

ブロック型エディタを用いた Web システム概念理解の促進

三浦 元喜^{1,a)}

概要：Web システムの構成を理解するためには、動的なページ生成のためのプログラミング言語の知識に加えて、データベースや SQL, HTTP 等の技術に対する知識が必要である。我々はこれまで Web ブラウザ上で動作するエディタを用いて、Web アプリケーション構築の敷居を下げる方法およびシステムを提案してきた。しかしプログラミングをテキスト記述で行う必要があるため、入力ミスが発生したり、どのような機能があるのかが確認しにくいといった問題があった。そこで我々はブロック型のプログラミングエディタを用いて、Web システムの学習と構築が行えるシステム Block Sweetie を開発した。HTML/CSS/PHP/SQLite を用いた簡便な Web システムを、ブロック配置とパラメータ調整によって作成することができる。ドロップダウンメニューの切り替えや、チェックボックスの ON/OFF を含む、ブロックの修正操作は即座に出力に反映され確認できるため、Web システムの構成や概念理解の促進が期待できる。

Learning of Web System Concepts using Block Editor

MOTOKI MIURA^{1,a)}

1. はじめに

平成 29 年 3 月に文部科学省が告示した、新学習指導要領(中学校)の技術・家庭科においては、「コンピュータや情報通信ネットワークなどの情報手段を適切かつ主体的、積極的に活用できるようにするための学習活動を充実する」ことが示され、「生活や社会における問題を、ネットワークを利用した双方向性のあるコンテンツのプログラミングによって解決する活動」が盛り込まれている。我々は、この「ネットワークを利用した双方向性のあるコンテンツのプログラミング」を学習するうえでは、現在利用されているインターネット上のサイトにおける基本的な Web アプリケーションの知識や、構築技術に準拠することが望ましいと考えている。その理由として、実際に利用されている技術の構成や仕組みを知ることで、自分のデータがどのように保存され、取り扱われるかをイメージしやすくなり、Web サービスの選択や利用に対する判断も適切かつ主体的

にできるようになると考えられるからである。

しかしながら、通常 Web アプリケーションの仕組みを知るためには、動的なページ生成のためのプログラミング言語や HTTP の理解が欠かせない。コンテンツを蓄積したり提供したりする Web アプリケーションを構築するには、それらに加えてデータベースやその問い合わせ言語 (SQL) に関する理解も必要となる。そのため、初心者にとって Web システム概念理解の敷居は、Web ページに比べて高い。

我々はこれまでに Web ブラウザ上で動作するテキストエディタを用いて、Web サイトを簡単に構築できるシステム Sweetie Editor[1], [2] や、Web アプリケーションを構築するための Web フレームワーク Sweetie Framework [3], [4] を提案してきた。これらのシステムやフレームワークを用いることで、初学者が Web アプリケーションを開発するうえでの理解を促進したり、試行錯誤を伴う作業の敷居を下げたりすることができる。しかし Sweetie Editor を用いて、Sweetie Framework の機能を利用した Web アプリケーションを構築するには、学習者は PHP のソースコードをテキストで入力したり編集したりする必要があった。そのため学習者によってはテキストエディタおよびブ

¹ 九州工業大学 基礎科学研究系
Faculty of Basic Sciences, Kyushu Institute of Technology
1-1 Sensui, Tobata, Kitakyushu, Fukuoka, 804-8550, Japan

