

情報科学演習

第6回

WWW と HTML

目次

1	本日の目標	1
2	用語の解説	1
3	本日の実習	1
3.1	HTML について	2
3.2	テキストベースブラウザを使う	4
3.3	スタイルシートを書く	4
3.4	Web ページを公開してみる	5
3.5	残りの時間の課題: Web ページの改良とバックアップ	6

1 本日の目標

- HTML の基本構造を知る.
- コンピュータ内の文書は, 人間とコンピュータ (ソフトウェア) の両方が読む事を意識する.
- 文章の構造と見栄えの分離を理解する.

これから, 学生生活の上でも, あるいはその後社会人になってからも, 皆さんには, 「文書の作成」という業務が必然的に要求されます. 例えばそれは, レポートの作成であったり, 論文の作成であったり, 報告書の作成です. これらの作成には, 多くの場合コンピュータが使われます.

皆さんが作成する文書は, 単純に「文字列の並び」ではなく, それなりの構造を持っています. 例えば, 表題があったり, 章があったり, 節があったり, 結論部分があったりです.

コンピュータでこのような文書を作成する時には, その文書の構造に従った補助機能が利用できます. というより, むしろ積極的にその機能を利用すべきです. ワードプロセッサでは, それはマウスを使った指示 (あるいはそれと同値なキーボードショートカット) で通常は実現されています.

この講義では, 「文書の構造の記述をテキスト (文字列) である」と言う事を体験していただきます. この講義では HTML を使いますが, 2 年次の「計算機概論 I」では, 数式の記述で絶大な威力を発揮する $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ で同じような内容を習う予定です.

2 用語の解説

WWW World Wide Web の略. 世界最初の Web browser が CERN(Conseil Européen pour la Recherche Nucléaire, 現在は L'Organisation européenne pour la recherche nucléaire, <http://www.cern.ch/>) で 1989 年頃から, Tim Berners Lee によって開発されたが, そのアプリケーションが WorldWideWeb.app (<http://www.math.u-ryukyu.ac.jp/~suga/www.gif>) 開発者に敬意を表して, World Wide Web, WWW, W3, Web 等という. 日本で, 「インターネット」という言葉が使われている事が良くありますが, インターネットとはネットワーク同士をつなぐ事で, Web を見る以外にももっと大きな内容を持つ言葉です.

HTML Hyper Text Markup Language の略. Hyper Text というのは相互参照をする文書の集合. World Wide Web で用いられる相互参照を記述するための言語. 現在は非営利団体 W3C (World Wide Web Consortium) で仕様が決められている.

http Hyper Text Transfer Protocol の略. Protocol とはここでは通信規約, すなわちネットワーク通信に於ける様々な約束事の事.

Web browser HTML 書かれた内容を解釈して, 画面に表示するためのソフトウェア. この講義では主に Firefox を使いますが, 過去に開発された Web browser が <http://browsers.evoltt.org/> にまとめて置かれています. また, 今日は w3m というブラウザも使います.

3 本日の実習

最初の講義で紹介した本,

「ユニバーサル HTML/XHTML, 神崎正英著, 毎日コミュニケーションズ」

をこのテキストでは、「参考書」と書いてあります。自分で購入するなり、図書館で借りるなりして、今後、必要な時には参照するようにして下さい。

3.1 HTML について

まずは復習です。emacs を起動して以前の講義で作成した index.html をバッファに読み込んで下さい。それができたら、次はホームフォルダの index.html をダブルクリック (あるいは、Firefox のウィンドウに index.html をドロップ) して firefox で表示して、両方を見比べて下さい。

Emacs で見える内容と firefox で見える内容が異なります。どちらも開いているのは index.html というファイルです。同一の内容のデータ (ファイル) でもソフトウェアによって解釈が異なり、表示が違って来る事を良く理解しておいて下さい。

このことは文字コードでも同じです。ASCII コードでは文字の 1 は数の 49(10 進) です。コンピュータ内の 49 という数を ASCII コードと解釈すると文字の 1 になるのです。

