WBT 教材の効率的な開発法に関する研究

鈴木　秀男[1]，長島　雄平[2]，田中　健太郎[2]

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[1)]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[1)]）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も併せて検討し、最新版 eXe の利用可能性について検討する。

[1] サイバー大学 IT 総合学部・教授
[2] 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 を使っての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

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。）
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）
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）
④ 不要であると思われる機能があればお書きください。
⑤ 追加で必要であると思われる機能があればお書きください。
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？
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？
⑧ その他、eXe について、ご意見があればご記入ください。

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も、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。

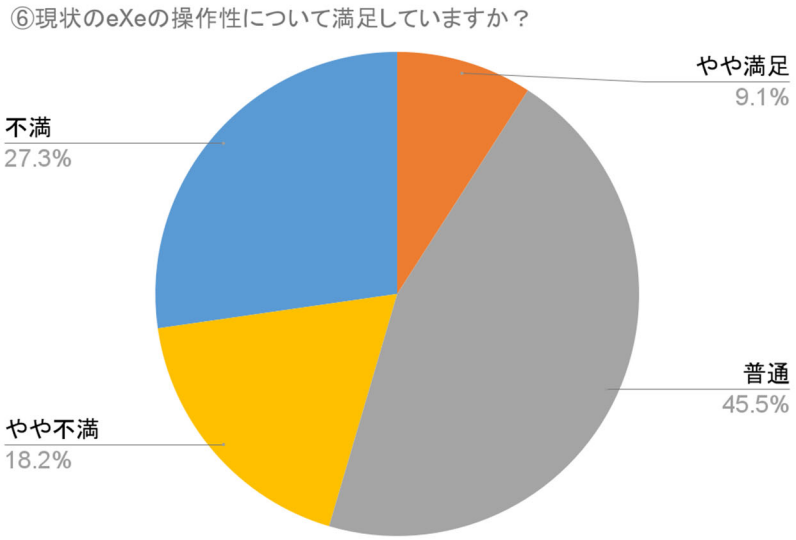

図１　eXe の操作性に関するアンケート結果

「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。

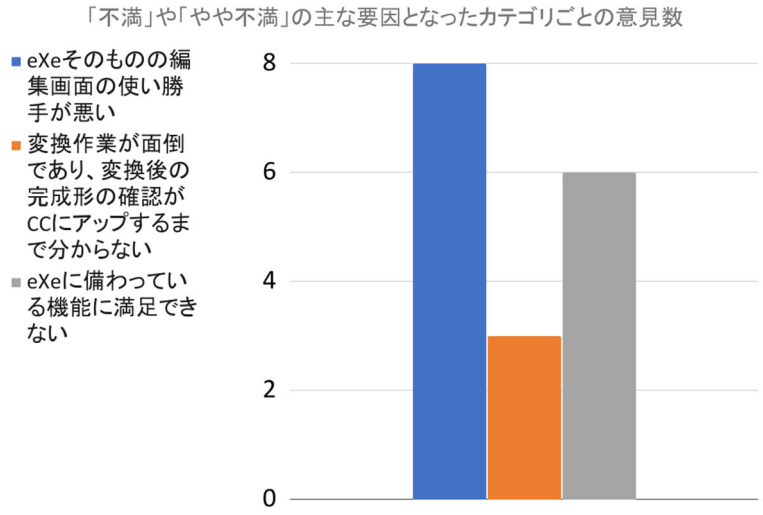

図２　「不満」や「やや不満」の主な要因

