

教育的配慮に基づく Web コンテンツ制作支援環境の開発

佐々木雄太*, 伊藤陽介**

高等学校「情報 I」の内容には情報デザインやプログラミング, 中学校技術・家庭科(技術分野)D 情報の技術の内容には双方向性のあるコンテンツのプログラミングの学習が含まれる。情報デザインとプログラミング教育における題材として Web コンテンツ制作が挙げられる。テキストエディタで作成した Web コンテンツは, Web ブラウザで表示させる必要があり, 両者の切り替えによって円滑な学習をしづらいという課題があった。本論文では, 情報デザインとプログラミングの各学習内容に対する教育的な配慮に基づき Web コンテンツの制作に要求される仕様を整理した。その結果に基づき設計し, C#を用いて Windows OS で動作するように Web コンテンツ制作支援環境を開発する方法とその利用手順について述べた。

[キーワード: Web コンテンツ, 制作支援環境, 情報デザイン, プログラミング]

1. はじめに

国際的な情報教育の高度化の流れを踏まえ, 2018 年に告示された高等学校学習指導要領では共通教科情報科の科目構成が大幅に改訂され「情報 I」が設置された。この内容は「情報社会と問題解決」, 「コミュニケーションと情報デザイン」, 「コンピュータとプログラミング」, 「情報通信ネットワークとデータの活用」の 4 つである[1]。情報デザインやプログラミングの学習において, 中学校技術・家庭科(技術分野)の内容D 情報の技術の学習内容である「双方向性のあるコンテンツのプログラミング」からの継続性を考慮すると, 情報デザインとプログラミング教育における題材の 1 つとして Web コンテンツ制作が挙げられる[2]。

Web コンテンツ制作は, 専用ソフトウェアを使う方法とテキストエディタを使う方法に大別される。情報デザインに加えてプログラミングの教育も行う場合, 後者の方が, 自由度が高いため指導を行いやすいと考えられる。テキストエディタで作成した Web コンテンツは, Web ブラウザで表示させる必要があり, 両者の切り替えによって円滑な学習をしづらいという課題があった。

そのため本論文では, 情報デザインとプログラミングの各教育, 及び, ICT 環境を配慮して設計した Web コンテンツ制作支援環境について述べるとともに, その開発例を示す。

2. Web コンテンツ制作支援環境の設計

Web コンテンツ制作支援環境は, 主に「情報 I」や技術・家庭科(技術分野)のプログラミング教育の授業で利用することを想定し, 情報デザイン教育, プログラミング教育, 及び, ICT 環境に配慮して設計する。HTML, CSS, JavaScript の 3 種類を個別のテキストエディタを用いて円滑にコーディングすることで, 学習者が目的に応じた Web コンテンツを設計・制作し, 評価・改善を順序良く効率的に行えることを目標とする。本制作支援環境に要求される主な仕様を表 1 に示す。

テキストエディタは, 文字コードが連続するデータのみから構成されるファイルの作成, 編集, 保存等の各処理を行う。テキストエディタによって編集支援機能は様々であり, 情報デザインとプログラミングの各教育に必要なものを選定する。編集支援機能が乏しいとコーディング, 修正に時間を要したりする。一方, 編集支援機能が充実しすぎていると, それらの機能を習熟するまでに時間を要したり, 誤操作したりする等, 初学者にとって必ずしも使いやすいものとはならない側面がある。

Web コンテンツを制作する学習過程では, テキストエディタで編集した各ファイルを Web ブラウザで表示し, コンテンツの内容を確認し修正する作業を繰り返す。タブレット型端末のように画面が小さく解像度が低い情報端末において, テキストエディタと Web ブラウザを同時に表示することが難しい場合, 両者を切り替えて表示することになり, 煩雑な操作が発生する。特に, 画面の切り替えに

