

Nodeを用いたプログラミング授業支援システムの開発について

松永 豊

情報教育講座

About Development of Programming Class Support System by the Node

Yutaka MATSUNAGA

Department of Information Sciences, Aichi University of Education, Kariya 448-8542, Japan

1. はじめに

現代は高度情報化社会であり、コンピュータに触れる機会は格段に増えた。しかし、それでも高校までにプログラミングを学ぶ学生は極めて稀である。大学で初めてプログラミングを学ぶ学生が大半であるということは、ほぼ全員がほぼ同じスタートラインに立っていることになる。しかしながら、プログラミング授業は数ある授業科目の中でも学生間で差が付きやすい代表的な授業であることは疑いようがない。プログラミング教育は他の科目と比べて何が難しいのであろうか。その答えは簡単には求められないが、教育改善のやりがいがある科目であるとも言える。

プログラミング教育において演習は極めて重要である。一般に、プログラミング言語の習得には多くの時間がかかり、その大半は演習に費やすことになる。基本的に演習は各自で行う必要があるが、高校までにあまり触れたことのない分野であるためか、どこから初めてよいかかわからないといった自学習が困難な学生も多かれ少なかれ発生する。すなわち、膨大な演習の提供、および、学習進捗のチェックが極めて重要であると言える。

そこで、まず、プログラミング教育における授業支援システムや授業支援教材においてどのようなものが要求されるかについて考えてみる。

2. 授業支援システムに求められるもの

プログラミング科目に限らず一般の授業においても、演習問題を提示するタイミングをコントロールすることは多い。例えば、講義→演習という流れの授業において、敢えて演習開始時に演習課題用紙を配布する場合などがこれに当たる。講義前に配布してしまうとすぐに問題を解こうとするものが現れ、講義を真剣

に聞かない、間違った方法で解く、などの弊害が起こる可能性があるためである。

また、授業時の発話とスライドを同期させることも多い。プレゼンテーション用ソフトには普通に付いている機能ではあるが、クリックするタイミングで文字を表示させるなどは発話とスライドの同期が目的と言えるだろう。特に一定のストーリー性を持たせた授業を構築する場合は重要となる。

すなわち、一定範囲の時系列的処理が必要と言いつけることができる。講義内容をスライドとしてスクリーンに表示させるだけなら一般的なプレゼンテーションソフトを用いれば十分である。しかしながらパソコンを直接操作する授業においては、ネットワーク経由で各自のパソコン上に映すことも有効である。なぜなら、やり方によっては操作時間を大幅に短縮することができる場合があるからである。例えば、長いURLをスライドに映して各自に入力させるとかなりの時間がかかる。URLを入力させること自体が演習目的ではないのなら時間の無駄である。一方、目の前の自分のパソコン上に表示されている授業資料内のリンクをクリックするだけなら一瞬で済むため、大幅な時間短縮につながる。これを実現した授業としてWikiを用いた授業が挙げられる。例えばPukiWiki [1]を用い、授業の流れに従い情報を小出しにアップしていけばよい。あらかじめアナウンスすることが決まっているならば、コメントアウトを外す操作だけで充分である。無論、教師から生徒への一方通行ならただのWEBページでも十分だが、教師がブラウザ上から操作できることに加え、将来、双方向通信をさせたい場合にすぐ対応できるためWikiを使うメリットは大きい。逆に生徒からの書き込みを禁止したい場合はページ書き込み権限を教師のみにすれば済む。また、PukiWiki独自の整形ルールはHTMLよりも単純なため、とっさの書き込み等も容易であり、アドリブ的な授業展開にも

対応可能である。さらには、授業後に一定期間公開すれば講義ノートとしての役割を果たすことにもなる。このように一見いいこと尽くめではあるが、汎用のシステムならではの欠点がないわけでもない。例えば、折角双方向の通信機能があるわけだから是非とも生徒側からの意見の収集やレポート提出などにも対応させたい。できないわけではないのだが、ファイル名に学籍番号を付けさせるなど一定のルールを決めないと管理が大変になる。また、たとえルールを決めたとしてもユーザがそれを守らなかったり全角半角をミスしたり打ち間違えたりしてルールから外れると管理が大変になる。そのためユーザIDを自動的に付加させるなどの機能が欲しくなり、後述のようなログイン管理システムが必要となる。