^{a)} miuramo@mns.kyutech.ac.jp

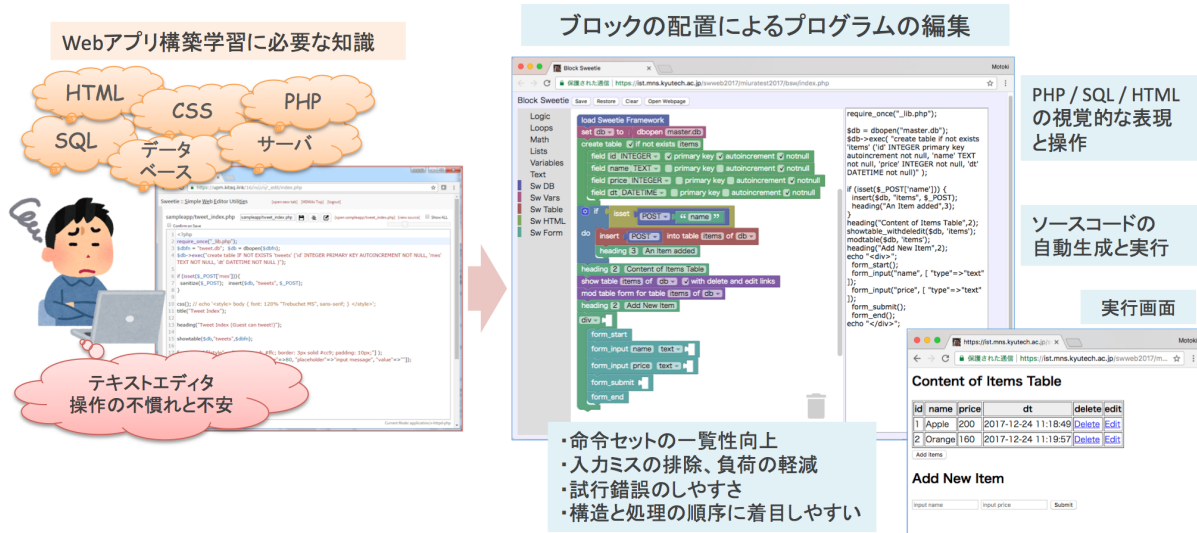


図 1 (左) 従来のテキスト記述による Web アプリケーション編集環境 (右) ブロック記述による Web アプリケーション編集環境

プログラミングに対する不慣れや不安を感じることがあった(図 1 左)。

我々は初学者に対する Web システム概念の理解や、Web システム構築の敷居を下げるため、ブロック型エディタを用いた Web アプリケーション学習システム Block Sweetie を開発した。

2. Block Sweetie

Block Sweetie は Sweetie Framework[4] が提供している関数や頻繁に利用される記述をブロックで表現し、それらを配置することによって、簡易的な Web アプリケーションを簡単に構築できるシステムである。ブロックベースのプログラミングは、初心者が従来のテキストベースの言語を学ぶのに役立つことが知られている [5]。我々はブロックベースのプログラミングを、Web アプリケーション開発に適用した。

2.1 システムの概要

Block Sweetie は基本的に PHP スクリプトの編集機能を提供する。PHP スクリプトには Sweetie Framework の機能を含むことができるので、SQLite データベースとテーブルの作成/管理が可能となっている。図 1 に、Block Sweetie の概要を示す。初学者は開発と学習のために 2 つのブラウザウィンドウを開いて活動する。1 つはブロックの配置によって PHP スクリプトを編集するエディタウィンドウで、もう 1 つは Web アプリケーションの動作を確認するアプリケーションウィンドウである。初学者がブロックを再配置したりパラメータを変更したりすると、Block Sweetie は即座に対応する PHP スクリプトをエディタウィンドウの右側にテキスト形式で出力しつつファイルに書き出す。そ

の直後、自動的にアプリケーションウィンドウを再読み込みしてスクリプトを実行する。したがって、初学者は、ブロックとスクリプトのソースコードとの対応を簡単に確認して確認することができる。

2.2 設計方針とメリット

本節では初学者および教授者に対する、提案システム Block Sweetie の手法設計とメリットについて説明する。

ブロックエディタを導入する主な目的は、初学者が試行錯誤しやすく、最小限の負担でさまざまな Web アプリケーションに関する選択肢を選んで体験できるようにすることである。Web アプリケーションの開発を学ぶ際、学習者は、HTML, CSS, PHP, データベースなどのいくつかの技術間関係を理解する必要がある。その関係を理解するために、学習者は自らの仮説を積極的に構築し、自らの主観的な操作で確認する必要があると考える。

ブロック表記とブロックエディタを Web 開発に導入することで、学習者には以下のメリットがある。

- ・ミスタイプによる構文エラーを排除できる。
- ・プログラム編集に対する不安を和らげることができる。
- ・ブロック一覧があるため、機能を記憶する必要がなく、認知負荷が軽減できる。
- ・関数呼び出しの引数の順序や意味を覚える必要がない。
- ・ブロックによって視覚的に表現されるため、プログラムの構造に着目しやすくなる。

一般にブロックエディタを用いると、テキストベースのプログラミングエディタに比べて、テキストの入力の手間を軽減することができる。また分類されたブロック一覧を提示することで、利用可能な機能を示唆できる。

我々はブロックエディタ上でブロックに変更が加えられ

た際、ソースコードにすぐに変換され、アプリケーションウィンドウに更新が反映されるようにシステムを設計した。この「即時更新」の設計によって、以下の利点が得られる。

- 即時更新/再読み込みによる、試行錯誤と短時間での確認作業を促進する。
- 学習者は、最小限の負担で、Web アプリケーション開発のさまざまな概念や知識を学ぶことができる。

「即時更新」設計を採用したうえで、我々は学習効果を向上させるため「切り替えて違いを観察する」方針に基づき、ブロックのオプションを設計した。たとえば、HTML におけるブロック/インライン要素の違いを理解しやすくするため、`<div>`と`<span>`をドロップダウンメニューによって選択できるようにした。関連して、リストの順序なし/順序ありの差を確認しやすいうように、`<ul>`と`<ol>`を、ドロップダウンメニューに設定した。これらのドロップダウンメニューの選択することで、学習者は選択肢と、その違いについて観察し、気づきやすくなることが期待される。

