライン数式に「\$」を使いたくなるが、文字としての「\$」と区別するために、eXe では使用することができない。

図 13 は、最新版 eXe での 2 次方程式の解の公式の表示例であり、実際の入力は「`$$x=\frac{-b \pm \sqrt{b^2 -4 a c}}{2a}$$`」のようになる。



$$x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$$

図 13 eXe に組み込まれている数式表示機能での 2 次方程式の解の公式

## 4. 最新版 eXe の機能拡張

最新版の eXe は、現状の eXe の機能を引き継ぎながらも、前章で述べたように、いくつかの新しい機能が追加されている。本章では、最新版の eXe を本学で使用する場合に、現状の eXe のような使用を前提に、どのような機能を拡張するかについて述べる。

### 4.1. 数式表示機能

前章で説明したように、最新版の eXe には標準で数式表示の機能が備わっている。しかし、TeX の表記に慣れている場合、あるいは、すでに TeX で教材を開発している場合に、あらためて eXe の形式に変換して入力する必要がある。本節では、TeX の利用者が無理なく教材を eXe に取り込めるように、TeX の記述に準拠した数式入力から表示までの機能の拡張について説明する。以下では、MathJax<sup>2)</sup>、CortexJS<sup>3)</sup>、KaTeX<sup>4)</sup>について説明する。

#### 4.1.1. MathJax による数式表示

MathJax による数式入力から表示を実現するには、最新版の eXe では、ヘッダセクションにリスト 1 の内容を追記する必要がある。ヘッダセクションへの追加は、「属性」から「パッケージ」タブを開き、「Advanced Options」の「Custom HEAD」セクションに MathJax を使用するためのリスト 1 のスクリプトを入力する。一旦、入力しておけば、すべてのページで数式の使用が可能となる。リスト 1 では、TeX での入力をそのまま適用できるように、インラインでの数式用に「`$`」、独立した数式用に「`$$`」を使えるようにしている。

## リスト 1 MathJax を使用する際のヘッダセクションの内容

```

<script>
window.MathJax = {
  tex: {
    tags: "ams",
    inlineMath: [ ['$','$'], ["\\(", "\\)"] ],
    displayMath: [ ['$$','$$'], ["\\[", "\\]"] ],
    processEscapes: true
  },
  options: {
    ignoreHtmlClass: 'tex2jax_ignore',
    processHtmlClass: 'tex2jax_process'
  }
};
</script>
<script src="https://cdn.jsdelivr.net/npm/mathjax@3/es5/tex-autoload.js"
id="MathJax-script">
</script>

```

MathJax を利用した数式の表示例を図 14 に示しておく。図 14 では、独立した行内の数式として、2 次方程式の解の公式を表示している。なお、実際の数式の入力、**TeX** 形式で、「`$$x=\frac{-b \pm \sqrt{b^2 -4 a c}}{2a}$$`」のように記述している。

$$x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$$

図 14 MathJax による 2 次方程式の解の公式

リスト 1 をヘッダセクションに追加することにより、**TeX** で作成した教材をそのまま再利用することができるのが利点である。図 15 の表示例は、鈴木が担当するゼミナールで使用する **WBT** 教材の一部であり、元の原稿は **TeX** で作成している。これを **eXe** へ読み込ませることで、**TeX** のように綺麗に数式が表示されていることが分かる。また、**MathJax** を使用しているため、数式を右クリックすることで、サブメニューが表示され数式を再利用することができる。

図 15 では、数式に番号を付けて他の個所でその数式を参照できるようにしている。通常の **TeX** であれば、一つのファイルを通して、ページや章節に関係なく数式を参照できるが、**eXe** ではページを跨いで数式を参照することができない。数式に番号を付けて、その数式を他の個所で参照する場合は、同じページに記述するしか方法がない。この場合、内容によっては、数式の参照をするために、1 ページが長くなることもあり、見栄えが悪くなる可能性もある。しかし、最新版の **eXe** には、「折り畳み機能」があるため、これを効果的に使用することで、結果的に見やすいページを作成することができる。

$$I = \int_a^b f(x) dx \quad (1)$$

を計算するのに、積分区間 $[a, b]$ を $n$ 等分して

$$h = \frac{b-a}{n} \quad (2)$$

とおき、 $I$ を和

$$I_n = h \left[ \frac{1}{2} f(a) + \sum_{j=1}^{n-1} f(a+jh) + \frac{1}{2} f(b) \right] \quad (3)$$