次にレポート提出用支援ツールについて考えてみる。前述の通り、プログラミングにおいて演習は極めて重要である。細かい演習課題を数多くこなすことで基礎が身に付き、応用も可能となる。すなわち、一週間に1つなど提出課題が極めて多いのが特徴である。このようなケースにおいては、管理が極めて重要となる。具体的にはユーザIDとの関連付けが重要となる。また、過去の自分の提出状況や提出物に対するコメントなどが各個人で把握できるシステムであればなおよい。これを実現する最もわかりやすい方法はログイン管理システムである。学生自身に学生番号等を付けてもらうやり方は教育的にはそれなりに意味を見出すことも可能ではあるが管理という意味ではやや不安要素が残る。ログイン、パスワードで本人認証した後、自動的にユーザ情報を付加するほうが確実である。

以上のことを踏まえ、筆者は過去さまざまなプログラミング授業支援ツールを作成してきた [2] [3]。一言でいえばPukiWikiをベースとした改造システムなのだが、実際の授業に合わせてチューンナップを繰り返してきたため相当きめ細やかな対応ができるように工夫されている。大雑把な機能としては

- 授業内容のアナウンス
- 自動出欠記録
- レポート提出システム
- レポートコメント機能
- 提出レポート保管庫
- 面接予約システム
- 電子カルテ
- プログラミング授業間の連動
- 教員間スケジュールの連動

などが挙げられる。無論、ログイン型のシステムであるため提出ファイルに自動的にユーザ情報が付加されるなどファイル管理も楽である。レポートにコメントも付けられるため、学生の進捗チェックにも役に立つ

し、きめ細やかな指導にも繋がる。

このシステムの作成は、PukiWikiにログイン機能とファイル提出機能を付加させるところから始まった。両方ともPukiWikiに簡易的な機能は搭載されていたが、ユーザ管理がしやすくなるように独自の改造を施したのがきっかけである。その後、主に筆者自身が担当する授業が進めやすくなるように次々と改造し、チューンナップを繰り返した。現在、いくつかの授業で実際に稼働中であり、効果も得られているのだが、このシステムにもやや不満が生まれてきた。改造を繰り返してきたため、やや継ぎはぎ感が高まっていることに加え、そもそもPHP [4] ベースであることも原因かもしれない。PHPはサーバサイド動作するWebアプリケーションの開発に極めて適した言語であり、ユーザ数も非常に多い。筆者も好んでPHPベースのWebアプリケーションを作成し、起動管理している。しかしながらPHPは原則サーバサイドでの操作のみを対象とするため、クライアント側で複雑な動作をさせるためには別の仕組みと融合しなくてはならない。最もポピュラーな選択肢としてJavaScriptが挙げられ、実際、JavaScriptとの連携を前提としているといっても過言ではない。クライアント側としてプラグインまで含めればJava、Flash、Silverlightなど選択肢は広がるが、ユーザにプラグインを強要することがどうしてもネックになる。また、OSやブラウザによってはプラグイン開発が行われないケースもある。一方、JavaScriptはブラウザへの搭載の歴史は長く、ブラウザによって性能差があるものの全く無視されるということはない。また、近年、Ajaxが浸透し、jQuery [5] などクライアント側の動作をより豊かにする仕組みが確立したため、クライアント側の開発言語としてJavaScriptを選択することは極めて妥当と考えられる。しかしながら、PHPとJavaScriptは必ずしも相性が良いものではない。基本的にはPHPのprint命令（あるいはそれに準ずる命令）でJavaScriptのソースコードを直接吐き出しているだけであり、無論、言語のルールも異なるので開発の際には2つの言語を意識しながら作ることになる。

筆者の開発した授業新システムも基本的にはPHPとJavaScriptで構成されているが、最低限必要な仕組みとしては主にサーバサイドの処理が多いシステムであったためさほど問題にはならなかった。しかしながら、ある程度システムが成熟してくると最低限の機能だけでなく使い勝手の良さまで気を使う余裕が出てくる。ユーティリティに関してはPHPが中心で構成されているが、ユーザビリティに関してはJavaScriptが中心になるため、開発能率の低下が著しく感じたのである。結局これは、「どのプログラミング言語が優れているか」の論争に近いものがあり、あまり意味がないものだと考えながら改造するしかなかった。しかし、