Sweetie Framework では、見出しのレベルを引数で設定できる `heading()` 関数のような、利便性を高める関数を提供している。しかし抽象度の高い関数がどのような効果をもたらすのかの直感的な理解が難しい場合がある。そのため Sweetie Framework の関数を、標準的な HTML/PHP で記述した場合の表記をエディタウィンドウ右側のソースコード表示エリアで確認しやすくするための「(html)」チェックボックスを導入した。付録の図 A.5 から図 A.10 に示すように、このチェックボックスは、Sweetie Framework ブロックの多くに反映されている。

我々は Sweetie Editor と同様に、Block Sweetie も Web アプリケーションとして構築した。Block Sweetie は Sweetie Editor を提供するとき配布するフォルダの一つとして、組み入れることができる。これまで我々が構築してきた Sweetie Editor では、学習者が自由に個別/グループ作業フォルダを作成したり、削除したりすることができた。そのため Block Sweetie も Sweetie Editor と同じく、学習者が自由に試行錯誤できる仕組みを踏襲している。

我々は Google Blockly[6] を利用して Block Sweetie を開発した。Google Blockly はもともと、PHP スクリプトを出力する機能を提供している。そこで我々は、この出力機能を利用して、ブロック記述から Sweetie Framework 対応ソースコードを生成できるようにした。

3. ブロックとコードのサンプル

本章では、我々が設計した Block Sweetie におけるブロック記述によるコードのサンプルと、その意図について説明する。

我々は以下の 6 つのサンプルを構築した。

- HTML と PHP の基本 (図 2)
- HTML の表とリスト (図 3)

- PHP 連想配列 (図 4)
- HTML フォーム (Get/Post メソッド) (図 5)
- SQLite の導入 (図 6)
- サンプルアプリケーション (おこずかい帳) (図 1 right)

これらのサンプルは、Block Sweetie エディタのサンプル読み込み機能から呼び出し、自由に変更、保存することができる。

3.1 HTML と PHP の基本

図 2 に、HTML と PHP の基本サンプルを示す。このサンプルでは、PHP の `echo`, `date`, `random` 関数に加え、Sweetie Framework が提供する `title()`, `heading()`, および `br()` 関数を使用している。最初に、load Sweetie Framework ブロックを配置して、Sweetie Framework 関数を利用可能にしている。また、乱数を表示することによって、動的なコンテンツの概念理解を促進する。2.2 で述べたように、Sweetie Framework 関数は「(html)」チェックボックスを備えている。このチェックボックスが ON になっていると、関数の機能が展開された状態で、ソースコード部に表示される。たとえば、`heading(2, "...")` ブロックは、`<h2>...</h2>` を `echo` で表示したのと同じであることを示している。`<div>` および `<span>` ブロックは、引数として PHP の連想配列をとることによって、属性 (attribute) を変更できる。この部分を変更することによって、学習者は CSS によるスタイル指定を確認できる。出力例を付録の図 A.1 に示す。

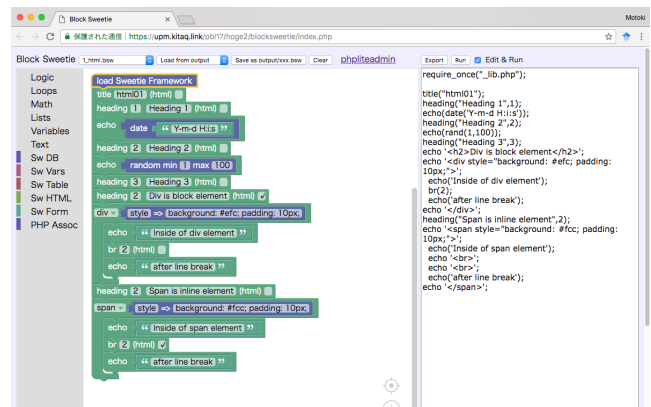


図 2 HTML と PHP の基本サンプル

3.2 HTML の表とリスト

図 3 に HTML の表とリストのサンプルを示す。このサンプルは、静的なコンテンツの例であるが、学習者に HTML での表とリストの表現方法をブロックの入れ子で視覚的に見せることによって、構成方法を理解してもらうことを目的としている。加えて、`<ul>` と `<ol>` の違いや、`<th>` と `<td>` の違いを、ドロップダウンメニューの切り替

