生活社会科学科

基礎演習 IIIB テキスト

三木　邦弘

平成12年9月28日

目 次

1 Webページの作成 **2**
1.1 準備作業 ・・・・・・・・・・・・・・・・ 2
1.2 ファイル名とURL ・・・・・・・・・・ 2
1.3 HTML入門 ・・・・・・・・・・・・・・ 3
1.3.1 簡単な例 ・・・・・・・・・・・・ 3
1.3.2 基本タグ ・・・・・・・・・・・・ 3
1.3.3 文字の形式について ・・・・・ 6
1.3.4 リンクの付け方 ・・・・・・・・ 6
1.4 画像の指定の仕方 ・・・・・・・・・・ 7
1.5 画像の入力と編集 ・・・・・・・・・・ 9
1.5.1 画像ファイルの作成 ・・・・・ 9
1.5.2 画像ファイルの転送 ・・・・・ 12

2 会話的なWebページ **13**
2.1 プログラムの作成方法 ・・・・・・・・ 13
2.2 定型的なプログラムの起動法・・・・・ 15
2.3 プログラムの出力の取り込み・・・・・ 16
2.4 入力用ページの作り方 ・・・・・・・・ 17
2.4.1 Form: 形式とプログラムの指定 17
2.4.2 Input: ボタン入力 ・・・・・・ 17
2.4.3 Input: 短い文字列の入力・・・ 18
2.4.4 Input: 定形データを送る・・・ 18
2.4.5 Textarea: 長い文字列の入力・ 18
2.4.6 Select: 選択メニュー ・・・・・ 19
2.4.7 入力されたデータの表現方法 19

3 HTML中級編 **21**
3.1 画面の背景の設定 ・・・・・・・・・・ 21
3.2 文字の修飾 ・・・・・・・・・・・・・ 22
3.3 表の指定 ・・・・・・・・・・・・・・・ 22
3.4 画面の分割 ・・・・・・・・・・・・・ 23
3.5 表示先の指定 ・・・・・・・・・・・・ 24
3.6 動く画像 ・・・・・・・・・・・・・・・ 25
3.7 透明な背景色 ・・・・・・・・・・・・ 25

4 CGI中級編 **27**
4.1 ページ内容の自動更新 ・・・・・・・・ 27
4.2 Cookieによる利用者の識別 ・・・・・ 28
4.2.1 Cookieの設定 ・・・・・・・・・ 28
4.2.2 Cookieの読み取り ・・・・・・ 29
4.3 応用課題「みきっち」 ・・・・・・・・ 30
4.4 応用課題「簡易チャットルーム」 ・・ 34

1 Webページの作成

ここでは、ワークステーション上に自分のWebページ(ホームページ)を作るためのやり方を説明します。現在学内のいくつかのワークステーションでは各利用者が自分のホームページを作成する事が可能になっています。そしてそれは学内だけでなく、学外(もちろん海外も含む)からも見ることができるようになっています。どのような内容の物を作るのも自由ですが、自分の作ったものに対する責任は忘れないようにしてください。

簡単なページを作成するだけならば1時間もかかりません。高度な事を行おうとすると、それなりに学ばなければならないことが多くなりますが、むしろ内容を工夫することに勤める方が良いと感じています。

1.1 準備作業

自分で作って自分で見るだけならばどこのディレクトリーに作成しても良いのですが、後に出てくるcgi機能を使ったり、他の人からも見えるようにするためには、自分のホームディレクトリー(loginした時に居るディレクトリー)の下のwwwと言うディレクトリーの中に、関連するファイルを入れる必要があります。このディレクトリーは通常存在しないので作らなければなりません。また、このディレクトリーが他の人から見えるようにサーバーに設定もしなければなりません。以上の作業をするためには、次のようなコマンドを実行してください。

```
cc07[miwako]% w3setup
```

このw3setupの実行は毎回何か作業をする前に必要です。w3setupは必要ディレクトリーの作成だけでなく、ディレクトリーの移動など様々な設定の変更をするからです。cc07[miwako]%はmiwakoさんの場合で、通常は自分の登録名が[ ]の中に表示されます。

せっかく色々作成したが、もう公開はやめようと言う時には、

```
cc07[miwako]% w3close
```

を実行して下さい。作成したファイルは消えませんが、ディレクトリーの名前がwwwからxxxに変更されて他の人からは見ることができなくなります。再び公開する場合はw3setupを実行します。