HTML は、Hyper Text Markup Language の名が示す通り、一種の (Markup = 印付けのための) 人工的な言語です。Hyper Text というのは相互参照を持つ文書の集まりの事で、1960 年代にはこの概念は存在していました。この概念を考案したのは Douglas Engelbart という人で、マウスの発明家でもあります。この概念に Hyper text という名前をつけたのは Ted Nelson という人のようです。(詳しくは、参考書第 1 章 1.1, 1.2 を読んで下さい。) HTML 以前にもハイパーテキストを扱うソフトウェアは存在していました (有名なのは Apple の HyperCard)。HTML がこれだけ浸透したのは、ネットワーク経由でのアクセスを想定した事とともに、インターネットの発展と時期を同じにしたと言う理由です。

HTML は文書の内容とともに「文書の相互参照と論理構造」を記述していきます。Browser は、HTML に書かれた論理構造に従った表示をするように設計されています。従って、例えばもとの文章にあった改行は、browser でみると無視されます。改行は論理構造に関係ないからです。

もう 1 つ注意して欲しいのは、HTML は、テキストファイルを利用している事です。前回、テキストファイルと画像ファイルの大きさの比較をしましたが、大きさの桁が違っています。最近ネットワーク通信のスピードが速くなりましたが、それでも他のデバイス (CD やハードディスク) から比べると遅いものです。その通信に、firefox で表示されている index.html の形の画像を送っていたのでは、効率が悪すぎます。実際、index.html の大きさは 1KB 程度ですが、firefox で表示されるものを画像にすると、例えば PDF ファイルでは、50KB 程度になり、約 50 倍の大きさになります。そこで、通信データは少なくし、表示する方で工夫するという「分散処理」の考え方が、ネットワーク通信では基本です。

HTML にはいくつかのバージョン (版) がありますが、この講義では現時点での最新バージョン 4.01 に基づいた話をします。¹ このバージョンでは相互運用性という事が重視されており、見た目ではなく、browser や使う人に対する依存性をなくする事が重視されています。

HTML では、論理構造と相互参照はタグ (tag) によって記述されます。タグは不等号 <, > で囲まれます。タグは基本的に開始タグと終了タグがあります。例えばパラグラフ (段落) の開始タグは、<p> でそれに対する終了タグは</p>です。タグは大文字小文字の区別がありません。<html> と<HTML> は同じ意味です。

¹今年中に Version 5.0 が策定される予定です。

index.html で使われているタグ

詳しくは、参考書 2.1, 2.2(pp 13-23) を見て下さい。

`<html>` - `</html>` HTML 文書全体の開始と終了. `lang="ja"` は、このページが日本語を使っている事を宣言しています. タグには、その要素に対して固有の役割を持たせる事が出来ます. これを「属性値 (attribute)」と言います. 属性値は、属性名="値" という形 (=と"で囲む) で指定します.

`<head>` - `</head>` Header(ヘッダ, 適切な日本語訳は無い). ブラウザやサーチエンジンなど、この文書を利用するソフトウェア (まとめてユーザーエージェントという) に対して、補助的な情報を記述する.

`<meta ... >` `meta` タグ. Header の中で文書の付加的な情報を記述し、ユーザーエージェント (ブラウザやサーチエンジン) に対して、その動作を指示するために使う. `index.html` では、文書の文字コードの情報を与えるものを記述した. 終了タグは無し.

`<link ... >` 上と同様、ユーザーエージェントに対してこの文書とリンクしてるファイルや、補助的に用いるファイルを記述する. 終了タグは無し. `index.html` では、スタイルシート (後述) の指定をしてある.

`<title>` - `</title>` 文書の表題. グラフィカルなブラウザでは、タイトルバーにこの文字列が現れる事が多い.

`<body>` - `</body>` 文書の本体 (本来の内容)

`<h1>` - `</h1>`, `<h2>` - `</h2>` 見出しタグ. 数字は、見出しのレベル. `<h1>` から `<h6>` まで 6 つのレベルがある. グラフィカルなブラウザでは、レベルに応じて文字の大きさが変わるが、文字の大きさを変えるために使うのではなく、文書の中での意味としての見出しとして用いる.

`<hr>` 水平線を引く. 閉じタグは無し.

`<p>` - `</p>` パラグラフ (段落). 終了タグは省略可.

`` - `` Unordered list(番号の付かない箇条書). 箇条書に付いては、後の講義でもう 1 度取り上げる.

