

情報科教育法 II 2003 # 1

久野 靖*

2003.10.22

1 はじめに

こんにちは、久野です。自分の本務先は筑波大学 (東京地区、昔の「東京教育大学」跡地) ですが、非常勤講師として後期の「情報科教育法 II」を担当させていただきます。学芸大の人ではないので、授業に関する連絡や掲示、レポートの授受などはネット経由で行うこととなりますが、その辺ご了承ください。では半年の間ですが、よろしくお願いします。

1.1 科目の運営方針

本科目をはじめ、教科教育法の科目はその教科を担当するに当たって知っておかなければならない固有のことから身につけて頂くことを目標としています。そして、教科「情報」の場合は次のような特殊性があります。

- 他の教科であれば皆様は中学・高校時代に生徒としてそれらの教科の科目を学んだ経験があり、「どのようなものか」が大体判っている。しかし「情報」の場合は 2003 年度から始まった教科であり、そのような経験は一切期待できない。
- しかも「情報」では実習が非常に重視されるため、その教員にとっては実習をうまく運営する能力が欠かせない。ここで「実習」というのは必ずしもコンピュータに向かって行なうものには限定されず、たとえばグループでの討論、共同製作、プレゼンテーションなども含まれることに注意。
- さらに「情報」では座学的な部分においても、技術的な事柄から「情報倫理」「情報社会」など社会的な事柄まで、カバーしなければならない範囲が非常に広い。他方、専門教科「情報」では非常に専門的な内容まで含まれている。「情報」の免許を持っているれば、それらも建前上は教えることができることになっている。
- そして「情報」の教員として着任した場合、教科を教えているだけでなく、「学校の情報化」などの問題にも関与することが否応なく求められるだろうと予想される (すべて「情報」の教員に押しつけられたらたまったものではないが、そういう危惧があることは熟知しておくに越したことはない)。

このように色々と大変なことがあります。それを系統的に講義する、というのも大変です。講義を聞くだけでは身につかないことが多く含まれていると考えます。そこで、この科目では「各自に宿題として授業計画案を作成していただき、また時間中に 20 分程度の模擬授業を実施しながらそれに基づいてさまざまな事柄を学ぶ」という方針を採用したいと思います。模擬授業は皆様に順番に行なってもらいます。

ただし、皆様の中には単位数の必要からこの科目を選択していて、「情報」の免許は取得しない人もいと承っています。それらの人は「生徒」の役割だけやってくればよいことにします。多少不公平ですが、「先生」になって生徒を教えるのは重大な責任を伴うわけですから、「先生」になる人に応分の修行をして頂くことはどうしても必要だと考えます。よろしくご理解ください。

*筑波大学大学院経営システム科学専攻

1.2 教科書/参考書/資料

本講義の教科書としては次のもの2つを指定しています。

- 第一学習社, 検定教科書「情報C」, 2003, 890円.
- 大岩, 橘, 半田, 久野, 辰己: 情報科教育法, オーム社, 2001. ISBN4-274-07922-8, 2000円.

1冊目は「情報C」の教科書として、模擬授業等もすべてここから選んでやって頂きますので、必ず毎回持って来てください。高校の授業と同様(かどうか知りませんが)、持って来ているかどうかチェックし、教科書を持って来ない場合に出席点を減点することがあります(あんまりやりたくないですが)。ときになぜ情報Cなのかという点ですが、皆様には「使い方」ばい情報Aや「技術的内容」ばい情報Bに比べて「社会的内容」の多い情報Cが一番なじみがなく、従って十分に知っておく必要があると考えたからです。

2冊目は「情報科教育法」のテキストです。毎回みっちりテキストに沿ってやるつもりはないのですが、教科「情報」の概要や授業指導案作成など参考になる事項が多く含まれていますので、必要に応じて(特に課題をやる時などに)参照することも多いと思います。これは持ってこなくてもいいですが、購入はしておいてください。

あと、前期の教科書に指定されていたと思いますが、高等学校「情報」指導要領解説(100円の冊子)も必要です。万一買っていない人がいれば購入しておいてください。

自分はテキストを順番に教える授業というのは嫌いなので、本講義では毎回プリントおよび必要に応じて関連資料コピーなどを配布し、それに基づいて授業を進めて行くつもりです。

これら以外に、参考になるとと思われる書籍(参考書)についてはそのつどプリントなどで紹介するつもりです。情報教育は全体として非常に広い範囲に渡っており、テキストや配布資料でカバーされることはそのごく一部にすぎません。ですから、参考になるとと思われるものにはできるだけ多く目を通して見て、よく参照しそうなものは購入するなどしてください。

1.3 クラスのWebサイトと掲示板

この科目に関するお知らせや情報を掲載するためのページを用意しました。随時情報を追加していきます。

<http://www.u-gakugei.ac.jp/~kuno/ggu03/>

ここからたどれる「お知らせ」については必ず内容を定期的にチェックしてください(週に2~3回くらいは)。「お知らせ」を見なかったことによる不利益は救済しないことがありますので注意。また、「掲示板」については質問や意見交換などに自由に使ってください。Geocitiesの掲示板ですのでニックネームとかも構いません。が、他人の名前を語ったりとか荒らしは遠慮したいのでよろしく。

1.4 メーリングリスト/WWW上の情報

今日ではネット上でも「情報教育」に関して非常に多くの情報が取り寄せられます。暇があればいろいろ見てください。メーリングリストも意欲があれば参加してみてもよいと思います。参加しなくても過去のメッセージがWeb経由で読めるものがあります。

