情報科学演習第 7 回

WWW と HTML

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という考え方を理解する.

2 ご託宣

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」と言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, 最近では通常コンピュータが使われます (数学のレポートだと数式をきちんと書くのが面倒なので, 手書きでも OK なことが多い).

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して文書を作成します. 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. ワードプロセッサを使う際の最大の問題点は, 文章構造を意識するところが曖昧になる事です. なぜ文字の大きさや形を変えるかという理由を, 論理的に意識すべきなのです.

この講義では, 「文書の構造の記述をテキスト (文字列) でする」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. 数式入力では, Word でも Apple の Pages でも, $\text{T}_{\text{E}}\text{X}$ スタイルで入力するのが通常となったようです.

3 用語の解説

WWW World Wide Web の略. 世界最初の Web サービスが 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 browser の名前が WorldWideWeb.app (<http://www.math.u-ryukyu.ac.jp/~suga/www.png>). 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という意味で, Web を見る以外の, 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. 非営利団体 W3C (World Wide Web Consortium) で仕様が決められていたが, 現在は, WHATWG(Web Hypertext Application Technology Working Group) が, HTML Living Standard の仕様を決めている.

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. 過去に開発された Web browser が <http://browsers.evolt.org/> にまとめて置かれています. スマートフォンだと, iPhone は Safari, Android 系は Google Chrome が標準的な Web browser です. PC では, Mac だと Safari, Windows だと Microsoft Edge, Chrome OS なら Chrome, Linux だと Firefox あたりが標準的な Web ブラウザです.

4 本日の実習

4.1 HTML について

まずは復習です。前回作成した index.html をダブルクリックして、Web ブラウザで表示して下さい、さらに index.html を右クリックしてテキストエディタで開いて、両方を見比べて下さい。

エディタで見える内容とブラウザ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。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。ブラウザは、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、ブラウザ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。また、論理構造に従って文字の大きさを変えたり、改行したりリストの中黒・をつけたりしています。

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。最近ではネットワーク通信のスピードが速くなりましたが、それでも内蔵ストレージなどの直接繋がっているデバイスから比べると遅いものです。その通信に、ブラウザで表示されているような 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(キロバイト) 程度ですが、ブラウザ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。通信回線は、多量のデータをやり取りするのに時間がかかる場合が多くあります。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。

さらに、データの検索でも、テキストファイルの方に利点があります。例えば、index.html の文書内で「特技」という文字を検索することは、コンピュータには簡単です。これが、画像ファイルであったとすると、コンピュータには、かなり難しい処理となります。

HTML はいくつかのバージョン (版) を経て現在に至りますが, この講義では 2014 年に確定した HTML5 に基づいた話をします. 現在では, HTML5 は廃止され, HTML Living Standard というのが標準になっているのですが, こちらは, 内容が変化するので追跡するのが大変です. ただし, HTML 5 からの変更はほとんど無く, 機能の追加がほとんどなので, HTML 5 を勉強して損はありません. HTML 4.0 の仕様策定の頃から, 見た目ではなく, ブラウザ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 (少数の例外を除いて) 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. index.html を見ればわかりますが, タグは入れ子構造で用います (ある開始タグと終了タグの間に別の開始タグと終了タグを入れる).

HTML の開発動機は, 効率的な情報共有です. この考え方は今も変わっていません. このことも頭の隅に入れておいて下さい.

index.html で使われているタグ (要素)

- `<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, 適切な日本語訳無し). 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 要素タグ. 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) 等に対して, 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. 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. 多くのブラウザはタイトルバーにこの文字列を表示する.
- `<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. 段落と言う概念を持っていない人が多いようですが, どんな文書も「段落の集まり」で形成されています. 文書を作るとき (今の皆さんだとレポート課題が多いと思いますが) には, 常に「段落」を意識して作ってください.
- `<ul>` - `</ul>` : 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 度述べる.
- `<li>` - `</li>` : 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> : 文書型定義 (Document Type Definition) HTML5 では、単純に HTML と書くだけ。以前は、HTML のどのバージョンかの記述があった。

<!-- ... --> : 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間のための仕組みです。

4.2 テキストベースブラウザ

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言うと、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。問題はどうやって大学のマシンを大学外から利用するかという事ですが、それについてはこの講義では触れません。興味のある方は調べて下さい。2 年次向けの計算機概論 I では、講義します。

4.3 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ブラウザ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。

テキストエディタを起動して、下の 4 行の内容を index.css という名前のファイルで、index.html がある場所と同じフォルダに保存して下さい。

```
body {background-color: rgb(224,255,224)}
h1 {color: maroon; text-align: center}
h2 {color: #000088}
p {text-indent: 1em}
```

保存が終わったら、ブラウザの再読み込みボタンを押して、表示がどのように変化するかを見て下さい。(再読み込みボタンが分からなければ、一旦ブラウザを終了して、index.html をもう一度ダブルクリックで開く。)

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。iPad + Koder の場合は、残念ながら、そのままではスタイルシートの内容は反映されないようです。

CSS は、次のような形で記述されます。

```
セレクタ {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ありません。

最後のセレクタ p の指示での text-indent は、段落の最初の字下げの指定です。1em 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。

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。)

4.4 Web ページの公開

前回も言いましたように、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、大学内からですと、比較的簡単にファイルを転送できますが、学外からだとやはり「ファイル転送ソフト」が必要です。これについては、後の講義で述べる予定です。

4.5 課題: Web ページの改良

次回までに、index.html のタグの記述をチェックして、間違いがないようにして下さい。特に閉じタグを間違えたり、忘れたりする場合がありますので注意して下さい。

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。

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。Web で調べれば色々出てくるはずですが。(良い Web ページが少ないので困るのですが...)

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。

HTML5 についても幾つかの解説書がでています。どの本が良いのかは、今のところ判断ができませんが、「初心者向け」を謳っている本に、「悪い本」の割合が高いのは、昔からです。

HTML は情報共有のための規格です。例えば、HTML 文書内で「文字の大きさや色を変更する」というような属性指定は、非推奨です。HTML5 では、HTML 本体に見せ方を指定せず、見せ方の指定はスタイルシートとブラウザにまかせます。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、スタイルシートを使うべきです。

最初の方で述べたように、HTML は、HTML5 から HTML Living Standard に移行しています。HTML Living Standard は HTML 5 を基礎としているので、今流通している主な browser の多くは、HTML 5 対応です。HTML5(や HTML Living Standard) のタグを(自分で調べて)利用する事は、積極的に行ってください。