図 15 ゼミナールで使用している教材の表示例

#### 4.1.2. CortexJS による数式表示

CortexJS は、単に数式の表示機能だけでなく、動的に数式を処理する機能も実装されている。そのため、リアルタイムで数式を処理することも可能である。数式を表示するための機能が MathLive として、数式を動的に処理する機能が Compute Engine として提供されている。数式の表示である MathLive は、最新版の eXe でも問題なく動作しているが、数式を動的に処理する Compute Engine については、処理そのものは動作するがデータの受け渡しに関しては、処理結果を eXe 側で受け取る機能が実装されていないため、現時点ではコンソール上でのみ確認可能な状態である。インターフェース部分を独自に開発して、処理結果を取り込むことも可能である。

CortexJS は、現在でも開発が続いており、特に、Compute Engine については、未実装の機能もあり、現時点で安定的に使用できるのは、MathLive の方である。しかし、動的に数式を処理することができるという利点は大きいため、今後の開発に期待したいところである。

最新版の eXe で CortexJS を使うには、「Custom HEAD」セクションに CortexJS を使用するためのスクリプトを入力する。スクリプトの内容は、リスト 2 となる。CortexJS は、MathJax との同時利用も可能であることを検証しているため、必要に応じて使い分けることもできる。

リスト 2 CortexJS を使用する際のヘッダセクションの内容

```
<script type="module">
  window.addEventListener('DOMContentLoaded', () =>
    import('/unpkg.com/mathlive?module').then((mathlive) =>
      mathlive.renderMathInDocument()
    )
  );
</script>
```

CortexJS では、行内の数式は、「¥(」と「¥)」で囲む必要があり、「\$」は使用できない。独立した数式の場合は、「¥[」と「¥)」で囲むか、「\$\$\$」で囲むことになる。CortexJS でも TeX をサポートしているが、複数行に渡るような複雑な数式を与えると、TeX のソースがそのまま表示されることがある。この点は、完全に TeX をサポートできていないためと思われる。

図 16 は、CortexJS による 2 次方程式の解の公式の表示例である。実際の入力は、「<math-field virtual-keyboard-mode="manual"> x=¥frac{-b¥pm¥sqrt{b^2-4ac}}{2a}</math-field>」となるが、エディタ画面では、数式のみが表示となる。数式を修正する場合は、HTML エディタを起動して修正する必要がある。


$$x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$$


図 16 CortexJS による 2 次方程式の解の公式

図 16 のように CortexJS では、表示されている数式をマウスでクリックすれば、動的に数式に修正を加えることができる機能を実装している。図 16 の右側にあるキーボードマークをクリックすれば、仮想キーボード (図 17) が表示され、カーソルが数式の中へ移動して GUI を使って動的に数式を修正することができる。図 17 から分かるように、多様な数式、関数や記号も用意されているので、これらを使い分けることで様々な数式を構成することができる。この機能は、数学系の科目において、学生から数式を受け取り、Compute Engine で動的に処理をする際に利用できる。e ラーニングの数学教材開発において、とても便利な機能であると考えられる。

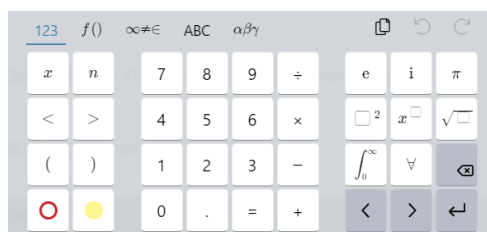


図 17 仮想キーボード

#### 4.1.3. KaTeX による数式表示

KaTeX は、MathJax よりも高速に動作するというのが特徴の一つである。近年では MathJax も改定を重ね、速度については両者の差はそれほどないものと思われる。しかし、現在でも KaTeX を利用して数式を表現しているサイトは、多く見受けられる。その一つの理由として、開発当初から、TeX を意識していることがあげられる。そのため、KaTeX では、行内の数式表現に「\$」を許しており、TeX のソースをそのまま再利用できるように