- 文部科学省「情報化」 http://www.mext.go.jp/a_menu/shotou/zyouhou/ — 「教育の情報化」に関する情報がまとめている。
- 学芸大附属高校 <http://www.gakugei-hs.setagaya.tokyo.jp/> — 附属高校の取り組みについて分かる。
- InformationStudy <http://www2.ics.teikyo-u.ac.jp/academic/> — 情報教育メーリングリスト。
- kjm <http://www.tt.tuat.ac.jp/kjm/> — 情報科教員養成メーリングリスト。

- 2ちゃんねる「教育・先生板」 <http://school.2ch.net/edu/> — 日本最大の匿名掲示板。情報はまあ玉石混淆だけど。

このほか、ぐるぐる (<http://www.google.co.jp/>) で、『教科「情報」』などで検索するとさまざまなサイトが見つかります。

1.5 次回以降の模擬授業割り当て

上で述べたように、次回から模擬授業をやって頂きます。内容は「情報C」教科書の連続する1~3項目(1項目2ページ)くらいを各自選んでください。みな同じところをやってもつまらないので、やりたい個所の希望を取るようになりますが、とりあえず来週とさ来週やる人は自由に選んでいただいて結構です。誰かさっさとやりませんか？

時間は20分ですから、通常の高校授業(50分)の前半部分、実習を含む場合は実習に少し入ったくらいで打ち切るという感じになるでしょう。実習は選んだ单元によりますができるだけ盛り込むようにしてください。割り当たった人が用意するものは次の通り：

- 指導案。50分ぶん作成し、先頭または適当なところから20分ぶん実地してもらいます。もともと、時間配分がまだ分からないでしょうから、時間については柔軟に考えていただいてもいいです。
- 配付資料。あってもなくても良いですが、一般的に言えばなるべく資料も配付した方が授業はやりやすくなると思います。

上記については、すみませんが自腹でこのクラスの人数ぶんプラス1(久野のぶん)、コピーしてきてください。そのほか、授業にプレゼンソフト等を使う場合はその準備もお願いします。

1.6 出席レポート、成績

毎回、A5サイズの紙に「出席レポート」を書いて提出してもらいます。書く内容はそのつど指示しますが、模擬授業の感想、その回の内容で取り上げたことに関する自分の意見などになると思います。出席点を差上げますから、紙の冒頭に日付、学籍番号、氏名を記入してください。

さらに原則として毎回、次回までの課題を出します。基本的には「~の内容に関する授業指導案を作り、どのような工夫をしたか解説せよ」という感じにしたいと思います。×切は次回授業の前日(23:59:59)まで、課題はメールで提出して頂くようにしたいと考えています。

また、冬休みには紙で提出して頂く大レポートを課します。期末にやっても他の科目の試験やレポートとぶつかって大変なだけですから、お正月にやって頂き、冬休み明け提出とします(ただし2004年1月7日は久野が学会につき休講とするので、提出期限は翌週1月14日です)。

成績は各回の出席、(該当する人は)模擬授業の担当、各回の課題、大レポートを総合してつけます。期末試験は実施しません。

2 学習指導案の作成

2.1 学習指導案の考え方

先生は授業をやるときに、思いつきで教科書を先頭からやっているわけではない。そんなことをしていたら内容が抜けだらけになってボロボロになるのは目に見えている。ではどうするか？ → 指導案を作って何と何を教えるか、その際に何を重視し何は時間が無ければ捨てるかなどを考えておく。そうすれば、予定外のことが起きても重要事項だけは押えて終わることができる。…生徒には「見えない」先生の重要な仕事。

指導案があるおかげで、上記も含めて次のようなメリットがある。

- 各授業において、何が重要かを確認した上で取り掛かれる。
- 授業が終わった後で、その授業の出来不出来を自己評価する基準となる。
- 反省点を一緒に記しておくことで、次からはよりよい授業ができる。

では、うまい先生の指導案を貰えば完璧なのだろうか？ 否！ 実は最も重要な指導案の機能は次のこと。

- 指導案を作ることで、自分の行う授業に対するイメージができ、予め頭の中で「練習」しておくことができる。

たとえばプレゼンテーションの発表であればきちんとできるまで発表練習を繰り返すことになるが、授業の場合は生徒がいなければ練習できないし、生徒を練習に使うこともできない(たまにいるけど…)。また、プロになったらとてもそんな時間はない。だから、指導計画案を作りながら教室の様子をイメージして、こういう授業をやろう、と自分なりに考えを固めることが最も重要な目的。そういう意味では、指導案は自分が考えたことの「メモ書き」であり、それを見て自分のイメージを思い出されればよい(他人が見てもあまり役立たないのはこのため)。

このように重要なものであるため、この授業でも皆さんに指導案を作ってみて頂く。なお、指導案と言った場合一般には次の3レベルがある(教科書 p72 参照)。

- (1) 年間計画
- (2) 単元の学習計画
- (3) 各時限の授業案

(1)と(2)はいろいろ参考になるものも出ているし、ここで一般論を言ってもあまり役立たないのでパスし、(3)を重点的にやる。

各時限の指導案の場合、次のような順番で考えるのがよいと思う。

- その時限で何を学ばせたいかの目標を決める。
- その目標が達成されたかどうかは、何を評価すれば分かるか考える。
- その他注意すべきことを思いつく限り挙げる。
- 最後に時間配分して進行案を表にまとめる。