`` - `` List item(箇条書の中のリスト)

index.html で使われているタグでない html の要素

`<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01//EN">` 文書型定義 (Document Type Definition) HTML のどの規格に基づいた文書であるかの記述.

`<!-- ... -->` コメント. 複雑な処理等をする場合に、それを作った人が、処理内容などについてのメモ書きを残すための仕組み. 多くの場合、中身を改変する際に自分自身へのメッセージとなる. ブラウザはこの部分を完全に無視する.

コメントという仕組みが最初は理解できないかも知れませんが、プログラミング言語や、表計算ソフト等複雑な処理をする処理系では、必ず 1 部分を処理系が無視をしてくれるようにする仕組みがあります. これは、その処理内容を読む人のための仕組みです.

3.2 テキストベースブラウザを使う

Web ブラウザは Firefox や Safari の様なグラフィカルなものではありません。前回もいいましたが、グラフィカルな環境が使えない事もあります。例えば、この講義の Web ページや情報処理センターの Web ページには、個人情報保護の観点やセキュリティ上の問題から、大学外からの閲覧を制限してる情報があります。では、それらは本当に大学外から見えないかというと、琉球大学のアカウントとネットワーク環境とテキストベースブラウザがあれば、読む事ができるのです。(実際には、学外から利用可能な情報処理センターのマシンに、テキストベースブラウザが入っていないのでダメですが。)

ここでは、Vine linux に入っている w3m というテキストブラウザを使ってみます。(w3m = WWW wo miru の略、当然日本人の作品です。) Vine linux の環境では、絵や写真も見えるように拡張されていますが、本来はそのようなものの表示を目的としません。次の操作をして、w3m を使ってみて下さい。

1. Gnome 端末を起動する。
2. w3m index.html (w3m と index.html の間には空白を入れる。) と押して ENTER キーを押すと、w3m が起動して、今まで作った index.html が読めます。

カーソル移動は、カーソルキー (矢印の付いたキー) でできます。あるいは、Emacs と同じ C-b, C-f, C-p, C-n でもできます。

他の URL に移るには、大文字の U (Shift + U) キーを押します。最下段に Goto URL: とでますから、URL (例えば <http://www.math.u-ryukyu.ac.jp/~suga/joho/2010>) 入力して ENTER を押すとそこに飛びます。リンクのページに飛ぶには、そのリンクの場所にカーソルを持って行き、ENTER を押せばリンク先に飛びます。1 つ前のウェブページに戻るには、大文字の B キーです。

終了は小文字の q キーで、押すと終了するかどうかを尋ねられますから y を押します。

3.3 スタイルシートを書く

HTML は、文章の内容を論理構造を表すタグとともに記述するように考えられており、その見栄えや表現方法は browser に任せるようにしています。これは、Web をアクセスする人が、さまざまな環境である得る事を想定した仕様で、当然の事です。

しかし、多くの場合 Web にアクセスするのはグラフィカルな browser で、文書の多彩な表現が可能です。このような環境に対して内容の色や配置を指定するには、スタイルシートを利用します。この講義では、最も標準的に用いられている カスケイディングスタイルシート (Cascading style sheet, CSS) を利用します。スタイルシートには、CSS の他にも XSL (Extensible Stylesheet Language) などいくつかの種類があります。

emacs を起動して、下の 4 行の内容を index.css という名前のファイルで保存して下さい。

```
body {background-color: rgb(224,255,224)}
h1 {color: maroon; text-align: center}
h2 {color: #00FF00}
p {text-indent: 1ex}
```

保存が終わったら、Firefox で再読み込みボタン (青色の円形の矢印のついたもの) を押して、表示がどのように変化するかを見て下さい。

もし何も変化しなければ、ファイル名が index.css となっているか、index.html の上から 7 行目付近の `<link rel="stylesheet" type="text/css" href="index.css">` の部分が間違っていないかを確認して下さい。

CSS は、次のような形で記述されます。

セクタ { 種類: 指定値; 種類: 指定値; ... }

何の どの部分を どの値に ...

