

ホームページ作成授業における HTML 文法チェッカーの役割

関東学園大学 経済学部

吉澤 康介, 三宅 修平, 安斉 公士

〒373-8515 群馬県 太田市 藤阿久町 200

Tel 0276-32-7800 Fax 0276-31-2708

e-mail: {z-yoshi,z-miyake,z-anzaik}@ue.kanto-gakuen.ac.jp

概要

本学の情報教育の一環として実施しているホームページ作成授業では、問題発見・解決能力の育成とコンピュータの仕組みの理解を重視する立場から、エディタを用いて HTML のタグを直接入力させるスタイルを採用している。ここで問題となるのが、学生がどのようなタグを覚え、それを正確に使用しているかどうかを評価する仕組みである。本学の場合、受講しているのはいわゆる文科系の学生であり、授業の目的もプログラムの養成にあるのではなく、自学自習で何かを創作するという経験を積ませることにある。したがって、タグのチェックに、プログラマ向けの文法チェッカーを利用することは「過大仕様」である。そこで、学生が使用したタグをカウントするだけの簡単な HTML 文法チェッカーを作成し、これを授業の中で使用した。この程度の仕掛けでも、「自分なりに参考書調べて試行錯誤すること」に消極的な学生などに対して、ほどよいストレスを与え、学習を進める上で一定の効果を上げていると思われる。

1. はじめに

現在、多くの大学・短大等において、情報教育の一環としてホームページ作成授業が行われており、本学でもいわゆる情報リテラシ教育の後半の大部分をこれにあてている。

ホームページ作成の「教授法」にはさまざまな流儀があると思われるが、本学では、テキスト・エディタを用いて HTML のタグを直接入力させるスタイルを採用している。(以下、この教授方式を「タグ直接入力方式」と呼ぶ。)

一方、ホームページ・エディタを使えば、いちいちタグを覚えなくても、GUI ベースで、より簡単に、よりきれいなホームページを作ることができる。あるいは、ワープロソフト(Word など)で作った文書を「HTML 形式で保存」しても、ホームページを作成することができる。(以下、これらの教授方式を一括して「HP エディタ方式」と呼ぶ。)

表面的には HP エディタ方式がより簡便・適切な方式のように思えるが、筆者らの教育コンセプトを実現するためには、タグ直接入力方式と適切な学習成

果の評価システムを組み合わせたスタイルの方が望ましい。

2. ホームページ作成授業のデザイン

2.1. 情報リテラシ教育の基本コンセプト

筆者らは「いかに社会が変化しようと、自分で課題を見つけ、自ら学び、自ら考え、主体的に判断し、行動し、よりよく問題を解決する資質や能力」の涵養を、情報リテラシ教育の目的に据えている。[1]

いわゆる「問題発見・解決能力」、より端的に言うところ「理解力」、「思考力」の向上を目的とした教育である。

情報リテラシ教育の目的を「代表的な OS やアプリケーションの操作法教育」とする考え方も根強いが、IT 革命という言葉に象徴される昨今の状況の中では、特定ソフトの操作法に関する知識はたちまち陳腐化してしまう。[3]

そもそも、さまざまな情報機器を活用して自分の仕事や生活を効率化し豊かにしていくためには、単なる操作法「以外」の部分、すなわち、旺盛な好奇心、積極的な学習態度、他者とのコミュニケーション能力、

¹ A Role of a HTML Syntax Checker in a Home Page Making Class.
K. Yoshizawa, S. Miyake, K. Anzai
Kanto-Gakuen University.