実は一番重要なのは「欲張って内容を詰め込み過ぎない」ことだが、これは(2)の受け持ちでもあるのでそう注意するにとどめる。

時間配分については、まず「導入」でなぜこのような内容を学ぶのかはっきりさせ、次にそれに基づいて具体的な内容を述べる。実習が含まれるのであれば、その後実習の説明をして実習させる。最後にその回学んだことのまとめを述べて終わる。

実習が含まれる場合、50分ではかなり苦しい。が、「情報」に2時間連続のコマが貰えるかどうかは学校の事情まかせで心許ない。実習で時間の掛かる内容であれば、しょうがない場合はある時限に実習の準備をすべて済ませ、今回は最初から「前回説明の通り実習開始」とする(が、説明を覚えていてくれないだろうからかなり悩む)。

2.2 授業案をHTMLで書く

このクラスでは、皆様に授業案を書いて頂くのに「HTMLで作成し」「電子メールで久野あて送付」してもらおうようにする(全員の授業案を集めてサーバに載せるようにしてみたい)。その「ひな型」を用意したので付録に載せておく。ファイルもあとで配るのでこのフォーマットで統一して欲しい(後で説明するがスタイルは変えてよい)。

3 HTML と CSS

3.1 HTML/CSS を取り上げるわけ

これから上記授業案はじめ、HTML を書く機会が多々あると思うので (あと、生徒に HTML を指導することも当然よくあると思うので)、今回のメインの内容としては「HTML と CSS」を取り上げることにした。そもそも、HTML は「ある程度いい加減に書いてしまってもブラウザがそれなりに表示してくれる」ため、いい加減な (自己流の) 書き方のまま固まっている人があまりにも多い。教科「情報」の教員になるのなら、そのような状態のままでは困るので、ここで改めて整理して取り上げる。なお、皆様は一応 HTML の教育は受けているものと承っているが、既に知っている人でも一応整理のつもりで聞いて頂きたい。

3.2 マークアップと WYSIWYG

本題に入る前に、マークアップと WYSIWYG の違いについて整理しておこう。

- マークアップ (Markup — 印つけ方式) とは、テキストの中に (ちょうど HTML のタグのように) 特別な「印」が埋めてあって、それに従ってプログラムが内容を整形して表示したり印刷したりする方式。編集するときは単にこれらの印と内容を一緒に編集する。TeX がこれ。
- WYSIWYG (What You See Is What You Get — 見たまま方式) とは、Word などのように画面に「印刷したのと同様の状態が見えて」いて、ソフトを操作してその状態を見ながら編集する。

さて、あなたはどちらが「好き」ですか？

好き・嫌いとは別にどちらでもよいけど、マークアップにも WYSIWYG にもそれぞれ利点と欠点がある (何だと思う？ この「利点と欠点を列挙させてみる」という技は授業でよくお世話になるので覚えておくとよい)。まず WYSIWYG の利点から挙げてみよう。

- できあがりの様子が絶えず確認できるので安心だし分かりやすい。

そうだそうだとか思います？ しかし WYSIWYG には次のような欠点がある。

- × 画面の解像度でしか確認できない (印刷するとずっと細かいズレが気になったりすることも…しかしそれを調整するのは困難)。
- × 画面に見えないものはそのままでは編集できない。たとえば WWW のリンク先とか…
- × まとめて一括して指定を変更するようなことがやりにくい (たとえばすべての下線つきのところをボールドに変更するとか)。
- × 編集するのにマウスやメニューを駆使するのはすごく時間がかかって疲れる (私は TeX でないと本なんか書く気にならない…)
- × 最終出力 (できあがり) を 1 つ決めなければならない。別のできあがりになろうと思うと作業し直しになる。

こうしてみると、ワープロみたいに手元のプリンタで出せばそれでいいようなものならともかく、少なくとも WWW には WYSIWYG は極めて向いていないと言える。

では、マークアップはどうか？ 上で述べたことの裏返しが多い。

- 命令で指定するのでどんな細かさでもそういう命令を用意すれば指定できる。
- 最終出力に見える見えないと関係なくすべての情報が命令として見えているので確認や修正が容易。
- 指定のまとめた変更とかもテキストエディタの一括置換などでできる。
- エディタは WYSIWYG ソフトより軽くて使いやすしいし、マウス操作よりもキーで指令を打ち込むだけの方が慣れると楽。

○ 指令の設計によるが、1つのソースをさまざまな行き先の用途にそのまま変更なしで使えるようにできる。もちろん、弱点もある。

× 指令を覚えていないとまったく使えない。できあがりの様子が整形ソフトを動かしてみないと分からない。これらの点は当然、初心者にとって敷居が高い原因となる。だから世の中は Word だらけなのだけど、それがいいかどうか？ 沢山使う人であれば指令は覚えてしまうし、できあがりも想像できるようになるので、マークアップの方が結局いいとこだらけになるのでは？ まあこの辺はそれぞれの考えということで。

3.3 HTML に関する質問

HTML に関する次の記述は正しいと思うか、間違っていると思うか？

- (a) HTML で「<なんとか>」と「</なんとか>」が対になっているタグの場合、曖昧さがなければ閉じるタグを省略してもよい。
- (b) HTML で「<なんとか 属性=値 属性=値>」のような属性を指定するとき、「値」は必ず「"..."」で囲まなければならない。
- (c) どのバージョンの HTML かは HTML ファイルの冒頭に記述しなければならない。
- (d) HTML では<head>...</head>とか<title>...</title>とかは省略してもよい。
- (e) HTML の文法は厳密に定まっているが、手元のブラウザで表示させてみてちゃんと見えるならそれで問題ない。