複数の部分を指定する場合には、; で区切って書きます。例えば上で、

```
h1 {color: maroon; text-align: center}
```

としてありますが、第 1 レベルの見出しの色は maroon にし、そのテキストは画面の中央に配置すると言う意味です。

また、色指定は 3 通りを例示してあります。rgb(224,255,224) と #E0FFE0 は同じ意味になります。どちらも光の 3 原色赤 (red)、緑 (green)、青 (blue) の輝度を、数値で指定しており、始めの方は 10 進で分離して書いてあり、後の方は 16 進でまとめて書いてあります。輝度の値の範囲は $0 \sim 255 (= 2^8 - 1)$ です。代表的な色は、色の名前 (例えば maroon=栗色) で指定する事も出来ます。html で定義されている色の名前はそう多くなく 16 色で、参考書の p79 にあります。これ以外の色の名前の定義されており、グラフィカルなブラウザでは、その名前を使ってもほとんどの場合は、大丈夫です。ただし、色の名前については自分で調べて下さい。

最後のセクタ p の指示での text-indent は、段落の最初の字下げの指定です。1ex は 1 文字と言う単位の指定です。単位の指定方に関する詳しい事は、参考書を見て下さい。

スタイルシートを書いた後のページも、w3m を使って見て下さい。文字の色を変えるとか、文字を中央に置くとかの指示は、w3m では無視されます。ウェブページを書き換えた時には、必ず w3m でどう見えるかもチェックするようにして下さい。

HTML のタグやスタイルシートの指定などは英語がもとになっていますが、この程度の英単語の意味は、各自調べておくとともに、この際覚えてください。(会話よりもむしろこういうものの方が、使える英語だと私は考えます。)

3.4 Web ページを公開してみる

前回は言いましたように、今のままではネットワークを使って皆さんのウェブページにアクセスする事は出来ません。それが出来るようになるには、Web サーバと呼ばれるコンピュータの然るべき場所にファイルを置いて、アクセスできるように然るべき設定をする必要があります。

いわゆるプロバイダを利用する場合には、ファイル転送ソフトを用いて Web ページのファイルを転送しますが、琉球大学の情報処理センターでは、皆さんのホームディレクトリ (フォルダ) は、Web サーバからアクセスできるような設定になっています。ただし、ホームディレクトリの全てのファイルが、Web サーバ越しに見えたのでは、個人的なファイルが置けませんので特定のディレクトリ (フォルダ) 内だけが Web サーバ越しに見えます。情報処理センターの場合、ホームディレクトリの中の「WWW」(W は大文字) というディレクトリに HTML 関連ファイルを置いて、そのディレクトリが Web サーバのソフトウェアから読み取り可能にしなければなりません。具体的には、次のようにします。(ここからの作業は、Windows では不可能ではありませんが大変です。Mac

では、ファインダの「フォルダの情報を見る」から、所有権とアクセス権の詳細な情報と言う場所を見て、グループとその他に「読み出しのみ」を選ぶ事で、同様の事が出来ます。）

1. 「xxx のホーム」を右クリックしてプロパティを選ぶ。
2. 「アクセス権」のタブを選んで、右端の方にある「実行 (X)」の 3 箇所全てにチェックをいれ (最初は 1 箇所だけにチェックが入っている)、ウィンドウを閉じる。
3. 「xxx のホーム」をダブルクリックして、ホームディレクトリをファイルブラウザで開く。
4. ファイルブラウザのファイルメニューから「フォルダの生成 (F)」を選ぶ。
5. 「未タイトルのフォルダ」と言う新しいフォルダが出来るので、キーボードから「WWW」とタイプして、名前をつける。(W は大文字)
6. フォルダ「WWW」を右クリックして「プロパティ」を選ぶ。
7. アクセス権のタブを選び、グループ、その他の「書き込み (W)」のチェックが外れており、それ以外に全部チェックがついている事を確認し、確認したらプロパティのウィンドウを閉じる。
8. 今まで作った、index.html と index.css を WWW に入れる。(マウスの左ボタンを使って、上のファイルを掴んで、WWW の上に重ねて、ボタンを離す。)
9. フォルダ「WWW」をダブルクリックし、index.html のプロパティのアクセス権を見、所有者、グループ、その他の「読み込み (R)」に全てチェックがついている事を確認する (ついていなければ、マウスでクリックしてチェックをつける)。
10. index.css についても index.html と同じアクセス権になっている事を確認する。