えによって簡単に確認することができる。出力例を付録の図 A.2 に示す。

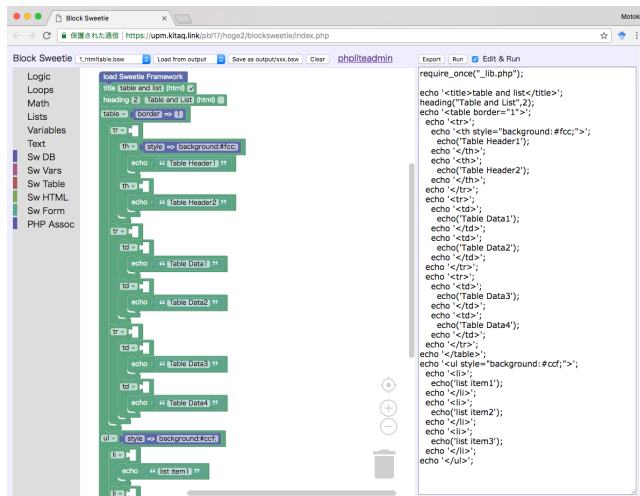


図 3 HTML の表とリストのサンプル

3.3 PHP 連想配列

図 4 に PHP の連想配列サンプルを示す。連想配列の概念について理解してもらうために、具体的な連想配列の作成や追加例を示している。また、表示確認の方法を 2 種類提示している。とくに foreach 構文によって、連想配列データを順番に取り出しながら（変数に割り当てながら）処理をする例を示すことによって、配列データの典型的な扱い方についての理解を促進できると考えられる。出力例を付録の図 A.3 に示す。

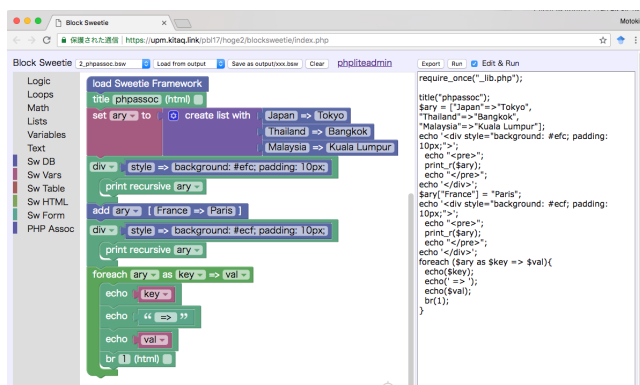


図 4 PHP の連想配列サンプル

3.4 HTML フォーム (Get/Post メソッド)

図 5 にシンプルなフォームの例を示す。このスクリプトは GET および POST メソッドで送信されたデータを表示することができる。学習者は、GET メソッドと POST メソッドの違いを、[form.start] ブロックの属性を変更したうえで、アプリケーションウィンドウから実際にデータを送信することによって、理解することができる。[form.input]

ブロックは、type 属性に text, password, file, check-box, radio, および range を設定できる。また、連想配列で属性を追加することも可能である。フォームタグ関連のブロックも Sweetie Framework の関数呼び出しとして抽象度の高い状態で表現されているが、「(html)」にチェックを入れることによって、関数の具体的な出力をソースコードエリアに展開して表示することができる。出力例を付録の図 A.4 に示す。

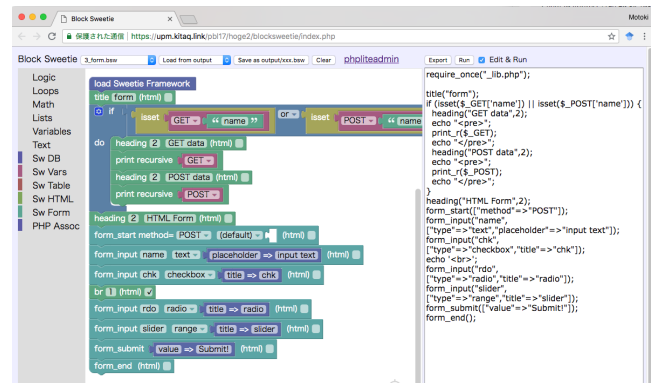


図 5 HTML フォーム (Get/Post メソッド) のサンプル

3.5 SQLite の基本

図 6 に SQLite を利用したデータベースのテーブル作成や、データの挿入の例を示す。我々は、テーブル定義を行うため SQL に相当する [create table] および [field] ブロックを設計した。学習者は、field ブロックを追加することによって簡単にフィールドを再定義できる。また、[if not exists] チェックボックスを OFF にするだけで、テーブルをリセットすることができる。[field] ブロックには、SQLite で利用可能な integer, text, datetime, real, blob, image をドロップダウンで選択できる。

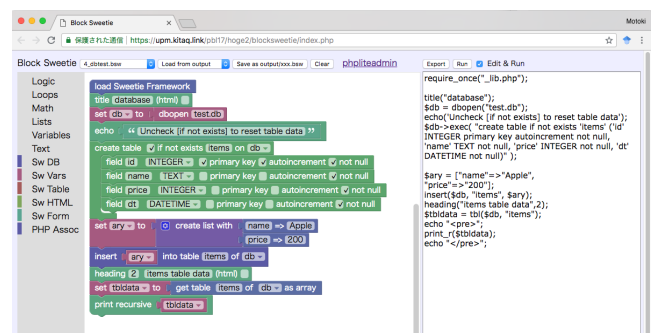


図 6 SQLite の基本サンプル

以前の Sweetie Editor では、学習者は SQL のテキスト記述を直接書き換えてテーブルを定義する必要があった。この部分をブロック記述に置き換えることによって、編集時のミスを防止することができる。加えて、primary key, auto-increment, not null といった属性をチェックボック