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）。

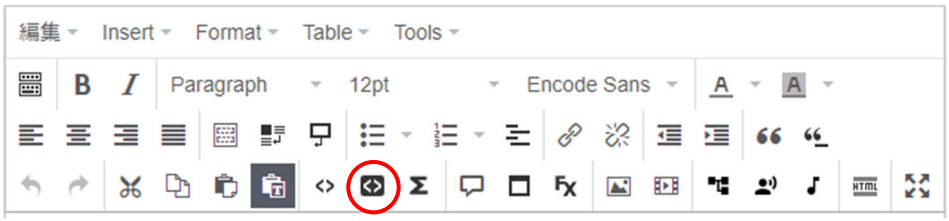

図３　テキスト編集画面のメニュー（Insert HTML code as text）

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。

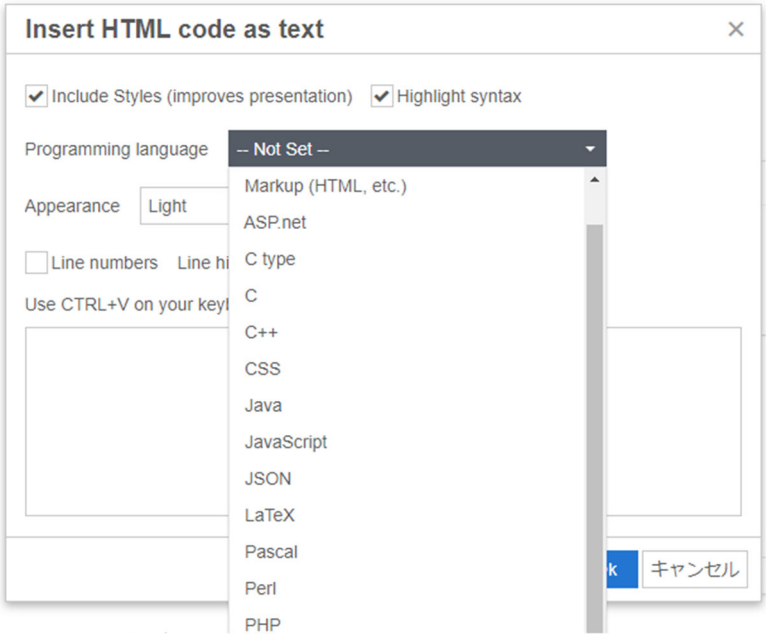

図４　Insert HTML code as text 画面

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。

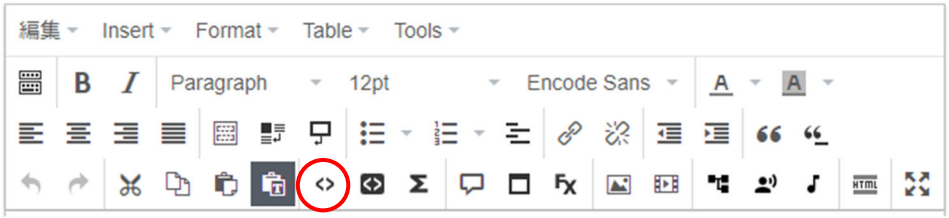

図６　テキスト編集画面のメニュー（Paste HTML fragment(embed code)）

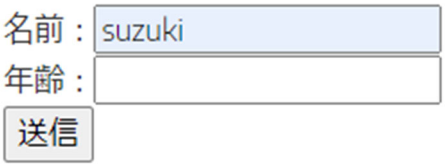

図 7　埋め込まれた HTML に eXe から直接入力

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。

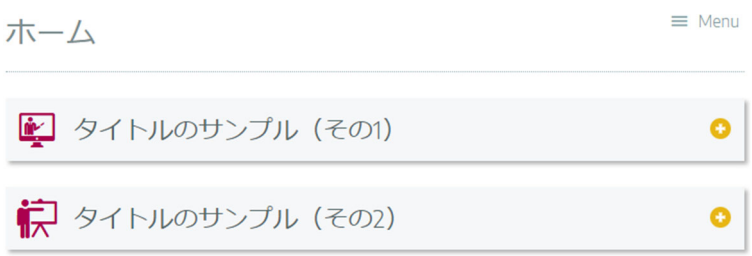

図９　折り畳まれた表示の例

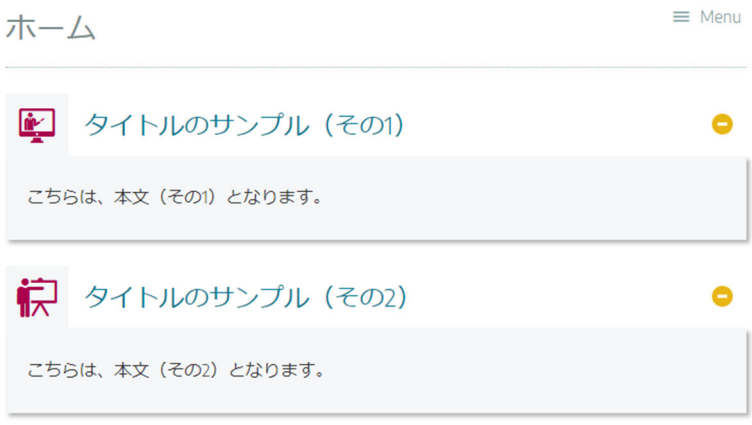

図 10　展開された表示の例

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[2)]、CortexJS[3)]、KaTeX[4)]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-chtml.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、実際の数式の入力は、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している 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があり、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。