1.2 ファイル名とURL

wwwの下にaaa.htmlと言うファイルを作成した場合のURLは、

http://www.center.sugiyama-u.ac.jp/グループ名/登録名/aaa.html

になります。このURLで海外からでもアクセスできます。

index.htmlと言う名前のファイルを作ると、このファイルのURLは少し短くなって次のようになります。

http://www.center.sugiyama-u.ac.jp/グループ名/登録名/

短いほうが伝えるのにも楽ですので、公開する場合にはindex.html と言うファイルを使うと良いでしょう。なおw3setupを実行したときにindex.htmlと言うファイルが存在しない場合は自動的に作成されるようになっていますので、その内容を適当に直してご利用下さい。また各自のindex.html に対してwww.center.sugiyama-u.ac.jpからのリンクが自動的に作られるようになっています。そのときの見出しとなる文章はtitleと言う名前のファイルに入れておいてください。

1.3 HTML入門

WWWのサーバーはクライアント(利用者)からの要求に従ってデータを送ります。クライアントはもらったデータを表示するのですが、そのデータはHTML (HyperText Markup Language)と言う言語で記述されています。WWWは予め用意してあったデータを送るだけなので、自分のデータを公開したい場合には、自分のデータをHTMLで記述する必要があります。ここではその概要を説明します。

1.3.1 簡単な例

例えば次のような内容を、aaa.htmlと言うファイルに入力してみましょう。

```
<HTML>
<Head>
<Title>簡単な例</Title>
</Head>
<Body>
<H1>これはレベル１の見出し</H1>
HTMLの世界へようこそ。
これは１番目の段落です。<P>
そしてこれは２番目の段落です。
</Body>
</HTML>
```

無事入力ができたら形式を検査するプログラムがあるので次のようにして実行してみます。

```
cc07[miwako]%jweblint aaa.html
```

何も出なければ合格です。何かメッセージが出た場合は問題があると言うことなので指摘された所を直してください。そしてInternet ExplorerやNetscape Navigator等のブラウザで見ると、見出しの所の字が大きくなっていたり、段落ごとに改行して、段落の間に少し隙間ができていたりします。

HTMLは文章中に様々なマークアップタグ (markup tags) を挿入して様々な指示を行います。この例では、< >で囲まれた部分がそうです。< の次にはタグ名が続きます。これは大文字と小文字の区別は無いので、<TITLE>の代わりに<title>と書いても構いません。タグ名が／で始まっているのは有効範囲の終わりを示します。</XXXX>は<XXXX>の終わりを示しています。通常のタグは全て終わりのタグと対になって使われますが、例外も幾つかあります。段落の切れ目を示す<P>等がその例です。

Explorer等で「ソース表示」をメニューで選ぶと、このHTMLの記述をそのまま見ることができます。どうも思ったような表示が得られないときに確認するのに利用できます。

1.3.2 基本タグ

ここでは、先程の例にも出てきた基本的なタグについて説明します。

- 全体：ここの記述はHTMLによるものだと言うことを示すものです。ファイルの最初と終わりに必ず入れます。

  ```
  <HTML>
  ...   HTMLでの記述   ...
  </HTML>
  ```

- 設定：ページの設定のような事を記述している部分がここにあることを示します。ここで説明している基本的なタグの中では<TITLE>だけがこの中に入ります。

```
<Head>
...  <Title>などの記述  ...
</Head>
```

- 表題：文章の表題を示すものです。通常本文とは別の場所に表示されます。また索引などの見出しに使われることもありますので、文章の内容を的確に示すものが望まれます。タグの形式は次のようなものです。

```
<Title>表題の文</Title>
```

- 本体：実際のHTMLの表示されるページに関する記述はここの中にします。

```
<Body>
...  HTMLでのページの記述  ...
</Body>
```

- 見出し：HTMLは、1から6までの6つのレベルの見出しが可能で、レベル1が一番大きな見出しになります。見出しとして指定された文は独立した左詰めの行として表示されます。タグの形式は次のようなものです。ただし、yの所は実際は1〜6の数字になります。

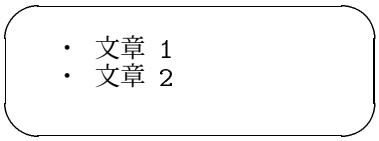
```
<Hy>見出しの文</Hy>
```

- 段落：何もタグの付いていない文章は、クライアント側の都合(通常画面の幅)に合わせて詰め込まれます。段落として独立させたい場合には、段落の切れ目に次のようなタグを付ける必要があります。

```
文文文...文<P>
```