3.4 HTML の基本的な解説

HTML は SGML(Standard Generalized Markup Language) と呼ばれる規格 (ISO/JIS 標準になっている) に基づいたマークアップ言語。SGML 的には文書の個々の構成部分を「要素」と呼ぶ。要素の範囲を区切る印 (マークアップ) として使うのが「タグ」。何でもタグと呼ぶのはヘン。

HTML の要素には次の 2 種類がある。

- <名前 属性...>内容...</名前> — 対のタグで囲む要素
- <名前 属性...> — 単独のタグだけから成る要素

対のタグで閉じタグが省略できるものもあるが、それをいちいち覚えるより全部閉じタグを明示した方が話が簡単なのでこれを勧める (生徒にやらせるときはタイプ量が多くてしんどいという問題があるが)。HTML の次の標準規格である「XML」「XHTML」では閉じタグは省略不可だから、今からそれに備えて習慣をつけておく、という意味もある。

また、属性を指定するときは「特殊な文字を含まないもの」については引用符で囲まないでもいいが、その「特殊な文字」を覚えるより毎回囲んでしまう方が簡単なのでそれを勧める。引用符は「"..."」でも「'...'」でもよい。

HTML にはいくつかのバージョンがある。

- HTML 2.0 — 現在使われているうちではもっとも古い (機能の少ない) 規格。
- HTML 3.2 — 文字に色をつけたり揃えを指定したりといった機能が追加されている。またフレーム機能もここから入った。
- HTML 4.01 — HTML の最終版。現在いちばん広く使われている。実際にはさらに 3 種類 (厳密版、互換版、フレーム) に分かれる。
- XHTML 1.0/1.1/... — XML によって HTML を記述し直したもの。これからこれが増えてくると思う。

自分が書いている HTML がどのバージョンなのかはちゃんと意識した方がよい。そもそも、HTML の「規則」は DTD(Document Type Definition、SGML が定めている定義方法) によって厳密に規定されていて、それに合致しているものが正しい HTML。HTML 文書の冒頭にはどのバージョンかを表す DOCTYPE 宣言を次のようにつけるべき (だが、つけてない人やいい加減なものをつえけるツールが多い)。

- `<!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML 2.0//EN">` — 2.0
- `<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 3.2//EN">` — 3.2
- `<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN">` — 4.01 厳密版
- `<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">` — 4.01 厳密版
- `<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Frameset//EN">` — 4.01 フレーム版

これも、本当はつけるべきだが、生徒につけさせるのがしんどいとは思う。

3.5 HTML の「正しさ」について

HTML は上述のように DTD によって「正しい」書き方が厳密に決まっている (その上で「正しい」省略方法も厳密に決まっている)。しかし世の中では適当なブラウザで表示させて大丈夫ならあとは気にしないという人も多い。なぜそれでは困るのだろうか? 実は世の中にはこれ以外にも「標準規格からは外れているけど、使えるからいいと思われている」ことが沢山ある。

- 2 バイト文字の丸数字やローマ数字 (機種依存文字)
- quoted-printable でエンコードされたメールや HTML 添付メール
- MS Word の添付ファイル

なぜ規格を守るべきなのだろう? 独自拡張機能で「より便利に」なるなら、そのような機能を使ったらいい、という説もあるがどうだろう?

答えは、さまざまなソフトやツールは規格に従ったものを処理することが保証されているのであり、自分がたまたま規格外の機能を (独自拡張により) 扱えるソフトを持っているからといって、他の人も同じ機能が使えるとは限らない、さらには自分のソフトもバージョンが変わったらその機能が変わってしまうかも知れない、という点にある。だから、相手と合意の上であれば独自拡張を使ってもいいのだが、WWW の場合は「世の中の人」に向かって情報を送るわけだから、規格は守っておくべきである。

では、HTML を書いたときにその正しさはどうやって検証すればいいだろう? そのために、HTML の正しさをチェックするためのツールがある。その代表的なものである `htmllint` を使って自分が公開しているページをチェックしてくれるサイトがある:

<http://htmllint.itc.keio.ac.jp/htmllint/htmllint.html>

また手元のファイルを簡単にチェックできるように、こちらのマシン `cedar` にも久野のディレクトリに入れておいたので活用して頂きたい。

`~kuno/htmllint` ファイル名

で使える (簡単でしょう?)

3.6 演習 1: HTML を打ち込んでチェックしてみる

1. マシンを立ち上げ、エディタを起動して、次の内容をこの通りに打ち込む。

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN">
<html lang="ja">
<head>
<title>test...</title>
</head>
<body>
<h1>テスト</h1>
<p>これはテストです。</p>
</body>
</html>
```

2. ファイル名を「test.html」として保存する。文字コードはUNIXでも使うためにEUCに設定するのがよい。
3. Astec-Xを起動し、cederにログインする。コマンド窓が開いたら

```
ls ←ファイル test.html がちゃんとあるか確認する
~kuno/htmlint test.html ←チェックする
```

もしも沢山エラーがあつて見られないようなら2行目を