試行錯誤を恐れない粘り強さといった能力の影響の方がはるかに大きいと考える。

こういった基本的な能力に富む学生は、適切な学習の場を与えてやりさえすれば、必要な操作法などたちまち習得してしまう。

また、こういった能力を伸ばす訓練を必ずしも十分に受けてこなかった学生に対しても、情報リテラシ教育が格好の訓練の場となりうる。

2.2. 理解と試行錯誤

いろいろと調べて、実際にやってみて、それでやっと少しだけ納得できる——「理解」とは、かくのごとく時間と労力を要する過程である。

ところが、一般の「教室」の中でこのような試行錯誤を伴う授業を実施するのはなかなか難しい。

「試行錯誤」が成立するためには、学生の成果物を評価して学生自身にフィードバックすることが必要不可欠であるが、例えばペーパーテストやレポートを利用している限り、このフィードバックが質・量ともに限られたものにならざるを得ない場合が多い。(評価が行われるのが期末テストの一回だけ、評価結果のフィードバックも「優」、「良」、「可」、「不可」といった内容だけ、という場合も多いと思われる。)

ところがコンピュータをうまく活用すると、この「評価」の部分を自動化できる。自動化できれば、学生は好きなだけ試行錯誤を繰り返すことができる。

したがって、自動評価システムの存在を前提として、成果物(学生に何をやらせるか)と評価方式(学生に何をフィードバックするか)を適切に選択すると、学生の「理解」の内容を大幅に向上させることができる。

この時の教育効果が、大きく分けて次の二つの異なったレベルで現れることに注意されたい。

(1)メタ・レベルの教育効果——「2.1. 情報リテラシ教育の基本コンセプト」で述べたような一般的な「問題発見・解決能力」の向上。(「思考力」、「理解力」あるいは「生きる力」の向上と言い換えてもよいであろう)

(2)ドメイン・レベルの教育効果——その授業の対象領域に関する理解。

要するに、評価の自動化と試行錯誤を前提とする授業スタイルは、一石二鳥の効果をもたらすのであ

る。

2.3. ホームページ作成授業への適用と概要

では、この授業スタイルをホームページ作成授業に適用することを考えてみよう。

筆者らの授業では、ワープロで簡単な文書が作成できるレベルの学生を想定し、最終的にかなり本格的なホームページを作らせることを目標としている。

この授業は大きく分けて、次の二つの段階から成り立っている。

(1)【前半】ホームページの基本的な仕組みを理解し、簡単なホームページを作成できるようにする。

(2)【後半】各自で一定のテーマを決めてホームページを作成し、学生を小グループに分けて相互批評させる。

本論文ではこの前半部分に「評価の自動化」を導入する場合を議論する。

(後半部分の内容と狙いについては参考文献[2]を参照されたい。)

2.4. メタ・レベルの教育効果をあげるために

メタ・レベルの教育効果をあげるためには、学生が「自分で課題を見つけ、自ら学び、自ら考え、主体的に判断し、行動し、よりよく問題を解決する」ような行動をとらざるを得ない状況を作り出すことが肝要である。

そこで、ホームページ作りの基礎を説明した上で、「あとはホームページ作りに関する参考書を自分で探し、自分で調べていろいろやってみなさい」という授業スタイルをとることにした。(作り方を逐一説明したり、「書いてあるとおりにやればできる」といった類いの資料は与えない。)

好都合なことに、HTML には「タグを一つ覚える」と何か新しいことが一つできる」という特性がある。したがって、タグ直接入力方式を採用すれば、学生が使ったタグの個数をカウントするだけでも、かなりの程度まで学生の努力の分量を評価できる、という利点がある。

ところが HP エディタ方式でホームページを作らせると、容易に「ハイレベル」なページができてしまう。(タグをカウントしても、それは HP エディタの能力(?)を測定していることにしかならない。)