そこに登場したのがNode (Node.js) である [6]。

3. Node について

Nodeはその特徴をまず一言でいうなら非常にシンプルながことが挙げられる [7]。Nodeの優れた部分は多々あるが、今回のテーマにおいて最も重要なポイントは「Nodeはサーバサイドで起動するJavaScriptである」に尽きる。実はサーバサイドのJavaScriptは1990年代から存在していたが、主流ではなかった。Nodeで初めて陽の目を見たということになるが、これは極めて画期的なことである。

先にも述べたようにクライアント側でJavaScriptを選択することは極めてリーズナブルである。jQueryなどを駆使すればユーザビリティの向上はかなり期待できる。そして、サーバサイドでも同じ言語でコントロールできるわけだから、開発効率も飛躍的に向上することは間違いない。

4. Nodeを用いた授業支援システム

すでに2章でも述べたように、既存のツールをそのまま使ってもある程度の授業支援は可能である。それにも拘わらずなぜ改造を施すか。結局は個別のチューニングであり主観的価値が大きく作用している。例えば、筆者は過去他の先生が作った（正確には卒業研究で学生が作った）面接予約システムを性能評価も兼ねて使ったことが何度かあるが、自分であればザックリ作る部分を非常に拘って作ってあったり、逆に自分なら細かく配慮する部分が抜けていたり興味対象がかなり違うと感じた。恐らく逆も同じことが言えると考えている。筆者が作ったシステムは何人かの先生に使ってもらっているが、ユーティリティ部分に関しては便利だと一定の評価をもらっているものの、ユーザビリティに関しては筆者が特に感じない部分にも不満があるかもしれない。つまり、誰もが納得できる完璧なシステムを提供することは容易ではない。（あるいは絶対に無理なのかもしれない）

不満を感じたら即プログラムの改造（即スクリプト等の生成）という選択は必ずしも好ましいものではないが、今の時代、プログラミング言語に触れる機会も多く、一定範囲のプログラミングが可能な教育者も多いとみなして、授業支援システムの開発を題材とし、プログラマ目線で述べることにする。

まず、今回の目標であるが、差し当たり既存のプログラミング授業支援システムに匹敵するシステムをNodeベースで一から作ることにする。既存のシステムの詳細は参考文献 [2] [3] を参照されたい。無論、最終目標はユーザビリティの向上だが、実質上の移植作業さえ完了してしまえば元システムより改造が容易

になることが見込めるので、今回はそれほど拘らないことにする。筆者自身の経験則を踏まえ、拘りたい部分は以下の通りである。

- ログイン方式のシステムとする
- いくつか権限を用意し、ログイン時にユーザ名から判別して与えるものとする
- 一定の条件をクリアするページはログイン前にもアクセスできるようにする
- 原則、ペーパーレスが実現するように工夫する
- 教員（管理者）はWeb上から一定範囲のページを編集できるようにする
- この時、HTMLより簡易なテキストであることが望ましい
- 各学生とも、ログイン後、自身のレポート提出状況や教員のコメント等が確認できる

まず、Nods.jsのExpressを利用してサーバのベースを構築した。今後の改造を鑑みて、sessionsとcssのStylusを有効にしている。

テンプレートとしてはJadeを採用した。実は筆者はJadeを初めて使うのだが、比較的シンプルなテキストになるため、これを直接ソースデータしてWeb上から書き換えが可能なシステムとすることにした。JadeはPukiWikiのページソースほどではないが、HTMLよりもシンプルなテキストではある。テンプレートも使用できるので、埋め込み用の部分だけでも良いのかもしれないが、とりあえずJadeファイル全体を対象とした (図1)。

Nodeサーバは独自のイベントドリブンループを用いてWebコンテンツを配信する。Expressの場合はExpressディレクトリのトップにあるapp.jsをnode命令で実行すればよい。app.js内ではExpressの各種設定、指定メソッド (getやpostなど)、指定ルートのコールバック設定などを行っている。ここでは実験的にNodeサーバを使用するため、もともと動いているHTTPサーバ (apache) を停止するわけにはいかない。