```
~kuno/htmlint test.html |& less
```

とするとよい(スペースで1画面ずつ見ることができ、「q」で終了)。エラーがあると点数がマイナスになるので、マイナスだったらちゃんとエラーを直すこと。(100点はめざさなくてよい。大変だから。)

4. OKになったら、UNIX上のNetscapeブラウザとWindows上のIEブラウザでファイルを表示してみてOKかどうか確認する。
5. 今度はさっきのファイルをわざと「壊して」みる。たとえば閉じタグをわざと消すとか、16ビット(全角)文字でタグを書いてみるなど。その状態でhtmlintやブラウザでチェックしてみよう。

htmlintを使うことで、ブラウザでいきなり表示させたのでは分からない間違いをチェックできる(あなたのブラウザでOKに見えてもエラーがあると他人のブラウザではボロボロになることがある!!!)。100点はめざさなくてよいので、自分で作ったHTMLは毎回これでチェックすること。(慣れるまでは沢山書いてからチェックすると山のようにエラーが出るので少しずつ書き足しながらチェックするのがよい。)あと、ブラウザによって見え方が違うことがあるので、複数のブラウザで確認する習慣をつけるのがよい。

3.7 HTML入門の授業案

付録Aに「文書による情報提示」(「情報C」教科書P90)の授業案を掲載した。50分では相当苦しいと思うので、実際にやるなら2コマ使うように工夫した方がいいかも知れない。内容的にはここまでで説明したことから生徒に分かってもらうべきことを精選して提示する(教員は生徒よりずっとたくさん分かっていなければ授業なんかできないので、ここでは沢山材料を提示している)。

なお、この案ではHTMLを手で打ち込ませるとしているし、自分もそうするのが一番だと思っているが、世の中には「HTMLなんか生徒には触れせずにホームページ作成ソフトだけでやる」のがいいと思っている教員は沢山いる。対立を招きやすい微妙な事柄かも知れないので注意。

3.8 CSS 入門の授業案

付録 B に「さまざまな表現方法の工夫」(「情報 C」教科書 P92)の授業案を掲載した。これは内容的にはいろいろなアプローチがあると思うが、私(執筆者)としては CSS を学んで欲しいと思って書いた(実際にこれをする先生は適当に自分が扱えることの中から選んでしまうと思うが)。この部分は模擬授業をやってみようと思うので、授業資料の形にしたものを付録 C につけた。

なお、ここでは class 属性を指定してスタイルをつける方法まで取り上げているが、それだと大変すぎるようなら単に要素の種類に対してスタイルを指定するところまででやめておいてもよいと思う。

3.9 Web ページに CSS を指定する方法

Web ページに CSS でスタイルを指定する方法としては次の 3 通りのものがある(上述の授業案では複雑になりすぎるため(1)だけを説明している)。

- (1) HTML ファイルのヘッダ部分 (<head>...</head>の中)に、次のような形で書く。この方法はファイル 1 つで済むので簡単。

```
<style type="text/css">  
スタイル指定…  
</style>
```

- (2) CSS の指定(上の方法で<style>...</style>の内側に入れる部分だけ)を「なんとか.css」という名前のファイルに入れておき、HTML ファイルからは次の書き方でそのファイル名を指定する(この書き方もやはりヘッダ内に入れる)。

```
<link rel="stylesheet" href="なんとか.css">
```

この方法を使えば、1つのスタイル指定で複数のページの表現を指定できる。

- (3) HTML の本体に出て来る各タグの中で style という属性を指定し、その中にその要素に適用するスタイル指定を書く。たとえば次のような感じ。