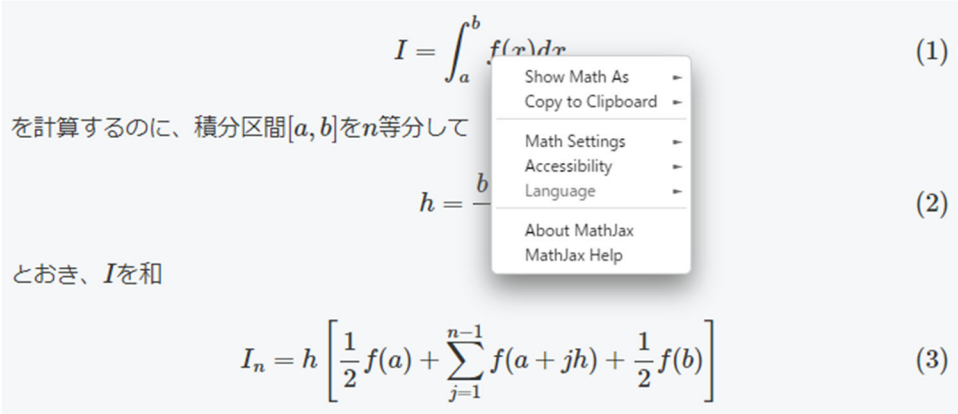

図 15　ゼミナールで使用している教材の表示例

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、数式のみの表示となる。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 からも分かるように、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。

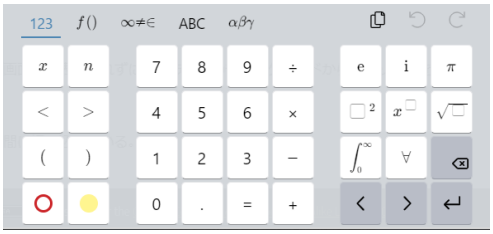

図 17　仮想キーボード

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こと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-
Qsn9KnoKISj6dI8g7p1HBlNpVx0I8p1SvlwOldgi3IorMle61nQy4zEahWYtljaz"
crossorigin="anonymous">
</script>
<script defer
src="https://cdn.jsdelivr.net/npm/katex@0.16.2/dist/contrib/auto-
render.min.js" integrity="sha384-
+VBxd3r6XgURycqtZ117nYw44OOcIax56Z4dCRWbxyPt0Koah1uHoK0o4+/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、「`$$x=¥frac{-b ¥pm ¥sqrt{b^2 - 4 a c}}{2a}$$`」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ので注意が必要である。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い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)。