この場合はホームページの「内容」の方を評価すればよいのだが、これを自動化するのは一筋縄ではいかない。

2.5. ドメイン・レベルの教育効果をあげるために

一方ドメイン・レベルでは、学生に対して次のようなことを理解してもらいたい。

- (1) コンピュータ全般にわたる基本常識——コンピュータはプログラムで動いている、ということ。
- (2) OS レベルの諸概念——ファイル、フォルダ、アイコンなどの基本概念、拡張子と対応するアプリケーションの関係、汎用性のあるデータ形式とアプリケーション固有のデータ形式、データとそれを操作するアプリケーションの区別など。
- (3) ホームページの一般的知識——ホームページは、HTML という、ある種のプログラミング言語によって記述されている。HTML は、文書の構造や表示法を指示するためのタグの集まりとして実装されているということ。
- (4) HTML の初歩的文法——基本的なタグに関する知識、URL やハイパーリンクの意義。

ホームページ作成授業においても、(1)や(2)のようなコンピュータ全般に係る知識は必須であるし、特に(1)を単なる字面の上だけでなく実感として理解して欲しいと考えている。この感覚の有無が、コンピュータを単なるブラックボックスとして見て、あてがわれた機能をあてがわれたままに使うのか、それともいろいろ調べて積極的に使いこなそうとする意欲を持てるのかに密接に関係していると考えているからである。(プログラミング言語を習得せよという意味ではなく、もっと広く捉えて、「コンピュータに対していろいろ指令できるんだ」ということを分かってくれたい。)

同時に、実際にホームページを作成するためには、(3)や(4)の知識が必要である。

ところが HP エディタ方式を採用する限り、これらの諸概念が HP エディタの背後に隠されて「見えにくく」になってしまう傾向が強い。また、特定の HP エディタに固有の知識と、コンピュータ全般に通用する汎用的な知識の区別が極めて曖昧になってしまう。

一方、タグ直接入力方式をとると、現在のコンピュータを理解するために必要不可欠なこれらの諸概念が、むき出しのまま学生の前提示されることに

なる。

敷居は高いが、いったん理解しはじめると余計なものも介在してないだけに、むしろ理解は進みやすいのではないかと考える。

2.6. タグ直接入力方式と HTML 文法チェッカー

以上述べてきたように、筆者らの教育コンセプトに沿った教育効果を上げるためには、「タグ直接入力方式」を採用することが望ましいと考える。

そして、学習成果の評価の自動化のために HTML 文法チェッカーを併用することにした。

実際の授業では分かりやすい目標を明示した方が学生の学習意欲を高められるので、

- (1) タグを 10~20 種類以上使うこと(最低要求個数は授業によって異なる)
- (2) 2 ページ以上作り、リンクで結ぶことを課題として学生に課した。そしてこの二条件を満たしているか否かを判定する HTML 文法チェッカーを開発した。

3. HTML 文法チェッカーの概要

開発に無制限に労力をかけられないので、HTML 文法チェッカーはタグの個数をカウントするだけ単純なものを用意した。

当初はフリーソフトウェアとして入手可能な文法チェッカーの使用を検討したが、その診断メッセージの内容が SGML の知識を前提としたものであったり、初心者にとってはあまりにも厳格すぎる判定を下したりする傾向があり、採用には至らなかった。

そこでシステムは自主開発とし、Perl で記述した CGI として実装した。その内部構造は非常に単純なものであり、

- (1) 学生にチェック用ホームページ上で自分のユーザ ID を入力させる。
- (2) ユーザ ID からホームディレクトリのありかを決定し、入り口の HTML ファイル(index.html)を読み出す。
- (3) 内容をごく簡単に解析し、<html>~<body>~本体~</body></html>といった基本構造を満たしているか検査する。
- (4) 本体部分で使われているタグを抽出してカウントする。
- (5) ハイパーリンクがあればそれを辿って処理を

繰り返す。

(6) 最後に診断メッセージを表示し、判定結果を記録する。

というものである。実行例を図1に示す。

セキュリティ対策であるが、このシステムは実習環境の閉じたイントラネット上にあるので、チェッカー自体には特別なアクセス制限はかけていない。各自のホームページ用コンテンツはユーザ毎にアクセス制限がかけられた WindowsNT 上の個人用フォルダにあるため、いちおう安全である。ユーザ ID を詐称することは可能だが、一度「合格」状態になるとそれを記録しているので、他人のユーザ ID を詐称しても「単に本人の代理でチェックしてあげている」ことにはかならない。(したがって誰かを困らせることはできない。)