```
<h1 style="border: blue solid 4px">はじめに</h1>  
<p style="margin-left: 1cm; color: red">この説明では…</p>
```

それぞれの個所で簡単にスタイルが指定できるが、ごちゃごちゃして見にくくなり、スタイルの統一という点ではあまりよくない。

3.10 HTML+CSS のまとめ

最初に述べたように、HTML は「適当に書いてもそれなりに表示される」ため、適当に済ませている人が多い。なぜそれでは困るか、では最低線として何は押えた方がいいか、といったことは知ってほしい。

また、HTML で「飾り」をつけるため、font など HTML 3.2 の機能を使う人が多いが、本来 HTML は文書の論理構造を示すためのものなので、表現指定は CSS を併用するのがよい(ブラウザによっては — 特に Netscape 4.x — CSS の実装が不完全で問題を起しやすいため困る)。CSS を使った方が HTML 3.2 より自由に多様な表現がつけられる。

実際にどのような表現指定をつけるかは、やみくもにつけるのではなく、ページデザインから考えるべき。この辺はまた回を改めて取り上げる。

なお、HTML や CSS の参考書は色々出ている。以下のもあるのでよかったらどうぞ。

- 久野, 入門 WWW — Unix による情報発信技術, アスキー, 2000. ISBN4-7561-3652-4.

別にこれでもなくてもいいけど、「クソ本」を買ってしまわないように。たとえば「H1 とは、文字を大きくするタグです」とか書いてあるのがあつらしい。こちら「のけぞる本」コーナーも見ておくとよい。

<http://www.ne.jp/asahi/minazuki/bakera/html/book/>

A 授業案: 文書による情報提示

A.1 目標

文章をその論理構造に従ってマークアップすることの意義について分かつた上で、HTML の書き方を学ぶ。

A.2 評価の観点

- マークアップと WYSIWYG の違いについて理解している。
- 論理マークアップの必要性について理解している。
- HTML の一般的な書き方について理解している。

A.3 留意点

- ブラウザの窓幅を変えるなどして、ブラウザの整形機能が働いていることを実感させる。
- 正しくない HTML だとブラウザによって見え方が違つることが示せるとよい。
- 次回にスタイル指定を使って見え方を変更するので、その際に今回打ち込んだものがちゃんと使えるように配慮する。

A.4 進行案

時間	内容
導入	
(5分)	普段目にしてる新聞、雑誌などの印刷物において、見出し、本文、囲み記事などの場所ごとに特定の「スタイル」が使われていることを気づかせる。
展開	
(5分)	HTML においては、文章の構造をマークアップしていることを説明し、それにどのような利点/必然性があるのか考えさせる。(WYSIWYG との対比)
(5分)	HTML の記法について概要を説明し、なぜそのような記法(標準)に従つることが重要なのかを考えさせる。
実習	
(20分)	例題として提示した HTML をその通り打ち込んで表示させ、タグが混ざつたテキストファイルがブラウザによって整形され表示されることを確認させる。
(10分)	打ち込んだ HTML ファイルに自分独自の変更をさせたり、内容を追加させ、ブラウザの表示がその通り変化することを確認させる。
まとめ	
(5分)	HTML という規則(標準)に従つてテキストファイルを用意しておく、ブラウザがそれに従つて整形表示してくれること、その表示方法はブラウザが適切に選んでくれることを述べてまとめる。

B 授業案: さまざまな表現方法の工夫

B.1 目標

スタイルシートによる統一的なスタイル指定方法が分かる。目立つのではなく読みやすいスタイルが重要であることが分かる。

B.2 評価の観点

- RGB による色の指定方法が分かっている。
- スタイル指定によって一括してページスタイルが設定できることを理解している。
- 読みやすいページになるような色づかいや要素の配置方法などが分かっている。

B.3 留意点

- 極端な色づかいや飾り指定を試してみせて、目立つことと読みやすいことは違うことを理解させる。
- 原色ではなくうまく中間色を使った配色が見やすいことを納得させる。
- 自分なりに個性を出せて見やすいようなスタイルをそれぞれに探求させる。

B.4 進行案

時間	内容
導入	
(5分)	新聞や雑誌のスタイルの話に戻って、それぞれの個所では文字の大きさや色づかいなどが統一されている、Web ページでもそれがやりたいという話をする。
展開	
(5分)	HTML 3.2 では文字の大きさや色や配置をタグで指定していたという話をし、それは何がよくなかったかを考えさせる。
(5分)	CSS による指定方法の原理と具体的な書き方を説明する。前回の HTML ソースにスタイル指定をつけ加えたものを見せて解説する。
実習	
(20分)	前回打ち込んだ HTML に CSS 指定をつけさせる。最初は例題の通りにつけてもらい、ある程度分かったら自分独自に変えてみるようにさせる。
(10分)	CSS 指定をいろいろ変えてみながら、どのような考え方でスタイルをつけるのがよいページなのか考えさせる。意見を出して議論させる。
まとめ	
(5分)	CSS を使うと統一的なスタイル指定ができること、どのようなスタイル指定をつけるかはページの「見やすさ」中心で決めるべきことをまとめる。

C 模擬授業資料: さまざまな表現方法の工夫

C.1 文書の統一的なスタイル

みんなが普段見ている雑誌や本などを見直してみよう

- 見出しは何レベルあるか
- それぞれの見出しはどれくらい大きい字が使われているか
- 地の文はどれくらいの字詰めで組まれているか

- 統一的に使われている「飾りもの」は何があるか
- 図や写真はどのようなスタイルで入っているか
- カコミ記事のようなものはどうなっているか

C.2 文書の統一的なスタイル (2)

このようにスタイルが統一されていると、どんないいことがある？

- 各ページの内容が見てとりやすい
- どのページを開けたときでもどこに何があるか分かる
- ページごとにスタイルを決める手間がはぶける
- ...(ほかに?)

C.3 では、HTML ではどうなのか？

HTML 3.2 — 見え方を指定する機能が色々追加された

- 文字の色 — `...`
- 文字の大きさ — `...`
- 中央そろえ — `<center>...</center>`
- 地の色 — `<body bgcolor="色">...</body>`

しかしこれではよくないとされて、HTML 4.01 ではやめた。なぜだと思う？

C.4 HTML とスタイルシート

- 見え方を替える場所ごとに同じ指定を繰り返す必要
- たとえば「見出しだけ抜きだして利用」と考えた時など不便
- 1つのHTMLをさまざまな用途に使いたい→さまざまな見え方にしたい

そこで、、、

- HTMLは「文書の論理的な構成のみを指定し、見え方は指定しない」ことにした。
- 文書の表現方法については「スタイルシート」と呼ばれる別の指定方法を用いることにした。→CSS(Cascading Style Sheet)

C.5 CSSを使ったページの例

```

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN">
<html lang="ja">
<head>
<meta http-equiv="Content-style-type" content="text/css">
<title>CSS Example</title>
<style type="text/css">
body { background-color: white }

```

```
h1, h2 { text-decoration: underline }
ul { margin-left: 1cm; margin-right: 1cm }
pre { margin-left: 1cm; background-color: rgb(255,200,180) }
</style>
</head>
<body>
<h1>スタイルシートを使って表現を指定する</h1>
```

```
<h2>スタイルシートとは</h2>
```

<p>「見出しはこういう表現」「段落はこういう表現」というふう
に、文書の論理的な要素ごとに表現を指定します。</p>

```
<h2>スタイルシートでできること</h2>
```

```
<ul>
<li>文字の色や背景(地)の色を指定できる</li>
<li>文字飾り(下線や点滅など)を指定できる</li>
<li>左右のマージン(余白)などを指定できる</li>
<li>枠つけや枠のスタイルを指定できる</li>
</ul>
```

```
<h2>書き方</h2>
```

<p>次の形のものをヘッダ (<head>…</head>の中)に指定します。</p>

```
<pre>
<style type="text/css">
要素の指定 { 性質: 設定値; 性質: 設定値 … }
要素の指定 { 性質: 設定値; 性質: 設定値 … }
…
</style>
</pre>
</body>
```

C.6 範囲指定のマークアップ

「すべての見出しはこう」ではなく、特定の見出しをこうしたい、という場合にはその要素の開始タグに「class="名前"」を指定する(この名前を「クラス名」といい、分かりやすいように自分で決める)。

また、見出しなど既にある範囲ではなく、たとえば「見出しと段落の集まり」といった単位を指定したければ、その範囲を div 要素にする。