* 鳴門教育大学大学院 高度学校教育実践専攻 教科・総合系 技術・工業・情報科教育コース 大学院生

** 鳴門教育大学大学院 高度学校教育実践専攻 教科・総合系 技術・工業・情報科教育コース

よる学習者の思考過程の中断を招くことで制作活動を阻害する一因になることが懸念される。そのため1つの画面内に、テキストエディタと制作しているWebコンテンツを表示する領域を配置する必要がある。

また、一般的なWebコンテンツはHTML、CSS、JavaScriptで記述されるが、三者の言語仕様は大きく異なる。そのため、単一のテキストエディタでまとめて編集するのではなく、それぞれ個別に配置されたテキストエディタを用いることによって、各言語に対応した予約語等の候補表示や補完機能、構文に考慮した色分けや強調表示(シンタックスハイライト)等によって制作を支援する。JavaScript用ライブラリを用いて外部プログラムとの連携も可能にし、地図コンテンツの表示や操作ができるLeaflet[3]や人工知能による画像認識等の処理ができるTensorflow[4]等も利用できるようにする。

Webコンテンツの制作、表示、修正の各段階を迅速に反復でき、学習者にとって制作手順が明確となるように、画面の左右にテキストエディタやWebコンテンツの表示領域を配置する。図1に設計したWebコンテンツ制作支援環境の画面構成を示す。利用者は、二重線で示した境界を移動させることによって各領域の大きさをコーディングしやすいように変更できる。

まず、学習者はWebコンテンツを構成する各ファイルが記憶されたフォルダを開く。HTML、CSS、JavaScriptの各コードが対応するテキストエディタの編集領域に表示される。次に、学習者は予約語等の候補表示や補完機能を使いながら、各言語仕様に対応する色分けや強調表示を参考にしてコーディングする。全角文字や全角空白の確認も行う。Webコンテンツをフォルダに保存した後、制作中のWebコンテンツを画面右側の領域に表示し、その内容を確認し必要に応じて修正する。

インターネット接続や無線LAN環境が不安定な状態であっても授業を実施できるように、オンラインのみならず、オフラインの状態でも、Webコンテンツ制作支援環境を利用できるようにする。

3. Webコンテンツ制作支援環境

3.1 開発方法

要求仕様に基づいて表2に示すソフトウェアを用いてWebコンテンツ制作支援環境を開発した。テキストエディタは、表1に示した要求仕様a7とa8を備えるAvalonEdit[5]を採用した。Webコンテンツ表示領域は、Microsoft Edge WebView2[6]を採用

表1 Webコンテンツ制作支援環境の要求仕様
(a) 情報デザイン教育に関する要求仕様

a1	Webコンテンツの「設計・制作・表示(実行)・評価・改善」の全てまたは部分的な段階を容易に繰り返すことができる。
a2	アイコン、ピクトグラム、地図のモデル化等によって情報を抽象化する方法に対応する。
a3	文字の配置、ページレイアウト、階層的なコンテンツ、ハイパーリング等を含むコンテンツ制作ができる。
a4	ユニバーサルデザイン、ユーザビリティ、アクセシビリティ、シグニファイア等を考慮したコンテンツ制作ができる。
a5	文字、図形、画像、映像、数式等の様々なメディアを組み合わせたコンテンツ制作ができる。
a6	全角と半角の文字や空白を容易に確認できる。
a7	予約語、コメント、リテラル、演算子、括弧、区切り文字等を構文に考慮して色分けや強調表示する。
a8	予約語等の候補表示、補完機能をもつ。
a9	部分またはすべての模写的コンテンツ制作ができる。

(b) プログラミング教育に関する要求仕様

b1	データやデータ構造、プログラムの構造を容易に理解できる。
b2	外部のプログラムとの連携ができる。
b3	プログラムの作成から動作確認、不具合の修正を一貫した操作で実施できる。
b4	部分またはすべての模写的Webコンテンツ制作ができる。

(c) ICT環境に配慮した要求仕様

c1	タブレット型端末のようにキーボードが無くても容易に文字入力できる。
c2	画面が小さく解像度が低い情報端末においても、コードの編集画面の切り替えが発生しない。
c3	無線LAN環境が不安定な状態であっても利用できる。
c4	ローカル側にコンテンツをファイルとして保存できる
c5	制作したコンテンツを1つのファイルにまとめ、管理できる。