上が全部終わったら、firefox で URI を入力する所に、<http://www.cc.u-ryukyu.ac.jp/~e1031xx> (1031xx は自分の学籍番号を入れる) と入力してみてください。それは、ネットワーク越しにアクセスしたページです。自分のページが見えたら、隣の人の学籍番号に変えてみてください。

ウェブページの公開を中止する

ウェブページを公開を中止するには、WWW というディレクトリを別の名前に変えると言うのが、1 つの方法です。

これ以外にも、ディレクトリ WWW のアクセス権で、グループとその他の読み込みと実行権を外すと言う手もあります。これを実際に行うと、ウェブブラウザで確かめてください。

特定の場所からはアクセスを許すが他からは許さないとか、その逆の設定をする場合には、.htaccess というファイルを作ります。これに関しては、後の講義で降れます。

3.5 残りの時間の課題: Web ページの改良とバックアップ

残った時間で、次の事をやって下さい。

1. index.html のプロフィールの下に簡単な自己紹介を付け加えるなどして、内容を充実させて下さい。ただし、あまり個人が特定できる内容は書き込まないで下さい。

2. バックアップ

index.html がディレクトリ WWW に移動したので、emacs で開く時は「WWW/index.html」でファイル名を指定します。ディレクトリの中のファイルを読むのに / を使う事は、知っておいて下さい。時間に余裕がある人は、index.css を変更してみてください。ただし、見づらい色使いにならないよう注意して下さい。文書作成時には、次の事も注意して下さい。

文書作成における注意

講義で何度か述べましたが、コンピュータで作成する文書は、人間以外にもコンピュータ (のソフトウェア) も読みます。日本語を使っている時に注意して欲しいのは、

1 単語の文字列に空白や改行を入れるな。

です。例えば「数学」を体裁を整えるために「数 学」と書くと、文書内での検索で、「数学」の文字列の検索に失敗します。目の不自由な人は、文章読み上げソフトなどを利用できるようになってきていますが、文章読み上げソフトは「かず がく」と誤判断してしまいます。

みなさんが使っている emacs では、80 文字 (全角で 40 文字) を越える行を入力しようとすると、自動的に改行するように設定されていますが、単語を考慮しての改行ではありません。したがって、画面の右の方に行がかかったら、適当な場所で自分で ENTER キーを打って、改行してください。

どうしても体裁を整えるために字間広げる時には、HTML の場合スタイルシートを使ってそのような指示をします。他の場合には、それに応じたやり方をします。例えばワードプロセッサだと、字間を広げると言う指示が出来るようになっています。

HTML 文書作成上の注意 (規格外と非推奨)

講義で紹介していないタグも、使いかたと意味を調べて積極的に使ってみてください。参考書の 2 章、3 章に詳しく書いてあるはずですが、いろいろなウェブページの HTML を参考にするのも 1 つの手です。Firefox の「表示」メニューから、「ページのソース」を選ぶと、今見ているページの HTML が表示されます。

ただし、WWW が成長して行く過程で、さまざまな独自拡張のタグが現れました。例えば「文字列を点滅させる」という独自拡張が以前はありましたが、現在では HTML 規格外です。最初の方でも述べましたが、HTML の考え方は文書の論理構造と相互参照なので、「文字列を点滅させる」のような概念は規格に入りません (文字列を強調するタグとして、 やがある)。規格外のタグを使うと、ブラウザでの表示は保証されませんから、使うべきではありません。HTML の (古い) 参考書の中には、このような内容のことを書いてあるものもありますので、本を選ぶときには注意してください。指定した参考書には、正確なタグの解説が書いてありますので、使う前に参考書の解説をまず読んで下さい。

同様に HTML 文書内で「文字の大きさや色を変更する」と言うような属性指定は、非推奨です。これも文書の論理構造や相互参照と関係ないからです。HTML は目印をつける (markup) ための物で、レイアウトやデザインを整えるための物ではありません。また、上で述べた相互運用の概念に反しています。目の見えない人、色の区別のつかない人が世の中にはいらっしゃいますが、これらの人たちへの配慮も必要です。目の見えない人は文書を点字印刷したり、音声に変換して読む事ができます。色の区別が付かない人は、スタイルシートを無効にして自分に都合の良い表示で読む事ができます。レイアウトや配置は、可能な限りスタイルシートを使うべきです。