なっている。しかし、実際に試してみると、CortexJS と同様に、複雑な数式には完全には対応できていないことが分かる。一方で、数式の表示（特に見栄え）については、より TeX に近いものとなっている。

KaTeX を利用するには、「Custom HEAD」セクションにリスト 3 のスクリプトを入力する。KaTeX では、行内の数式は、「\$」で囲むか、「¥(」と「¥)」で囲む必要がある。独立した数式の場合は、「¥[」と「¥]」で囲むか、「\$\$」で囲むことになる。

リスト 3 KaTeX を使用する際のヘッダセクションの内容

```
<link rel="stylesheet"
href="https://cdn.jsdelivr.net/npm/katex@0.16.2/dist/katex.min.css"
integrity="sha384-
bYdxxUwYipFNohQlHt0bjN/LCpueqWz13HufFEV1SUatKs1cm4L6fFgCi1jT643X"
crossorigin="anonymous">
<script defer
src="https://cdn.jsdelivr.net/npm/katex@0.16.2/dist/katex.min.js"
integrity="sha384-
Qsn9KnoKISj6dI8g7p1HB1NpVx0I8p1SvlwOldgi3IorMle61nQy4zEahWYtljaz"
crossorigin="anonymous">
</script>
<script defer
src="https://cdn.jsdelivr.net/npm/katex@0.16.2/dist/contrib/auto-
render.min.js" integrity="sha384-
+VBxd3r6XgURycqtZ117nYw4400cIax56Z4dCRWbxyPt0Koah1uHoK0o4+/RRE05"
crossorigin="anonymous">
</script>
<script>
    document.addEventListener("DOMContentLoaded", function() {
        renderMathInElement(document.body, {
            delimiters: [
                {left: '$$', right: '$$', display: true},
                {left: '$', right: '$', display: false},
                {left: '¥¥(', right: '¥¥)', display: false},
                {left: '¥¥[', right: '¥¥]', display: true}
            ],
            throwOnError : false
        });
    });
</script>
```

図 18 は、KaTeX による 2 次方程式の解の公式の表示例である。実際の入力は、「
$$x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$$
」のようになる。

$$x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$$

図 18 KaTeX による 2 次方程式の解の公式

#### 4.1.4. 数式表示機能のまとめ

数式表示の方法は主に上記の三通りがあり、本研究では、検証の結果として **MathJax** を利用することを推奨する。**CortexJS** は、動的に数式を処理できる利点があるが、未実装の機能もあり、安定的に利用することは難しいと判断した。ただし、動的に処理できる機能は、数学教材においては、有利であることから、今後の開発状況次第で取り入れても良いと考える。**KaTeX** は、現在となっては、表示速度の面で、さほど利点がないこともあり、採用を見送ることとする。したがって、数式に関しては、**TeX** との互換性に優れている **MathJax** を実装することとした。

#### 4.2. 文字飾り機能

最新版の **eXe** における文字飾りについては、現状の **eXe** と同様に **CSS** で実現できる。最新版の **eXe** では、標準でいくつかの基本となる **CSS** がサンプルとして提供されており、それらの **CSS** のスタイルを詳細に編集できるエディタも最新版の **eXe** には搭載されている。利用者は、サンプルの **CSS** を使いつつも、オリジナルのスタイルを自由に定義できるようになっている。

本学での教材開発では、教員が自由に **CSS** を編集するのではなく、現状の **eXe** のように、定型の **CSS** を事前にセットアップしておき、利用者がそれらを選択して文字飾りを施すようにすることが望ましい。本研究では、現状の **eXe** から本学で独自に追加した **CSS** を取り出し、それを最新版の **eXe** に移植することで、正常に動作するかどうかを検証した。

一般に、**eXe** の使い方としては、通常のワープロのように、画面で確認しながら編集することはできない。確認画面（プレビュー）と編集画面が異なり、さらに、**HTML** を直接編集できるエディタ画面もあり、**CSS** の移植には注意が必要である。以下では、はじめに、**Linux** 版について移植の過程を説明する。その後、**Windows** 版について説明するが、**Windows** 版では、**Linux** 版とはフォルダの構成が異なるので注意が必要である。