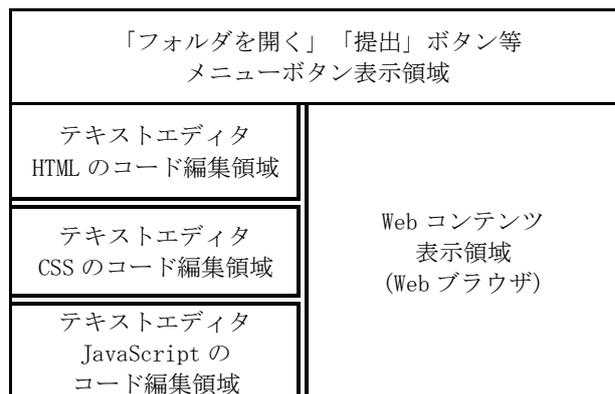


図1 Webコンテンツ制作支援環境の画面構成

した。Microsoft Edge WebView2は、開発者ツールを使用することも可能でエラー発見に役立つ。

Web コンテンツの制作やプログラミングにおいて全角と半角の文字を完全に区別してコーディングすることは必須の技能であるが、初学者にとって難易度の高い側面がある。全角と半角の空白については両者の区別は極めて難しく、数字や英字、記号等の字形も全角と半角で類似しており両者の

区別は難しい。そのため、本制作支援環境では、「全角文字の確認」ボタンを配置し、含まれている全角文字が何行目に存在するか表示する機能を備えている。

図2では、画面右側のWebコンテンツ表示領域内の「コピー」ボタンをクリックすると、テキストボックス内の文字列がコピーされるコンテンツを制作している状態を示している。水色にハイライト(強調表示)された行のみ編集可能となっており、編集行の制限により、入力間違い等を低減させることができる。HTMLのコード編集領域では、タグが青色、属性が黒色、文字列が青色等で表示される。CSSのコード編集領域では、セレクターは青字等で表示される。JavaScriptのコード編集領域では、予約語が青色、数値が藍色、文字列が茶色等で表示される。HTML、CSS、JavaScriptそれぞれでプログラムの構造が分かりやすくなっている。「保存して反映」ボタンをクリックすると、画面右側にWebコンテンツが即座に反映される。JavaScriptのプログラムも問題なく動作する。

表2 開発に用いたソフトウェア

(a) 開発環境

OS	Windows11, バージョン 23H2
統合開発環境	Microsoft Visual Studio 2022
バージョン管理システム	Git, GitHub

(b) コーディング言語

プログラミング言語	C# 7.3
UI 記述言語	XAML

(c) フレームワーク

UI フレームワーク	WPF Version 4.5
ソフトウェア開発フレームワーク	.NET Framework 4.7.2

(d) ライブラリ

テキストエディタ	AvalonEdit[5]
Web ブラウザ	Microsoft Edge WebView2[6]
フォルダ選択ダイアログ	WindowsAPICodePack-Core[7] WindowsAPICodePack-Shell[8]