```
<div class="名前">
…
</div>
```

同様に、段落等の中の特典部分を指定したければ

```
<span class="名前">...</span>
```

で指定する。そして、スタイルの方は次の形で指定する。

. 名前 { 指定… } ←そのクラス名のものすべて

div. 名前 { 指定… } ←要素の種類まで指定

これを使った例も次に示します。

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN">
<html lang="ja">
<head>
<meta http-equiv="Content-style-type" content="text/css">
<title>CSS Example</title>
<style type="text/css">
body { background-color: white }
h1, h2 { text-decoration: underline }
ul { margin-left: 1cm; margin-right: 1cm }
pre { margin-left: 1cm; background-color: rgb(255,200,180) }
.important { color: red }
div.sec { border: blue solid 2px; margin: 1mm; padding: 2mm }
</style>
</head>
<body>
<h1>スタイルシートを使って表現を指定する</h1>

<div class="sec">
<h2>スタイルシートとは</h2>

<p>「見出しはこういう表現」「段落はこういう表現」という
ふうに、文書の<span class="important">論理的な要素</span>
ごとに表現を指定します。</p>
</div>

<div class="sec">
<h2>スタイルシートでできること</h2>

<ul>
<li>文字の色や背景(地)の色を指定できる</li>
<li>文字飾り(下線や点滅など)を指定できる</li>
<li>左右のマージン(余白)などを指定できる</li>
<li class="important">枠つけや枠のスタイルを指定できる</li>
</ul>
</div>

<div class="sec">
<h2>書き方</h2>

<p>次の形のものをヘッダ (&lt;head&gt;…&lt;/head&gt; の中)
```

に指定します。</p>

```
<pre>
<style type="text/css">
要素の指定 { 性質: 設定値; 性質: 設定値 … }
要素の指定 { 性質: 設定値; 性質: 設定値 … }
…
</style>
</pre>
</div>
</body>
```

C.7 実習 2

1. スタイルシートの例題 HTML を次の場所からコピーする。

```
cp ~kuno/work/css1.html css1.html ←名前は変えてもいい
cp ~kuno/work/css2.html css2.html
```

2. ブラウザでコピーしてきたファイルを開き、見え方を確認する (2つのファイルのうち好きな方でよい)。
3. そのファイルをエディタで開き、CSS の部分を修正する (たとえば色を変えてみるなどする)。エディタで保存したらブラウザの「再読み込み」で見え方の変化を確認する。
4. 変化することが分かったら、自分なりにいいと思うように見え方を工夫してみる。別添資料に CSS で使える機能の抜粋を示してあるので参考にするとよい。隣の人どうし見せ合ってみる。
5. どのようなスタイルのつけ方がよいか、自分なりに考えてみる。また、隣の人と意見交換してみる。

C.8 まとめ

- CSS を用いることで、HTML は同じままで色々な表現ができる。
- 実は<style>...</style>の部分を別のファイルにしておく方法もある。そうすれば1つのスタイル指定でいくつでもページを制御できるし、1個所変えるだけで全部変更できる。
- つまり、スタイルシート→自分のページに統一的なスタイルをつけられる、という利点。
- どのようにスタイルをつけるか→見やすく利用しやすいページ
- 目立つことを目標にしない方がよい…「ユーザビリティ」とも言う。また別の時間に改めて学ぶ。

C.9 添付資料: CSS の機能の抜粋

C.9.1 長さの指定方法

CSS では「要素の大きさ」「字下げの幅」などはすべて単位をつけた長さとして指定する。長さの指定には、1mm(ミリメートル)、1px(ピクセル…画面上の点の大きさ)、10%(外側—たとえばページ幅—に対する割合)などの指定方法がある。

C.9.2 色の指定方法

文字や背景の色を指定する場合、名前または RGB を使って指定する。名前で指定できる色として、「aqua, black, blue, fuchsia, gray, green, lime, maroon, navy, olive, purple, red, silver, teal, white, yellow」の 16 種類がある。これ以外は RGB の 3 色を混ぜて指定する。形式としては「#rrggbb」(RGB をそれぞれ 2 桁の 16 進数で指定する)と「rgb(赤, 緑, 青)」(RGB をそれぞれ 0~255 の 10 進数で指定する)とがある。たとえば「rgb(0,0,0)」はどの色も全く光らないから真っ黒、「rgb(255,255,255)」はその反対で真っ白、「rgb(255,0,0)」は真っ赤になる。実際にはその中間のてごろな色を試して決める。

C.9.3 ファイルや URL の指定

背景画像などファイルを指定する場合は「url(ファイル名)」という書き方を使う。

C.9.4 代表的なプロパティ

CSS でそれぞれの要素に指定できる代表的なプロパティを挙げておく。

- color: 色 — 文字の色を指定する。
- background-color: 色 — 地の色を指定する。
- background-image: url(ファイル名) — 背景画像を指定する。
- font: 指定… — フォントの指定。大きさを表す名前またはパーセント値、フォント名などが指定できる…が、Unix 上の Netscape 4.x ではほとんど対応していない。
- text-indent: 長さ — 段落の 1 行目の字下げ量を指定。
- text-align: 指定 — left、right、center のいずれかで左揃え、右揃え、中央揃えを指定。
- text-decoration: 指定 — underline、overline、line-through、blink、none が指定できる。
- margin-top:、margin-bottom:、margin-left:、margin-right: 長さ — 上下左右の余白を指定。
- padding-top:、padding-bottom:、padding-left:、padding-right: 長さ — 上下左右の詰めものを指定。余白と詰めものの違いは枠を引いてみると分かる。枠の外側が余白、内側が詰めもの。
- border: 指定 — 枠の指定。色、種類(solid、double、groove、ridge、inset、outset、dashed、dotted、none のいずれか)、および幅を指定できる。
- float: 指定 — left または right を指定。これを指定した要素は左または右によせられ、空いた側にはテキストが流し込まれる。