おわりに

学生は、最初は戸惑いを見せるものの、間もなく自分で用意した本を調べながら黙々とタグを入力しては結果を試すという作業に集中するようになる。使っているタグの種類が次第に増えていくのがかなり励みになるらしく、けっして能力が高いとは言えない学生でも最終的に課題の目標をクリアできる場合が多い。したがって、「タグをカウントする」という単純な仕掛けでも、一定の教育効果を上げていると考えられる。

ここで、「単純な仕掛けでも使い方次第で大きな効果がある」ということを特に強調しておきたい。予算や時間をかけて本格的なものを用意することに固執する必要はないのである。

しかし、学生の作成した HTML ファイルの中身は雑なものが多く、より木目細かい診断を下してそれを学生にフィードバックできるシステムがより望ましいと考える。そのようなある種の教育用エキスパート・システムの研究・開発が、筆者らの次の課題である。

参考文献

- 1) 三宅修平, 安斉公士, 吉澤康介, 情報リテラシ教育の理念とその授業内容について, 平成 11 年度文部省情報処理教育研究集会講演論文集, pp.32-35, 1999
- 2) 吉澤康介, 三宅修平, 安斉公士, 情報リテラシ教育用システムのあり方に関する考察と実践, 関東

学園大学経済学紀要第 27 集第 2 号, 2000, (掲載予定)

- 3) 原田康也, 辰己丈夫, 楠元範明, 「情報教育」の情報化, 情報処理学会研究報告(2000-CE-55), vol.2000, No.20, pp.41-48, 2000

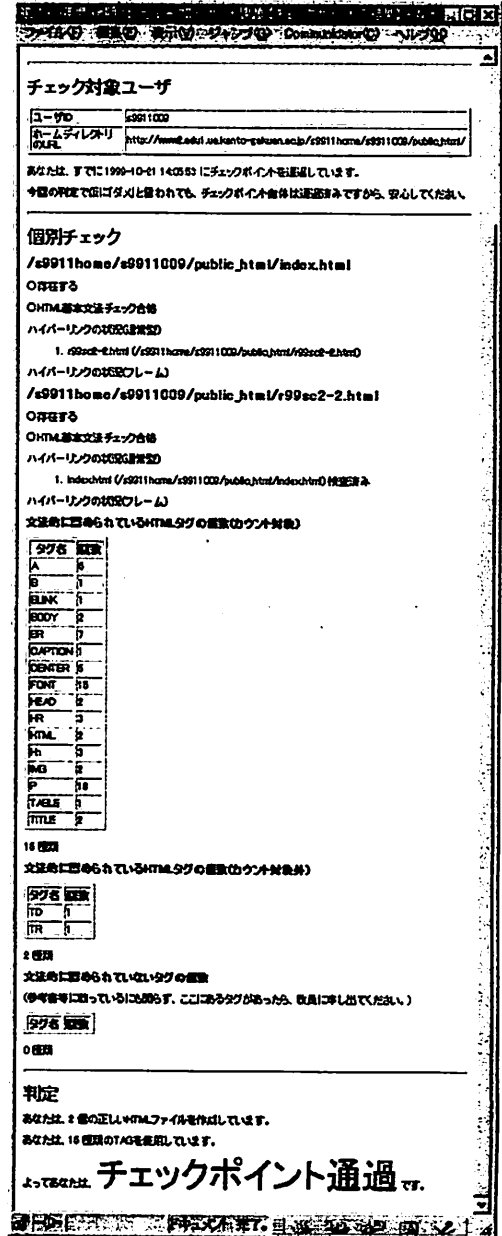


図 1 HTML 文法チェッカーの実行例