##### 4.2.1. Linux による文字飾りの実装

現状の **eXe** と同様に最新版の **eXe** にメニュー形式で文字飾りを実装する場合には、**eXe** の動作原理に注意しなければならない。**eXe** では、1. メニューの改変または新規追加、2. プレビュー画面での文字飾りの実装、3. 編集画面での文字飾りの実装の三つの該当箇所にあるファイルを修正しなければならない。特に、2. と 3. は、同一の文字飾りの定義を全く別のファイルに記述する必要があるため注意が必要である。本研究では、1. については、新規のメニュー追加ではなく、既存のメニューを改変し、文字飾りの項目を追加することとする。

最新版の **eXe** では、あらかじめサンプルとして **CSS** が提供されているため、既に存在している **CSS** を選択するメニューに文字飾りの項目を追加する。この場合、  
「./exelearning27/exelib/exe/webui/scripts/tinymce\_4/js/tinymce/plugins/easyattribute

s/plugin.min.js」に文字飾りの項目を追加する。このファイルの中の「rightClasses」にすでにいくつかのメニューが記述されているので、リスト4の「Midasivline\_R」を項目として追加した。この名前は、現状の eXe でも使われているもので、タイトルの前に縦線が表示されるものである。なお、メニューは、「rightClasses」に記載されている順に表示されるため、「exe-block-success」の後に記述した。

リスト4 新しい項目の追加

```
{text: 'Midasivline_R', value: 'Midasivline_R'},
```

プレビュー画面での文字飾りに関する CSS を追加するには、「./exelearning27/style/base.css」に文字飾りに関する CSS を追加する。このファイルの中には、標準で提供されている CSS が列挙されているので、適当な場所に、リスト5の「Midasivline\_R」の CSS を記述する。

リスト5 プレビュー画面に適用する CSS

```
.Midasivline_R {
  border-left : 7px solid #e3297d;
  padding     : .6em .8em;
  margin      : 1em 0 .6em 0;
  font-weight : bold;
  font-size   : 1.1em !important;
  display     : inline-block;
}
```

編集画面での文字飾りに関する CSS を追加するには、「./exelearning27/exelib/exe/webui/css/extra.css」に文字飾りに関する CSS を追加する。このファイルの中には、標準で提供されている CSS が列挙されているので、適当な場所に、リスト5と同じものを記述する。

プレビュー画面と編集画面で使用する CSS を異なるファイルに同じ定義で記述することは、メンテナンスの面で決して効率的とは言えない。例えば、CSS に修正が生じた際に、どちらかのファイルを未修正のまま使用すると、プレビュー画面と編集画面での表示が異なってしまう。したがって、CSS は、一つのファイルで管理して、そのファイルをプレビュー画面と編集画面用に使用するようにしたい。

上記の変更後に eXe を起動することで、新しい文字飾りである「Midasivline\_R」が有効となる。ここでは、既存のメニューに追加しているため、編集画面で文字飾りを付けた文字列をマウスで選択した後に、赤丸で示した左上の編集をクリックする（図19）。

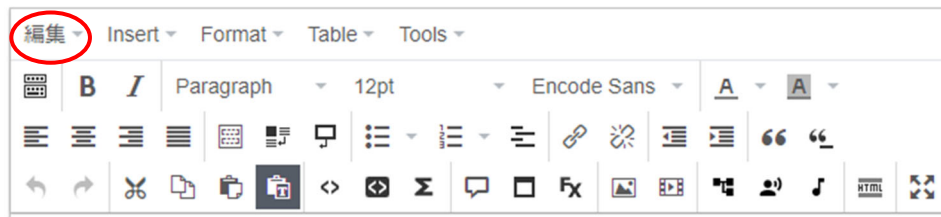


図 19 編集画面での文字飾り用のメニュー

編集をクリックすると、図 20 左側のようなメニューが表示されるので、「Insert/Edit Attributes」をクリックし、図 20 右側のような「Insert/Edit Attributes」のメニューを表示させる。「Add CSS class」からプルダウンで「MidasiVline\_R」を選択し、クリックする。

図 20 では、eXe であらかじめ与えられている CSS と同列に新規の CSS を追加したため、図 20 右側のように、他の CSS も表示されている。また、図 20 左側のメニューの一番下の「Edit CSS Style」を使用して、自由に CSS を設計することもできる。しかし、本学の場合、現状の eXe に合わせるのであれば、これらの標準で与えられている項目や機能は取り除き、代わりに現状の eXe で使用している CSS のみを再定義しても良い。また、メニューの構成についても、本研究のように、既に存在するメニューを再利用するのではなく、現状の eXe と同様に独自のメニュー画面を設計しても良い。