図1 Jade ファイルの編集

すなわち port 番号の指定が必要となる。実際に利用する場合は apache と同時起動するならば、トップページだけ apache で受取り、すぐに port 指定をした Node サーバにリンクすることにすれば port 番号は特に意識する必要はない。ただし、混乱を招く恐れがあるので、以後は Node サーバのトップページをトップページと呼ぶことにする。

まず、初めてトップページに訪れた際には、index.jade が示すページを表示することになる (図2)。このページでは主にシステムの説明とログインページへのアナウンスのみを行っている。なお、index、および、ページ名が ENT_ で始まるページは、本サーバへのログインが完了していなくても表示が可能なページとして設定してある。

次にログインページだが、このページもログイン前のユーザにも表示しなければならない。先ほど説明したルールに従い、ENT_login.jade とした (図3)。ちなみにログイン前に ENT_ で始まらないページにアクセスした場合は、すべて index ページが表示されるようになっている。

UserID、および、Password を入力するとサーバにデータが post され、ユーザ判定が行われる。この作業は Express ディレクトリの中の routes ディレクト

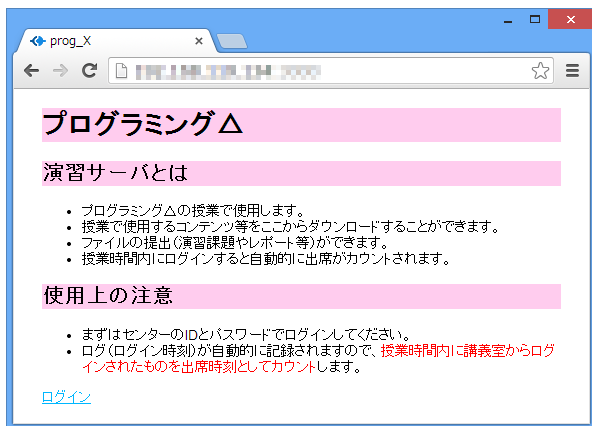


図2 初めて訪れた場合

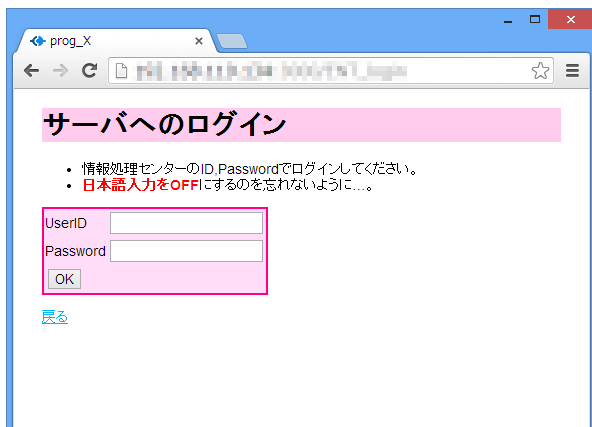


図3 ログイン画面

リ内のプログラムで行っている。ちなみに、Express では app.js から呼び出された各ルートのコールバックは基本的には routes ディレクトリ内に収めることになっており、index.js, login.js など、これらはすべて JavaScript ファイルになっている。具体的にはユーザ ID (必要に応じて正規表現が利用可能) とパスワードを確認し、認証が取れなかった場合は失敗メッセージを表示して再度やり直し、成功した場合はセッションの設定を行った後トップページへのジャンプを行っている。

無事、ログインが成功するとセッション情報が正しく設定されるため、トップページが index ではなく授業ページのトップページに変更され (図4)、以降、ENT_ から始まらないページもすべて描画対象となる。