- 番号なしリスト：この説明文のような•が先頭に付いた箇条書きをするためには次のようなタグを使います。

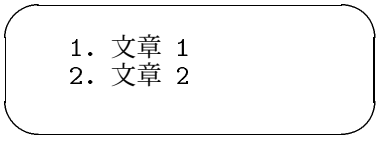
```
<UL>
<Li>文章 1
<Li>文章 2
</UL>
```



<LI>が•になる感じです。<LI>の部分は幾つでも構いません。また3重までの入れ子にすることも可能です。

- 番号付きリスト：先頭に1、2、3と数字が順番に付いた箇条書きをするためには次のようなタグを使います。

```
<OL>
<Li>文章 1
<Li>文章 2
</OL>
```



今度は<Li>が数字になる感じです。<Li>の部分は幾つでも構いません。また3重までの入れ子にすることも可能です。

- 定義型リスト：言葉ではちょっと説明しがたいものですが、次のような形にしたいときにこれを用います。

電子計算機
　　コンピュータのこと。
コンピュータ
　　かつて電子計算機と呼ばれたもの。パソコンの項を参照のこと。

これは、次のような3種類のタグを使って記述します。

```
<DL>
<DT>電子計算機
<DD>コンピュータのこと。
<DT>コンピュータ
<DD>かつて電子計算機と呼ばれたもの。パソコンの項を参照のこと。
</DL>
```

- 引用文：引用などで通常の文章よりも行頭が右に凹んだ文章を記述したいときには、次のタグを使います。

```
<BlockQuote>
文文文...文
</BlockQuote>
```

- 整形済み文章：既に整形が終わっていると言う事で、次のタグで囲まれた文章は入力したままの形で表示されます。つまり空白や改行が無視されませんし、勝手に改行されたりもしません。

```
<Pre>
                +- 食品栄養学科
生活科学部----+- 生活環境学科
                +- 生活社会科学科
</Pre>
```

この場合画面に表示されるのは、<Pre>のタグが無いだけで後は全く同じものです。

- 強制改行：段落を示すタグ<P>を使用すると段落の間に空行が入ります。それを避けたい場合には、次のようなタグを使います。

```
文文文...文<Br>
```

- 水平線：画面一杯の水平線を引くタグは次のようなものです。

```
<Hr>
```

- コメント：文章の説明的なもので、表示されては困るものは次のようなタグを付けておきます。

```
<!--  文文...文  -->
```

1.3.3 文字の形式について

通常の文字はそのままですが、< > & "の4文字は特殊な意味を持つためにそのまま使えません。それぞれ次のような形で記述します。

< → <
> → >
& → &
" → "

次のようなタグを付けることにより論理的な意味付けを文字に与えることが可能です。異なる意味付けのものはクライアントで色や書体の違いとして表示されます。

<Dfn>定義された語</Dfn>	通常イタリックで表示される。
<EM>強調された語</EM>	通常イタリックで表示される。
<Cite>本等の表題</Cite>	通常イタリックで表示される。
<Code>プログラムなど</Code>	通常等幅文字で表示される。
<Kbd>キーボードのキー</Kbd>	通常等幅の太字で表示される。
<SAMP>コンピュータの状態</SAMP>	通常等幅文字で表示される。
<Strong>強調された語</Strong>	通常太字で表示される。
<Var>変数など</Var>	通常イタリックで表示される。

また直接次のように直接字体を指定することも可能です。

<B>太字</B>
<I>イタリック</I>
<TT>等幅文字</TT>

1.3.4 リンクの付け方

他の文章へのリンクや同じ文章内にリンクを張ることができます。利用者はそこを指定するだけでそのリンクを張った先を見ることができます。このリンクが可能である点がハイパーテキストの由来だと言われています。

- 他文章へのリンク：これは次のような形式のタグを用います。

  <A href="URL">クリックされる文</A>

  URLの部分には実際にリンクする先の文章のURLが入ります。「クリックされる文」の所はブラウザでは下線が付いてちょっと色が異なる表示がなされます。ここはリンク先が判るような文にします。実際は例えば次の様な形になります。

  <A href="http://www.sugiyama-u.ac.jp/">椙山女学園大学のホームページ</A>
  <A href="betu.html">同じディレクトリーにあるbetu.htmlと言うファイル</A>
  <A href="/グループ名/登録名/friend.html">友人のfriend.htmlと言うファイル</A>

- 文書内へのリンク：予め次の様なタグ(アンカー)を入れておくと、そこへ行くリンクを張ることが可能です。

  文...文<A name="namae">文</A>文...文

  namaeは適当な語を使います。同じファイル中で同じ語は使えません。そしてリンクを張るときには次の様にします。

```
<A HREF="#namae">クリックされる文</A>
```