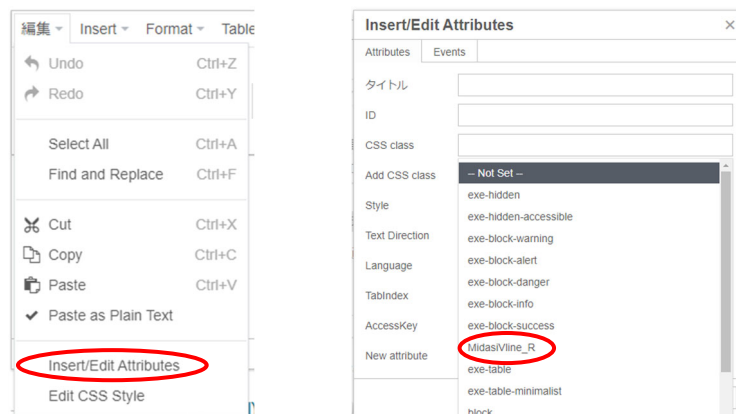


図 20 CSS 選択メニュー

プルダウンで「MidasiVline\_R」を選択すれば図 21 のように選択されている文字列に飾りが付く。図 21 は編集画面での表示例であるが、編集画面を終了すれば、プレビュー画面が表示される。プレビュー画面を単独で表示したものが、図 22 の表示例であり、編集画面と同様の飾りが付いた状態で表示されていることが分かる。

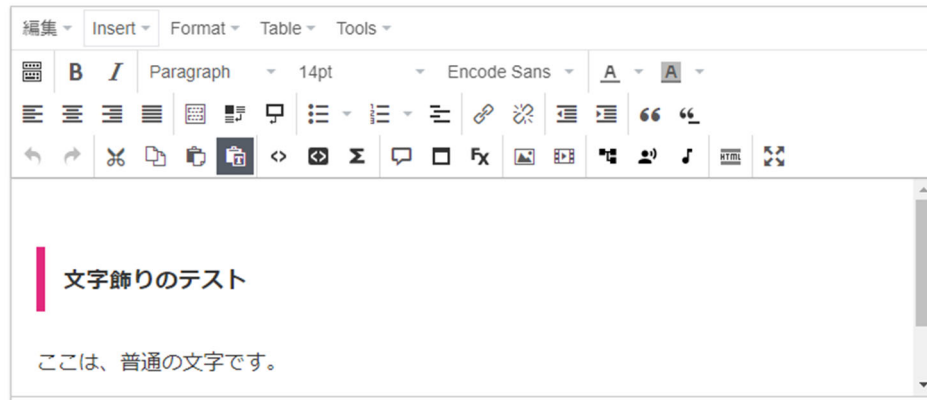


図 21 編集画面での文字飾りの様子

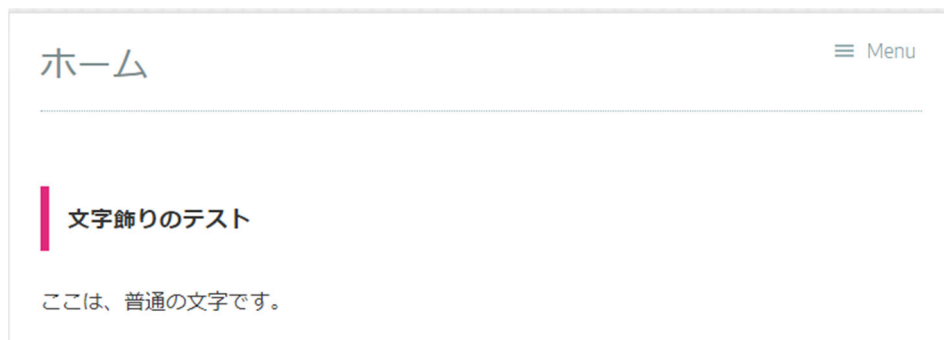


図 22 プレビュー画面での文字飾りの様子

#### 4.2.2. Windows による文字飾りの実装

Windows 版でも Linux 版と同様に、CSS を組み込むことが可能である。それぞれのファイルの場所及び本研究で追記した内容を以下に記しておく。

メニューの改変または新規追加は、「portable-INTEF.exe-2.7-win¥exelib¥scripts¥tinymce\_4¥js¥tinymce¥plugins¥easyattributes¥plugin.min.js」にあるファイルを編集する。ここでは、「exe-block-success」の後に、リスト 6 の文字飾り「MidasiVline\_R」及び「MidasiGrade\_R」、ターミナル用の表示「BoxTerminal」、黒板風の表示「BoxKokuban」をメニューとして追記した。