3.2 機能

各種ボタン等有している機能について表3に示す。Webコンテンツ制作支援環境を起動した時点は

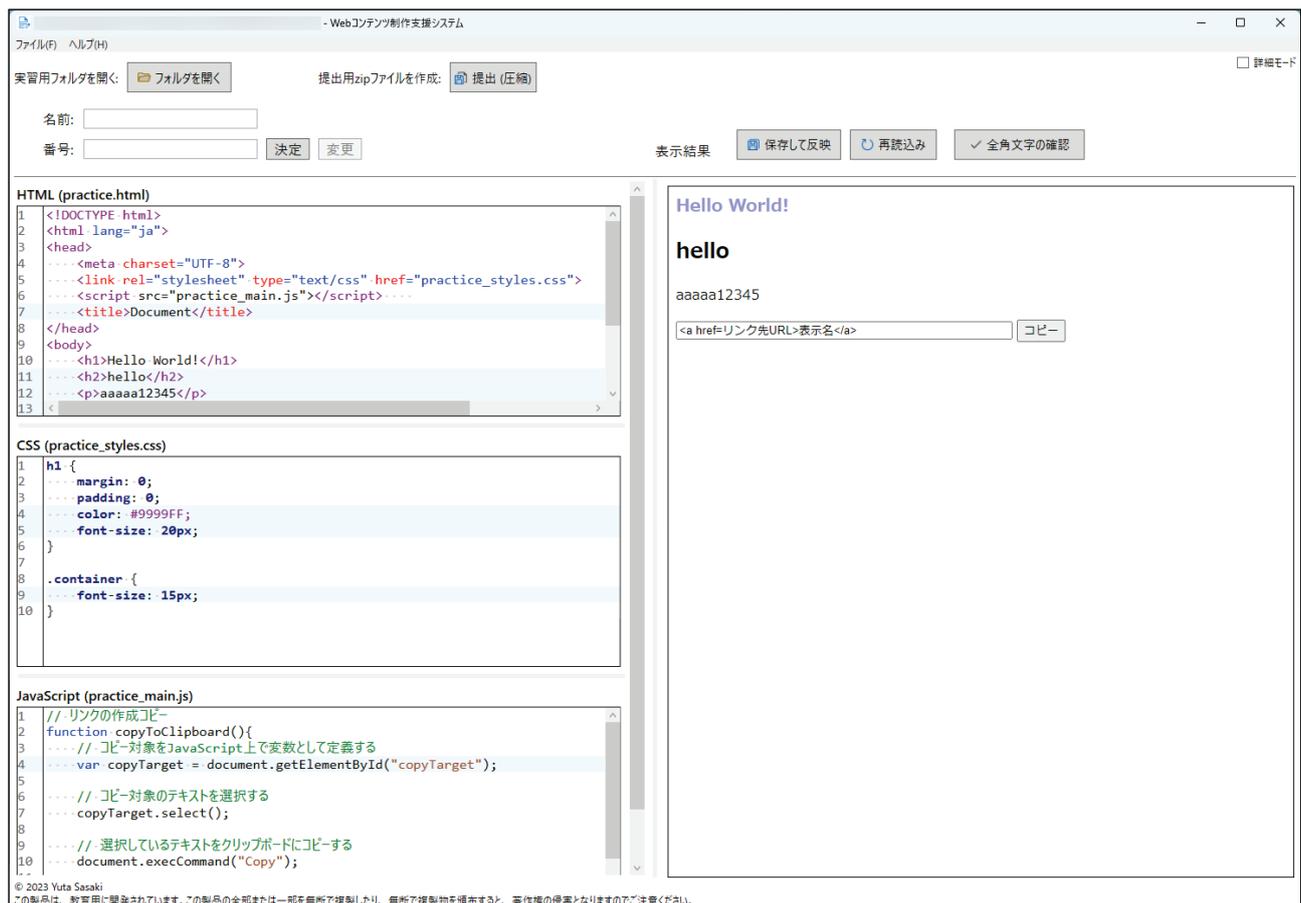


図2 Webコンテンツ制作支援環境の画面レイアウト

表 3 機能の一覧
(a) 基本機能

名称	機能
フォルダを開く	HTM, CSS, JavaScript, 設定ファイルが格納されたフォルダを開く。
名前	名前を入力する。
番号	出席番号等, 生徒を識別する一意の番号を入力する。
決定	名前, 番号の入力を確定し, 編集不可にする。誤操作を防ぐ。
変更	名前, 番号の変更が可能となる。
提出(圧縮)	HTML, CSS, JavaScript, 画像等, 各種ファイルを zip ファイルで圧縮する。名前, 番号がファイル名として使用される。
保存して反映	テキスト編集領域で入力したコードを保存し, Web ブラウザ部分に反映する。
再読み込み	Web ページの再読込を行う。
全角文字の確認	テキスト編集領域に存在する全角文字を検出する。
詳細モード	開発者ツールの表示等, 詳細な設定ができる。チェックを入れるとボタン等が表示される。