要するに先程指定した語の前に#を付けます。長めの文章で先頭の所に目次や索引を設けて、そこから後に続く文章の該当するところへリンクを張ると言う形でよく使われます。

この両者を同時に使う事も可能です。つまり他文書の中で予めアンカーを指定しておけば、リンクを張る側は、URLの後に#とアンカーで指定した語を書けば良いようになっています。

```
<A Href="http://cc01.center.sugiyama-u.ac.jp/~mailbase/student/index.html#ss">生社一覧</A>
```

演習問題

以下のような超簡易CAIを作れ。

1. caiq.htmlと言うファイルを作成し、この中には問題文と答えの選択子(4つぐ らい)を入れる。正解を選択したらcaio.htmlへ、間違った選択ならばcaix.htmlへ行くようにリンクを張る。
2. caio.htmlと言うファイルを作成し、正解者に対するメッセージを入れる。また「次の問題」と言う所を作成して、他の人のcaiq.htmlへリンクを張る。
3. caix.htmlと言うファイルを作成し、誤答者に対するメッセージ(ヒントなど)を入れる。また「問題にもどる」と言う所を作成して、caiq.htmlへリンクを張る。
4. Explorer等で文章が表示されるか、リンクが正しく張れたかを確認せよ。

各ファイルでは、<Title>なども忘れずに入れること。

1.4 画像の指定の仕方

WWWが他のサービス(メールやニュースなど)と比べて有利な点は文字情報だけでなく、画像や音声情報が扱える点です。しかしこれらの情報は文字情報に比べると扱いが難しいのでこのテキストでは簡単なイメージの扱い方についてのみふれます。

画像の入っているファイルをここでは画像ファイルと呼んでいますが、画像と言っても様々な種類があります。大きく分けるとモノクロとカラーに分かれます。モノクロも単に白と黒の点からなるものと、灰色の点も許すものがあります。カラーにしても同様に何色まで使えるかによっていろいろで、最も少ないもので8色から多いものでは1,600万色まであります。

色数の多い画像は非常に大きなファイルになります。よってそのままでは多くの記憶容量を必要とするほかに、通信の際に時間がかかるなどの問題が生じます。そこでこれを圧縮する方法がいろいろ考えられてきました。現在では画像を圧縮保存する方式が数十種類ありますが、その中でWWWで利用できるのは数種類です。カラー画像に限定すれば、通常GIF (Graphics Interchange Format)形式とJPEG (Joint Photographics Experts Group Bitmap)方式がよく使われています。数cm角以上の画像ではJPEG方式の方が小さなファイルになります。それ以下の画像ではGIF形式の方が小さくなります。学内で見ているのに関してはあまり問題はありませんが、より多くの人に見てもらう事を考えると、できるだけ容量は少ないのに越した事はなく、画像の大きさにあった方式を選ぶ他に、できるだけ一つのページに大量の画像を利用しないなどの工夫が必要でしょう。

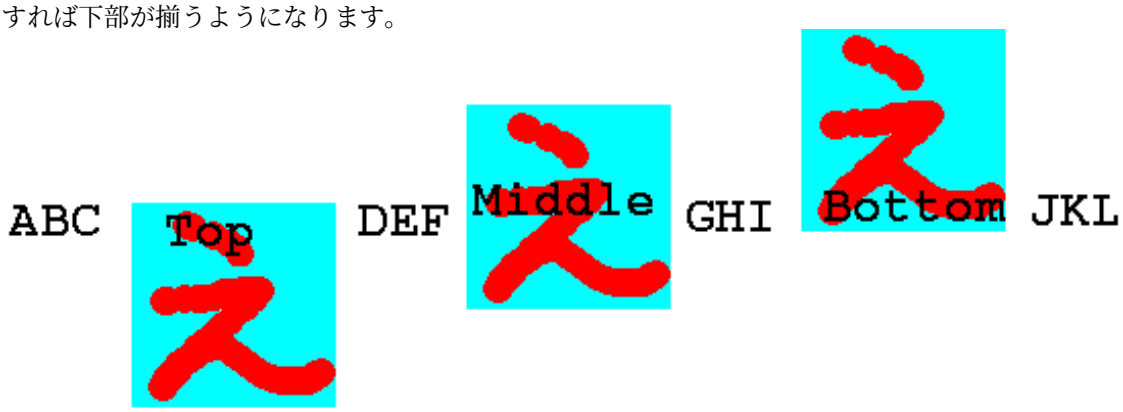
JPEG方式の画像ファイルの拡張子はjpgに、GIF形式の画像ファイルの拡張子はgifにします。そして画像を取り込みたい所に、

```
<Img src="画像ファイル名">
```