#### リスト 6 新しい項目の追加

```
{text: 'MidasiVline_R', value: 'MidasiVline_R'},  
{text: 'MidasiGrade_R', value: 'MidasiGrade_R'},  
{text: 'BoxTerminal', value: 'BoxTerminal'},  
{text: 'BoxKokuban', value: 'BoxKokuban'},
```

プレビュー画面での文字飾りの実装は、「portable-INTeF.exe-2.7-win¥style¥base.css」にあるファイルを編集する。Linux 版と同様に、ファイル中の適当な場所にリスト 7 の CSS を記述する。

リスト 7 プレビュー画面に適用する CSS

```
.Midasivline_R {
  border-left: 7px solid #e3297d;
  padding: .6em .8em;
  margin: 1em 0 .6em 0;
  font-weight: bold;
  font-size: 1.1em !important;
  display: inline-block;
}
.Midasigrade_R {
  display: inline-block;
  width : 95%;
  position: relative;
  border-left: 7px solid #e3297d;
  padding: 0.4em 0.8em;
  font-weight: bold;
  font-size: 1.2em !important;
  margin: 1em 0 .7em 0;
  color: #e3297d;
  background: -moz-linear-gradient(to right, #fff 79%,#fff 79%,rgba(255,
255, 255, 0) 100%);
  background: linear-gradient(to right, #fff 79%,#fff 79%,rgba(255, 255,
255, 0) 100%);
}
.Midasigrade_R::after {
  content:'';
  display: block;
  position: absolute;
  top: 43px;
  left: -7px;
  height: 1px;
  width: 100%;
  background: -moz-linear-gradient(left, #e3297d 79%, #e3297d 79%, #fff
100%);
  background: linear-gradient(to right, #e3297d 79%,#e3297d 79%,#fff
100%);
}
.BoxTerminal {
  position: relative;
  display: inline-block;
  width: 95%;
  background: #4d4d4d;
  padding: .6em 1em .8em .8em;
  margin: 1.4em 0 1.5em .1em;
  color: #fff;
  font-family: "Source Code Pro", Consolas, monospace, 'Courier New',
```

```
Courier, Monaco;
  line-height: 1.4em;
  letter-spacing: 0em;
  font-size: 1em;
  border: 1px solid #3498db;
  -moz-border-radius: 4px;
  border-radius: 4px;
}
.BoxTerminal::before {
  font-size: 1.2em;
  font-weight: bold;
  font-family: "Source Code Pro", "Consolas", "Bitstream Vera Sans Mono",
"Courier New", "メイリオ", Meiryo, "ヒラギノ角ゴ Pro W3", "Hiragino Kaku
Gothic Pro", Courier, monospace;
  letter-spacing: 2px;
  color: #3498db;
  content: '■ ' attr(title);
  position: absolute;
  top: -1.4em;
  left: 1em;
}
.BoxKokuban {
  display: inline-block;
  border: ridge 3px #936300;
  padding: 0.8em;
  color: #fff;
  background-color: #060;
  margin: 0 0 .8em 0;
}
```

編集画面での文字飾りの実装は、「portable-INTEF-exe-2.7-win¥exelib¥css¥extra.css」にあるファイルを編集する。ファイル中の適当な場所にリスト7と同じものを記述する。

各ファイルを修正後に、eXe を起動して編集画面から「Insert/Edit Attributes」のメニューを表示させ、「Add CSS class」をプルダウンで確認すると図23のように新しい項目が四つ追加されていることが分かる。