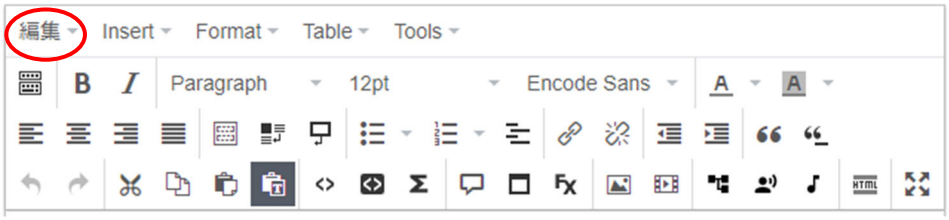

図 19　編集画面での文字飾り用のメニュー

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。

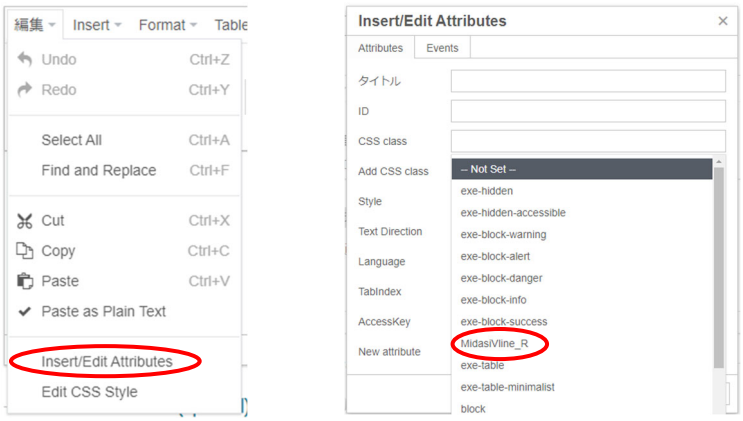

図 20　CSS 選択メニュー

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。

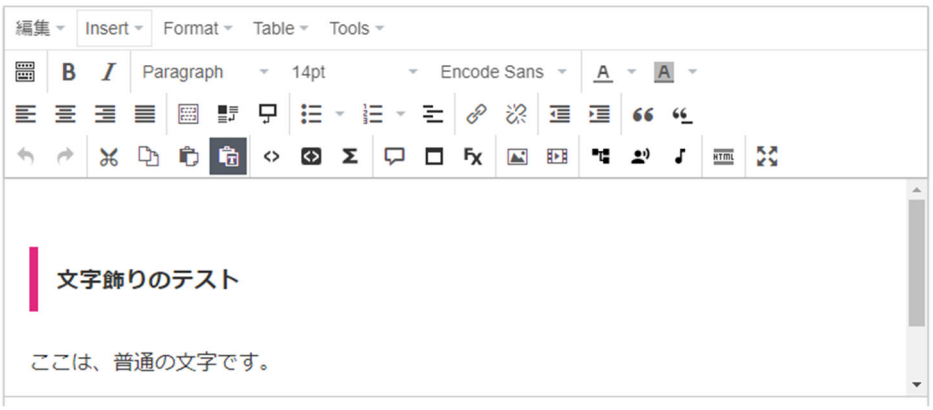

図 21　編集画面での文字飾りの様子

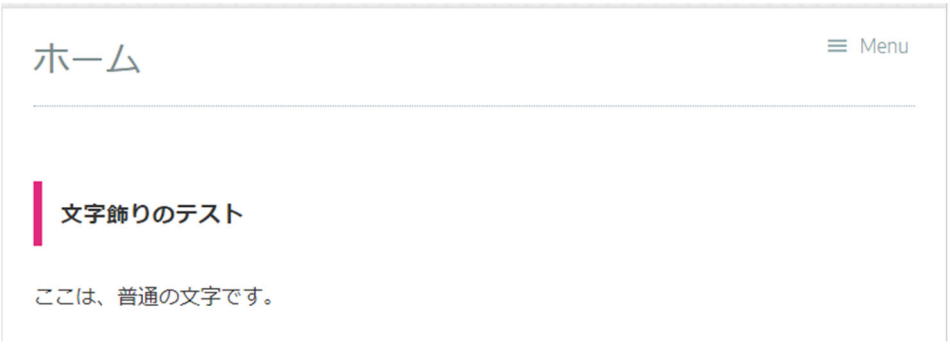

図 22　プレビュー画面での文字飾りの様子

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。

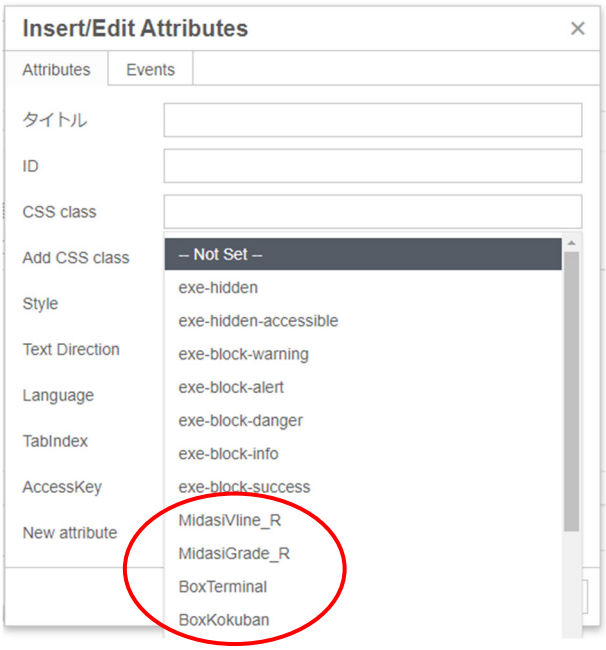

図 23　追加した CSS 選択メニュー

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に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、実用化できればと考えている。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）