(b) 詳細機能

上書き保存	個別に上書き保存ができる。
名前を付けて保存	個別に名前を付けて保存ができる。
ファイルパスを更新	名前を付けて保存を実行し, ファイル名が変更された場合に使用する。
設定ウィンドウを表示	テキストエディタに読み込むファイル名を変更する。
CORS を無効化	CORS エラーを回避するために使用する。
URL	ファイルパス, URL を入力する。表示機能も持つ。
決定	入力した URL にアクセスできる。
開発者ツールを表示	開発者ツールが別ウィンドウで表示される。

表 4 ファイル構成

ファイル名	使用目的
practice.html	Web ページの表示
practice_styles.css	スタイルシート
practice_main.js	プログラムの記述
html_settings.txt	HTML テキストエディタの設定ファイル
css_settings.txt	CSS テキストエディタの設定ファイル
js_settings.txt	JavaScript テキストエディタの設定ファイル

表 3(a)に示した基本機能が利用できる「基本モード」になっている。原則として生徒は基本機能を使って Web コンテンツを制作する。生徒は, ①「フォルダを開く」→②「番号」と「名前」を入力→③制作活動→④「提出(圧縮)」という流れで使用する。

「詳細モード」のチェックボックスにチェックを入れると, 表 3(b)に示した詳細機能が有効になる(図 3)。

主に, 詳細機能は指導者側が利用することを想定している。例えば, 生徒自身で構文エラー等を目視で発見できなかった場合, 開発者ツールを表示させ, 何行目でエラーが発生しているか確認するために利用する。この他にも, ローカル環境で人工知能による画像認識等の処理を行うライブラリ Tensorflow を利用すると CORS (Cross-Origin Resource Sharing)エラーが発生する。CORS エラーを回避するための「CORS を無効化」機能等の利用が可能である。

3.3 ファイル構成

Web コンテンツ制作支援環境を利用する際は, 表 4 に示す 6 つのファイルが必須となる。すべてのファイルは同一フォルダ内に配置する。表 4 に示したファイル名で読み込みが行われるように設定されている。ファイル名の変更は推奨されないが, 変更も可能としている。各ファイルの内容は, 編集可能であり, 授業の内容に合わせて変更できる。

「practice.html」, 「practice_styles.css」, 「practice_main.js」は, 制作する Web コンテンツを構成するファイルである。「html_settings.txt」は, HTML テキストエディタの編集可能領域を設定, 「css_settings.txt」は, CSS テキストエディタの編集可能領域を設定し, 「js_settings.txt」は, JavaScript テキストエディタの編集可能領域を設定するために使用する。

3.4 編集可能領域の設定方法

表 4 の「practice.html」と「html_settings.txt」, 「practice_styles.css」と「css_settings.txt」, 「practice_main.js」と「js_settings.txt」



図 3 詳細モードを有効にした状態

```

practice.html X # practice_styles.css JS ... html_settings.txt X
practice.html > ...
1 <!DOCTYPE html>
2 <html lang="ja">
3 <head>
4 <meta charset="UTF-8">
5 <link rel="stylesheet" type="te
6 <script src="practice_main.js">
7 <title>Document</title>
8 </head>
9 <body>
10 <h1>Hello World!</h1>
11 <h2>hello</h2>
12 <p>aaaaa12345</p>
13 <input id="copyTarget" size="50
14 <button onclick="copyToClipboar
15 <script>
16
17 </script>
18 </body>
19 </html>
20
html_settings.txt
1 false
2 false
3 false
4 false
5 false
6 false
7 true
8 false
9 false
10 false
11 true
12 true
13 true
14 false
15 false
16 false
17 false
18 false
19 false
20 false