D 授業案に使う HTML のひな型

以下に付録 C に出した授業案の HTML ファイルを載せておく。皆様が授業案を HTML で書く時はこれを参考にどうぞ(色などのスタイルは見づらくならない範囲で変えてもよいです)。

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN">
<html lang="ja">
<head>
<meta http-equiv="Content-type" content="text/html; charset=euc-jp">
<meta http-equiv="Content-style-type" content="text/css">
<title>授業案: さまざまな表現方法の工夫</title>
<style type="text/css">
```



```
body { background-color: white }
h1, h2 { text-decoration: underline }
table { background-color: rgb(220,255,255) }
tr.header { background-color: pink }
tr.section { background-color: rgb(180,220,255) }
div.author { text-align: right; color: green }
</style>
</head>
<body>
<div class="author">2002.10.16 00112233 久野 靖</div>
```

<h1>授業案: さまざまな表現方法の工夫</h1>

<h2>目標</h2>

<p>スタイルシートによる統一的なスタイル指定方法が分かる。目立つのではなく読みやすいスタイルが重要であることが分かる。</p>

<h2>評価の観点</h2>

RGB による色の指定方法が分かっている。
スタイル指定によって一括してページスタイルが設定できることを理解している。
読みやすいページになるような色づかいや要素の配置方法などが分かっている。

<h2>留意点</h2>

極端な色づかいや飾り指定を試してみせて、目立つことと読みやすいことは違うことを理解させる。
原色ではなくうまく中間色を使った配色が見やすいことを納得させる。
自分なりに個性を出せて見やすいようなスタイルをそれぞれに探求させる。

<h2>進行案</h2>

```
<table border="2" summary="class plan" style="width: 100%">
<tr class="header"><td style="width: 10%">時間</td><td>内容</td></tr>
<tr class="section"><th colspan="2">導入</th></tr>
<tr><th>(5分)</th><td>
```

新聞や雑誌のスタイルの話に戻って、それぞれの個所では文字の大きさや色づかいなどが統一されている、Web ページでもそれがやりたいと

いう話をする。

</td></tr>

<tr class="section"><th colspan="2">展開</th></tr>

<tr><th>(5分)</th><td>

HTML 3.2では文字の大きさや色や配置をタグで指定していたという話をし、それは何がよくなかったかを考えさせる。

</td></tr>

<tr><th>(5分)</th><td>

CSSによる指定方法の原理と具体的な書き方を説明する。前回のHTMLソースにスタイル指定をつけ加えたものを見せて解説する。

</td></tr>

<tr class="section"><th colspan="2">実習</th></tr>

<tr><th>(20分)</th><td>

前回打ち込んだHTMLにCSS指定をつけさせる。最初は例題の通りにつけてもらい、ある程度分かったら自分独自に変えてみるようにさせる。

</td></tr>

<tr><th>(10分)</th><td>

CSS指定をいろいろ変えてみながら、どのような考え方でスタイルをつけるのがよいページなのか考えさせる。意見を出して議論させる。

</td></tr>

<tr class="section"><th colspan="2">まとめ</th></tr>

<tr><th>(5分)</th><td>

CSSを使うと統一的なスタイル指定ができること、どのようなスタイル指定をつけるかはページの「見やすさ」中心で決めるべきことをまとめる。

</td></tr>

<h2>工夫・意見・感想…</h2>

<p>あとはお好きなように…</p>

</table>

</body>

E 出席、および次回までの宿題

最初に述べたように、出席として紙に感想等を書いて出してもらいます。書く内容は板書します。必ず、「日付、学籍番号、氏名」を先頭に書くこと。また、今回については『情報』の免許について「取得予定です」/「取得しません」のいずれかを明記してください。

次回までの課題ですが、とりあえず、次回模擬授業の個所(当たった人が決めたところ)の授業案を全員にHTMLで書いてもらいます。付録Dのサンプルのようなものを考えてもらえばいいです。ただし、必ず最後に「この案をまとめる上でどういう所に苦心した/工夫をこらした」かを(HTMLの一部として)書いてください。そのほか意見・感想もあれば書いてください。授業時に述べたように、htmlintでチェックし、重大なエラーがないようにすること。そのHTMLファイルをメールで久野(メールアドレス kuno)あて提出すること。提出期限は次回授業前日 23:59 までとする。

重要! 誰の提出物か分かること、および「代筆」等の防止のため、そのメールは必ず学芸大学センターのアカ

ウントから送られたものでなければならない。時間ギリギリ等で自宅などから出した場合は一時預りとし、後日上記アカウントから送り直して貰った時点で正式に受理する(その場合は遅刻とはしない)。

いずれにせよ、メール冒頭に提出日時、学籍番号、氏名を明記すること。HTML部分はそのまま相互に見られるように WWW で公開するので、氏名を出したくなければ学籍番号だけ書いておいてください。