スで簡単に設定できる。これらの工夫により、学習者はテキスト記述のミスに悩まされることなく、テーブル定義記述とその意味についての理解に注力することができる。

ちなみに、学習者は画面上部の [phpliteadmin] リンクをクリックすることによって、管理ツールのひとつである phpLiteAdmin [7] を利用して、データベースのテーブル定義やデータの確認作業を行うこともできる。

3.6 サンプルアプリケーション (おこずかい帳)

図 1 右に表示しているブロックは、これまでのサンプルで示した内容を利用した具体例 (おこずかい帳) のプログラムである。学習者はこのプログラムに改良を加えていくことによって、Web システムやアプリケーションに対する知識や技術を確認し、理解を深めることができる。

`showtable()` は、データベースのテーブルデータをすべて表示する Sweetie Framework の関数である。`modtable()` は、Javascript のコードを挿入し、テーブルデータの変更を容易にするための関数である。これらの関数は Web アプリケーション管理画面を簡易的に作成するのに適している。しかしながら、現状ではこれらの関数の内部構造を学習者が理解できるような構成にはなっていない。将来的な課題や改善の方向性としては、これらの関数定義の中身についてもブロックで表現することによって、理解を深めたい学習者が気軽に参照したり、定義を修正したりできるようにすることが必要と考えている。

4. 予備実験講義

Block Sweetie に対する客観的なコメントを得るため、我々は予備実験講義を行った。大学学部 3 年生 5 名に、40 分の講義をし、その後 10 分程度でアンケートに回答してもらった。今回講義を受けた学生は Web アプリケーションの開発に興味を持っており、そのうち 3 名はすでに HTML と Ruby on Rails の経験があった。

講義では、最初に、ブロック画面のズームの方法や、ブロックの有効化/無効化の方法など、Google Blockly が提供する機能について説明した。その後、3 節で説明したサンプルを順番に見せて説明した。また、学生にも個人用の PC と、Block Sweetie 操作環境を提供した。説明の後、以下のアンケート項目に自由記述で回答してもらった。学生は Web のフォームから入力して回答した。

- (1) PHP の配列について、今回わかったこと、学んだことを書いてください。
- (2) HTML のフォームと、PHP でのフォームデータ処理について、今回わかったこと、学んだことを書いてください。
- (3) データベース (sqlite) とその利用方法について、今回わかったこと、学んだことを書いてください。
- (4) Web アプリケーションの構成と、PHP による構築方

法について、今回わかったこと、学んだことを書いてください。(Rails など、他の構築方法と比較してもよいです)

- (5) 今回のツールについて、良い点と、悪い点 (改善点) を、なるべくたくさん挙げてください。

4.1 アンケート結果

最初の質問 (1) に対する、学生の回答を以下に示す。

- キーとバリューを入れることができること
- Ruby ではハッシュ, Python ではディクショナリと呼ばれるようなデータ構造で、PHP にもそういった構造を持つデータを扱えることを知った。インデックス呼び出しと比べ、可読性の高さを感じた。
- rails の書き方によりやく慣れてきたと感じ始めていましたが、php はまたさらにややこしい書き方だなと感じました。
- ruby より見づらさはあったものの、ブロックの内容を変えて、サブミットを行うことで、直ちに生成されるものだ、と感じた。
- データを一行で一括して表示することができる (注: `show_table()` 関数)。

質問 (2) に対する、学生の回答を以下に示す。

- GET と POST については、「データを送信するときは POST を使う」程度の認識であったが、実際どういった挙動の違いを見せるのかを知ることができた。
- GET と POST の違い
- form を GET で送るか、POST で送るか、の違いがある。
- データの受け取りに GET と POST があり、GET では保存可能なサイズが小さい。
- rails でもそうだが、POST と GET の違いを深く理解していなかったので今回の講義はためになったとおもいます。

質問 (3) に対する、学生の回答を以下に示す。

- null や primary など、rails で `db/migrate/*` のファイルに書かれるような記述が多く見られ、データベースとのやり取りが見られた。SQL についての知識は皆無なので、その辺りも勉強しようと思った。
- テーブルとして管理していること
- 何らかの処理を行うことで、データベースが生成される。
- ひとつのファイルをデータベースとして保存できる
- php も rails と同じく sqlite を使用しているので、php 視点だとこんなふうにするのかというようなことが学べた。