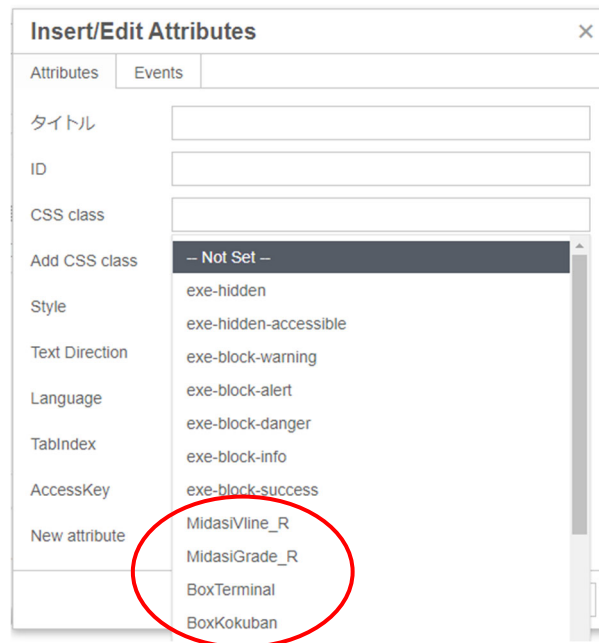


図 23 追加した CSS 選択メニュー

Windows 版では、デフォルトのブラウザが Edge に設定されているが、起動ブラウザは設定により変更可能である。ブラウザに関しては、Edge だけでなく、Firefox や Chrome でも正常に動作することを確認している。

図 24 は、追加したメニューから「MidasiGrade\_R」、「BoxTerminal」、「BoxKokuban」を使用した例である。



図 24 ターミナル表示と黒板風表示の例



ターミナル表示では、表示枠の上に、端末を表すコンピュータの画像を張り付けており、フォントもターミナル風のアレンジをしている。黒板表示では、表示範囲に合わせて、黒板の枠の大きさが決まるようになっている。チョークや黒板消しなどを配置することも可能である。

## 5. おわりに

本研究では、本学で使用している現状の eXe に対して、最新版の eXe が利用できるかどうかを検証した。特に、検証の対象としたのは、数式と文字飾りである。数式に関しては、ヘッダセクションを定義することで、MathJax を使用でき、TeX のソースもそのまま再利用できることを確認することができた。現状の eXe でも「\$」の扱いが問題になっているが、最新版の eXe においても、「\$」を使う場合と、使わない場合に分けてスタイルを作成すれば問題なく対応できる。また、TeX のソースも再利用できるため、すでに作成済みの TeX ファイルを使い効率的に教材開発ができる。

文字飾りについては、CSS を使ったタイトル用の飾りについて、現状の eXe と同様に動作することの検証が済んでいる。また、現状の eXe にもあるように、ターミナルや黒板風の複数行を対象としたまとまりのある文章への飾りについても動作を検証している。文字飾りのレイアウトの変更時や、黒板にチョークや黒板消しを配置する場合には、個別に CSS の調整が必要である。

以上のように、最新版の eXe への移行は、それほど問題もなく、実施できることを確認することができた。必要であれば、メニューの構成や、CSS の調整を経て、全体のスタイルの検討をした上で、実用化できれどと考えている。また、本稿では、代表的な機能のみを取り上げて解説しているため、実用化に際しては、ユーザマニュアル等で使用方法を周知できればと考える。

## 注および参考文献

- 1) <https://exelearning.net/en/> (確認日: 2022 年 12 月 28 日)  
eXe ラーニングとは、教育用の Web コンテンツを作成するために使用される無料のオープンソースエディタのことである。HTML を知らなくても GUI での操作を通して、テキストベースだけでなく、動画を含む教材も作成することができる。作成したコンテンツは、eXe ラーニングのプレビュー画面で確認することができる。
- 2) <https://www.mathjax.org/> (確認日: 2022 年 12 月 28 日)
- 3) <https://cortexjs.io/> (確認日: 2022 年 12 月 28 日)
- 4) <https://katex.org/> (確認日: 2022 年 12 月 28 日)



## eラーニング研究 第11号

---

2022年（令和4年）12月31日 発行

発行者 サイバー大学

〒813-0017

福岡県福岡市東区香椎照葉 3-2-1

[URL https://www.cyber-u.ac.jp/](https://www.cyber-u.ac.jp/)

編集 サイバー大学

---