本サーバのすべての機能を紹介&確認することは大変なので、本サーバの目玉機能の一つであるファイル提出の部分についてだけ紹介する (図5)。

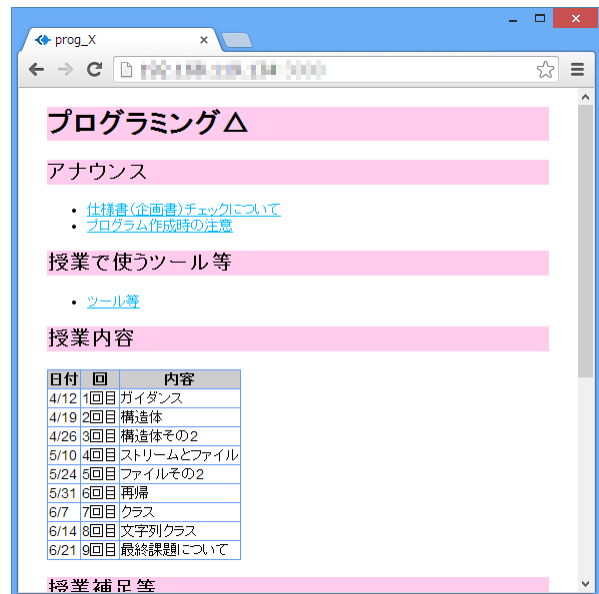


図4 正しくセッションが設定された場合

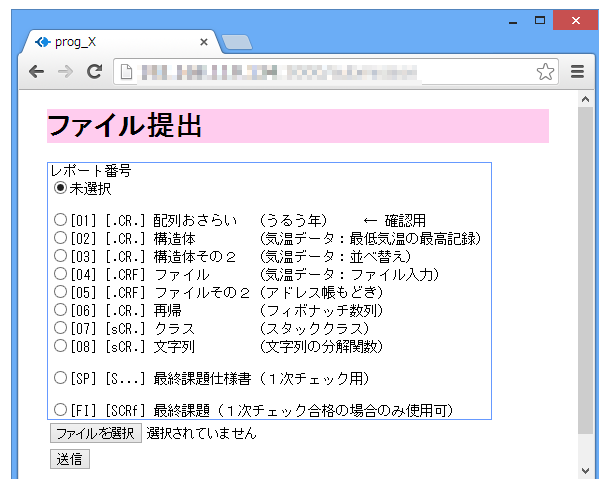


図5 ファイル提出

プログラミング授業においては基本的に毎週ファイル提出課題がある。一つ一つは比較的小さなプログラムになるが、数が多いので管理が大変になる。そこで、提出の際に学生にレポート番号を入力してもらい、UserIDとレポート番号を付加したファイルを格納している。提出されたファイルの中身は自動的に一定範囲解析され、不足ファイルの有無など、その結果が即座に学生の画面に現れるようになっている。そのため、学生自身が自主的に間違いを発見することができ、提出ミスが減らす効果としても役立っている。この処理は元々PHPでできていたため、すべてJavaScriptに移植することになったが、比較的スムーズに移行ができた。今回は移植をテーマとしたためわざと元のシステムに沿った形で作ったが（まず、それが可能か知りたかったため）、この部分はjQueryを使えばもっとわかりやすいインターフェイスが可能になると考えており、現在、修正中である。

5. まとめ

以上、本研究では、Nodeを用いたプログラミング授業支援システムの開発について述べた。まず、プログラミング授業支援システムを構築する上で重要なことを述べた後、Nodeを使った授業支援サーバの構築について述べた。今回は前システムの移植をメインとしてサーバ構築したが、必要な機能は比較的スムーズに移植できたと思われる。

ユーザビリティに関してまだ改造中のため、今回はほとんど触れられなかったが、感触は極めて良好であり、すべてがJavaScriptで作成できるメリットはかなり大きいと感じている。Webベースのシステム構築の際に参考になれば幸いである。

参考文献

- [1] PukiWiki: <http://pukiwiki.sourceforge.jp>
- [2] 松永豊、プログラミング演習授業支援システムの開発、愛教大学研究報告59輯（教育科学編）2009
- [3] 松永豊、プログラミング演習授業のための面接予約システムの開発、愛教大研究報告58輯（教育科学編）、2008
- [4] PHP: <http://www.php.net>
- [5] jQuery: <http://jquery.com>
- [6] Node.js: <http://nodejs.org>
- [7] Nodeクックブック、David Mark Clements、O'REILLY、2013
- [8] Jade: <http://jade-lang.com>

(2013年9月30日受理)