質問 (4) に対する、学生の回答を以下に示す。

- どちらも html や javascript のコードを生成するフレームワークであるが、Rails が Model, View, Controller

の3つに切り分けて動作を記述するのに対し、今回学んだフレームワークはページごとに記述するため、直感的で理解することができた。さらにhtmlとの比較が出来るので、初学者の学習にも有効であると感じた。

- ruby 以外の言語 (PHP) で Web アプリを作ることができることを学んだ。
- PHP はブロックで処理されたものがそのままプログラムにリンクしている。Rails はプログラムのみ動かして、ものごとを考えている。その点が違うと思う。
- 処理の区切り目が Rails より見分けにくかった。違いとしては、変数の置き方、関数名の違いぐらいだと感じた。
- rails も php もアクション、ビュー、データベースの基本要素から構築していくのでなれるのに時間はかかると思うが、やり方は一緒だなと感じた。

質問 (5) に対する、学生の回答を以下に示す。

良い点

- 非常に直感的に動作を記述することが出来る。html の学習にも使える。
- 構造の理解が容易。
- 簡単そう。
- センテンスごとにブロック化されているのでデバッグしやすいかなと感じた。

悪い点 (改善点)

- コード (ブロックが) 肥大化しやすく、それに伴い可読性やメンテナンス性が低下しやすい (vim にあるような折りたたみ機能があると解消されると思われる)。
- 慣れてくると、テキストのほうが楽かもしれない。
- (ブロックの) プログラムが見づらい。
- 深く理解していなくても、(とりあえず) 動くようなプログラムが書けそうだなと感じた。

4.2 議論

講義の時間は1時間程度と比較的短時間であったが、ほとんどの学生がHTML, CSS, PHP, SQLiteの意味や機能を認識していることが確認できた。ただし、今回は予備実験講義であり、3名の学生がすでにHTMLとRuby on Railsの経験を持っていたことを考慮する必要がある。彼らはBlock Sweetieのサンプルコードを、既存の知識とスキルに関連付けて、比較することができた。また、今回の講義で、学生はGET/POSTメソッドの違いについて理解できた。

今後はWeb開発に関する知識やスキルを持たない学生に対して講義を行い、提案手法の効果を検証していきたい。

5. 関連研究およびシステム

ブロックインターフェースをプログラミングの導入に適用する効果は[5]でも述べられており、よく知られてい

る。また、初学者用のプログラミングのためのブロックエディタインターフェースを採用する多くのソフトウェアツールが存在する。たとえば、Scratch[8]、Google Blockly [6]、Pencil Code[9]などは、初学者プログラマ用の教育ツールとして有名である。一般に、これらの初学者向け画面上でキャラクタを動作させたり、音を鳴らしたり、マウスやキーボードに対するインタラクティブな反応を記述するものが多いが、ロボットや物理的なデバイスを含めたプログラミングを行う例もある。たとえば、Arakliotisらは、Arduinoプログラミング[10]にブロックインターフェースを適用している。我々はWebシステム・アプリケーション開発という特殊な学習対象について、ブロックインターフェースを適用したシステムを構築した。

グラフィカルな操作のみでWebアプリケーションを簡単に構築できるWebサービスとして、Bubble [11]がある。ユーザはインターフェースビルダを用いて、画面上に入力フォームやボタンを配置したり、それらに関連付けることによって開発する。GUI画面による設定からWebアプリケーションを自動生成するツールとして、Web Performer[12]がある。Eclipse IDEのプラグインとして構築されており、本格的なWebアプリケーションを開発する際の省力化をはかることができる。これらのツールやWebサービスは、簡単な操作でWebアプリケーションを作成する点において本研究と類似している。しかし、BubbleやWeb Performerはソースコードを隠蔽しており、プログラミングを意識させない点が本研究とは異なる。本研究では、ブロック記述とコード記述を見比べることによって、初学者がWebシステムの基本的な概念を理解しつつ、プログラミング技術やデータベースの知識を学習することに主眼を置いている。

長瀧らは、リレーショナルデータベースの演習をWebブラウザから対話的に行うことを目的としたオンライン学習教材sAccess[13]を開発している。sAccessはブロックベースのクエリ構築インターフェースを提供しており、複雑なSQL文法を理解しなくてもデータの操作を体験できるという点において、本研究と類似している。Block SweetieはsAccessに比べて、HTMLによるWebアプリケーション構築という、より広い内容を扱っている。

小林らは、sAccessをベースに、SQLエディタやPHPによるWebアプリケーション開発エディタ(PHPエディタ)を構築し、高等学校での授業利用を行っている[14]。PHPエディタは、Sweetieと同様、テキストベースのWebプログラミングエディタであるため、ブロックインターフェースは備えていない点が本研究と異なる。

6. さいごに

「ネットワークを利用した双方向性のあるコンテンツのプログラミング」の一例として、ブロック型エディタを用いたWebアプリケーション学習システムBlock Sweetieを開