```

(a) practice.html と html_settings.txt の行の対応関係

```

practice.html X # practice_styles.css JS ... html_settings.txt X
practice.html > ...
1 <!DOCTYPE html>
2 <html lang="ja">
3 <head>
4 <meta charset="UTF-8">
5 <link rel="stylesheet" type="te
6 <script src="practice_main.js">
7 <title>Document</title>
8 </head>
9 <body>
10 <h1>Hello World!</h1>
11 <h2>hello</h2>
12 <p>aaaaa12345</p>
13 <input id="copyTarget" size="50
14 <button onclick="copyToClipboar
15 <script>
16
17 </script>
18 </body>
19 </html>
20
html_settings.txt
1 true
2 false
3 false
4 false
5 false
6 false
7 true
8 false
9 false
10 false
11 true
12 true
13 true
14 false
15 false
16 false
17 false
18 false
19 false
20 false

```

(a) html_settings.txt の1行目を「true」に変更した状態

HTML (practice.html)	
1	<!DOCTYPE html>
2	<html lang="ja">
3	<head>
4	----<meta charset="UTF-8">
5	----<link rel="stylesheet" type="text/css" href="practic
6	----<script src="practice_main.js"></script>----
7	----<title>Document</title>
8	</head>
9	<body>
10	----<h1>Hello World!</h1>
11	----<h2>hello</h2>
12	----<p>aaaaa12345</p>
13	<

(b) Web コンテンツ制作支援環境による表示
図4 編集可能な行の設定例

HTML (practice.html)	
1	<!DOCTYPE html>
2	<html lang="ja">
3	<head>
4	----<meta charset="UTF-8">
5	----<link rel="stylesheet" type="text/css" href="practic
6	----<script src="practice_main.js"></script>----
7	----<title>Document</title>
8	</head>
9	<body>
10	----<h1>Hello World!</h1>
11	----<h2>hello</h2>
12	----<p>aaaaa12345</p>
13	<

(b) Web コンテンツ制作支援環境による表示
図5 編集可能な行の変更例

txt」は、それぞれ行数を揃えて置く必要がある。

例えば、「practice.html」と「html_settings.txt」は、両方で20行ある(図4(a))。1行目に注目すると、「html_settings.txt」の1行目は「false」になっており、「practice.html」の1行目は編集不可となっている(図4(b))。図5(a)に示すように「html_settings.txt」の1行目を「true」に変更すると「practice.html」の1行目は編集可能になる(図5(b))。CSS, JavaScriptにおいても同様の方法で編集可能領域を設定できる。

4. おわりに

本論文では、情報デザインとプログラミングの各学習内容に対する教育的な配慮に基づきWebコンテンツの制作に要求される仕様を整理した。その結果に基づき設計し、C#と各種ライブラリ等を用いてWindows OSで動作するWebコンテンツ制作支援環境の開発方法とその利用手順について述べた。

本制作支援環境は、2023年10月~11月に徳島県内公立高等学校の「情報I」の授業(のべ12単位時間)で利用できた。今後、他県の高等学校においても利用し、Webコンテンツ制作上の課題を明らかにし、改善していく予定である。

参考文献

- [1] 文部科学省(2019) 高等学校学習指導要領(平成30年告示)解説情報編, 開隆堂.
- [2] 萩谷昌己・他10名(2022) 高校情報I JavaScript, 実教出版, pp.32-41.
- [3] Leaflet, <https://leafletjs.com/> (最終アクセス日:2024年2月1日).
- [4] Tensorflow, <https://www.tensorflow.org/> (最終アクセス日:2024年2月1日).
- [5] AvalonEdit, <http://avalonedit.net/> (最終アクセス日:2024年2月1日).
- [6] Microsoft Edge WebView2, <https://developer.microsoft.com/ja-jp/microsoft-edge/webview2/?form=MA13LH> (最終アクセス日:2024年2月1日).
- [7] Microsoft-WindowsAPICodePack-Core, <http://www.nuget.org/packages/Microsoft-WindowsAPICodePack-Core> (最終アクセス日:2024年2月1日).
- [8] WindowsAPICodePack-Shell, <https://www.nuget.org/packages/Microsoft-WindowsAPICodePack-Shell> (最終アクセス日:2024年2月1日).