発した。HTML/CSS/PHP/SQLite を用いた簡便な Web システムを、ブロック配置によって作成することができる。ドロップダウンメニューの切り替えや、チェックボックスの ON/OFF を含む、ブロックの修正操作は即座に出力に反映され確認できるため、Web システム概念の理解に役立つと考えられる。

予備実験的な講義を通して、Block Sweetie が HTTP の GET/POST メソッドの違いを理解するのに有効であることを確認できた。また、データベースの概要を含む、Web アプリケーション開発技術の導入説明について短時間で伝えることを確認した。

今後、実際に初学者に利用してもらい、Block Sweetie の効果や有用性を確認する。また、中学生が利用しやすいように、ブロックの表記を日本語に変更することを検討している。さらに、サンプルコードやブロックの追加によって、セッションを用いた認証やログイン機能についても対応していきたいと考えている。

Block Sweetie は以下の URL でダウンロードできる。
<https://github.com/miuramo/blocksweetie>

参考文献

- [1] 三浦元喜. Sweetie: 学生計算機上で動作する協働作業に適した軽量 Web エディタ環境. 情報処理学会情報教育シンポジウム (SSS2015), pp. 117–123, August 2015. <http://id.nii.ac.jp/1001/00144690/>.
- [2] Motoki Miura. Lightweight Web Authoring Environment. In *20th International Conference on Knowledge-Based and Intelligent Information & Engineering Systems (KES2016)*, Vol. 96, pp. 887–895, September 2016.
- [3] 三浦元喜. お手軽 Web アプリ開発環境 Sweetie Framework. 情報処理学会インタラクシオン 2017, pp. 149–153, March 2017.
- [4] Motoki Miura. Sweetie Framework: Simple but Practical Web Application Development Environment. In *12th International Conference on Knowledge, Information and Creativity Support System*, pp. 149–154, November 2017.
- [5] David Bau, Jeff Gray, Caitlin Kelleher, Josh Sheldon, and Franklyn Turbak. Learnable programming: blocks and beyond. *Communications of the ACM*, Vol. 60, No. 6, pp. 72–80, 2017.
- [6] Neil Fraser. Ten Things We’ve Learned from Blockly. In *2015 IEEE Blocks and Beyond Workshop (Blocks and Beyond)*, pp. 49–50, October 2015.
- [7] phpLiteAdmin: The Web-based Database Management Tool for SQLite. <http://www.phpliteadmin.org/>. (Visited: Sep 24, 2018.).
- [8] Mitchel Resnick, John Maloney, Andrés Monroy-Hernández, Natalie Rusk, Evelyn Eastmond, Karen Brennan, Amon Millner, Eric Rosenbaum, Jay Silver, Brian Silverman, et al. Scratch: programming for all. *Communications of the ACM*, Vol. 52, No. 11, pp. 60–67, 2009.
- [9] David Bau, D Anthony Bau, Mathew Dawson, and C Pickens. Pencil code: block code for a text world. In *Proceedings of the 14th International Conference on Interaction Design and Children*, pp. 445–448. ACM, 2015.
- [10] S Arakliotis, DG Nikolos, and E Kalligeros. Lawris: A rule-based arduino programming system for young students. In *2016 5th International Conference on Modern Circuits and Systems Technologies (MOCASST)*, pp. 1–4. IEEE, 2016.
- [11] Bubble Group Inc. Bubble: Visual Programming. <https://bubble.is/>. (Visited: Sep 24, 2018.).
- [12] キヤノン IT ソリューションズ. 超高速開発ツール Web Performer. https://www.canon-its.co.jp/products/web_performer/. (Visited: Sep 24, 2018.).
- [13] 長瀧寛之, 中野由章, 野部緑, 兼宗進. データベース操作の学習が可能なオンライン学習教材の提案. 情報処理学会論文誌, Vol. 55, pp. 2–15, 2014.
- [14] 小林史弥, 西川弘恭, 林康平, 島袋舞子, 長瀧寛之, 兼宗進. 情報システム学習を指向した Web ベースのツール開発と授業利用の報告. 情報処理学会研究報告. コンピュータと教育研究会報告 2016-CE-134(3), 1-8, 2016-03-05.

付 録

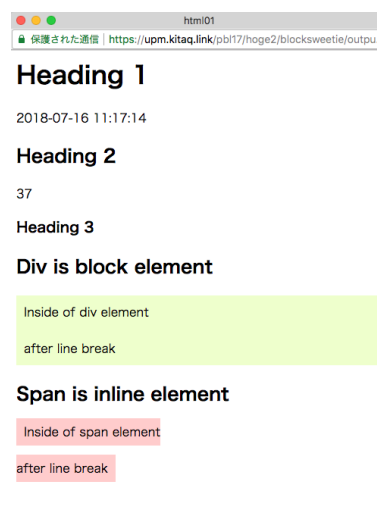


図 A.1 図 2: 「HTML と PHP の基本」の出力

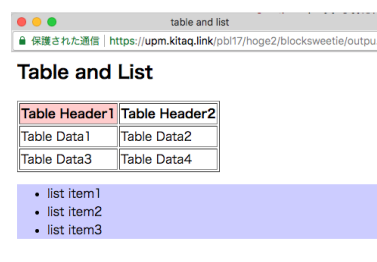


図 A.2 図 3: 「HTML 表とリスト」の出力

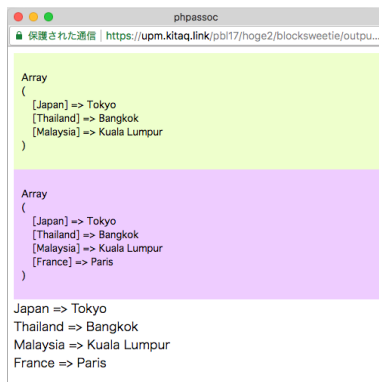


図 A-3 図 4: 「PHP 連想配列」の出力

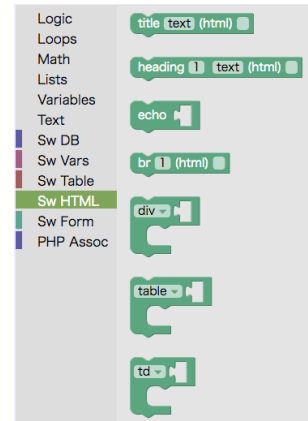


図 A-8 HTML 要素

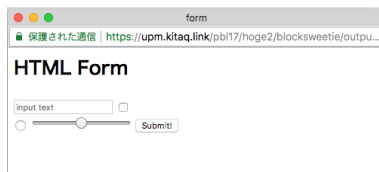


図 A-4 図 5: 「HTML フォーム」の出力

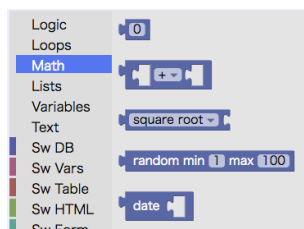


図 A-5 数学関数

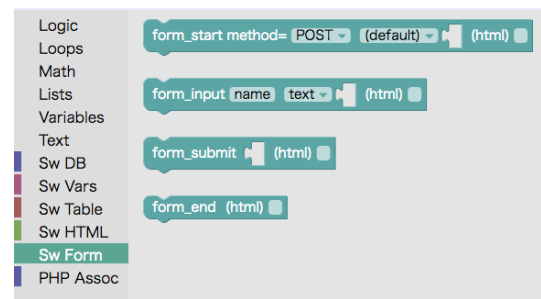


図 A-9 Form 要素

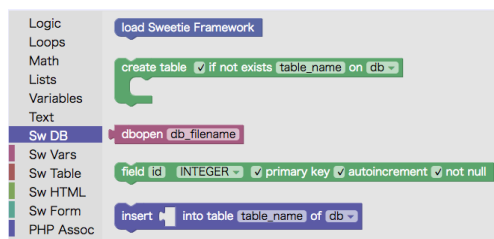


図 A-6 データベース操作

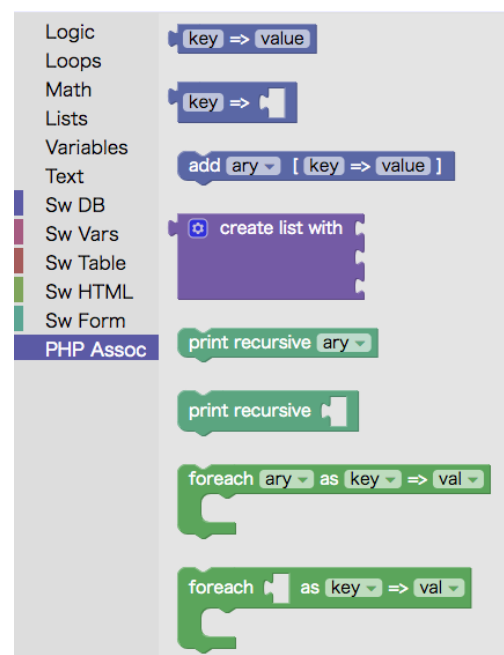


図 A-10 PHP 連想配列

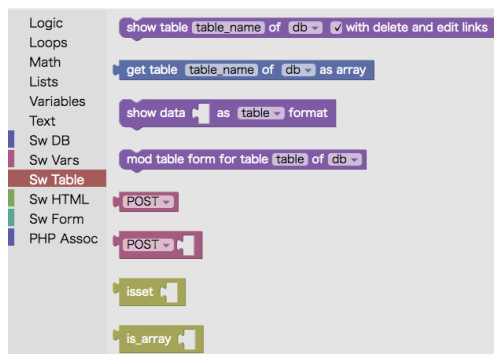


図 A-7 テーブル出力