を挿入します。一つの画像が一つの文字と同様に扱われます。ところが通常画像は文字よりも大きいので前後の文字と画像のどこを合わせるかでかなり違ったものになります。そこで、

```
<Img src="画像ファイル名" align=Top>
```

とすると前後の文字に画像の上部が並ぶようになります。Top の所を Middle にすれば画像の中央が、Bottom にすれば下部が揃うようになります。



画像を見る事ができない場合や画像が表示されるまでに画像の代わりの文字列を出す事ができます。

```
<Img src="画像ファイル名" alt="画像の表題">
```

のように指定します。できるだけ画像にはこの指定を付けるようにして下さい。

実は画像ファイル名の所には URL を入れることも可能です。これによって他のページで使用されている画像を借りてくることができます。ただこの場合改めてそちらから画像を取り寄せることになるので、一般的に表示されるのに時間がかかるようになります。また大抵のページの作者は画像だけ使われるのは嫌うので注意して下さい。

同じページの中に表示しなくても良い場合には、

```
<A href="画像ファイル名">文</A>
```

も可能です。こうすると「文」を選択すると画像が表示されるようになります。ページ内には小さく縮小したものを入れて、元の大きさを見たい人だけ選択すれば見えるようにすると良いと思います。

また通常のリンクとの組み合わせも可能です。

```
<A href="URL"><Img src="画像ファイル名"></A>
```

とすると表示された画像をクリックすると URL で指定したところへ行きます。

演習問題

1. 次のような手順で適当な画像を参照するページを作ってみよ。
   (a) 学園内のページから適当な画像を含むページを探す。
   (b) 「表示」メニューの中の「ソースの表示」を選択してそのページの画像の URL を求める。
   (c) 現在表示されている Web ページの URL の最後のファイルの部分を消して、代わりに画像の URL を追加したものを入力して、画像のみが表示されるかどうか確認する。
   (d) aaa.html にこの画像が表示されるようにせよ。
   (e) さらに画像をクリックすると画像のあったページが表示されるようにせよ。

1.5 画像の入力と編集

一般的には画像の入力方法には次のようなものがあります。

- 自分でマウス等を利用して入力する。
- 既にある画像をイメージスキャナーで読みとる。
- デジタルカメラで撮影する。
- ビデオ信号からビデオキャプチャー装置で取り込む。

などです。文章の入力でも後での訂正が必ず必要となるように画像の入力だけでは通常済みません。余分な部分を削ったり、全体の色の補正をしたり、拡大・縮小をしたり、ちょっとした画像を作成するのにもかなりの手間がかかります。また文章でも本などからそのまま取ってくると著作権の問題が生じますが、画像の場合も同様な問題があるので注意して下さい。

ここでは機器的な制約より、Windows95 のマシンでマウス等を使って画像を作成する方法について説明します。この場合、できた画像をワークステーションに転送する必要があります。ワークステーションで作成すれば転送する必要はありませんが、本体でしか作業ができませんので基礎演習で大勢でやるには不適当のようです。

1.5.1 画像ファイルの作成

「学生情報処理実習室メニュー」にある「お絵描きツール」のボタンをクリックして、作画のためのプログラムを起動します。Windows95 に標準的についてくるお絵描きソフトである「ペイント」でも画像を作製することも可能ですが、BMP と言う形式のファイルしか作製できないので後で GIF や JPG に変換する必要があります。

この後いろいろ操作をして画像を作成するのですが、次の点に注意して下さい。

1. 最初に「ファイル」メニューの下にある□のアイコンをクリックすると「新しい絵を描く」と言うウインドウが出ますので適当な大きさを指定します。画面上での大きさが実際に表示されたときの大きさとほぼ一致します。絵を描いてからサイズを変更すると色々問題になるので描く前にサイズを設定しましょう。「編集」メニューの中の「編集サイズの変更」を選択するとサイズを変えることができます。

2. 塗りつぶすツールで失敗すると悲惨な事になりますが、そういう場合もあわてずに、すぐ後述の 4 番目のアイコンをクリックするか、メニューの「編集」の中の「やり直し」を選択すればなんとかなります。このやり直しは他のツールでの失敗の時にも使えます。

3. 画像が完成したら「ファイル」メニューの中の「名前を付けて保存」を選択して、ファイル名を指定した上で保存します。ファイル名は後で指定する際に困らないように英数字によるものにします。このとき必ずファイルの種類として「GIF イメージ」を選択します。どこに保存したか忘れないように、また後での操作が容易なように「C:」に保存すると良いでしょう。

4. 保存をする際に何色で保存するかが出ます。少ない色数にした方が小さなファイルになります。ただしパレットの前の方の色だけ保存されるようになるのであらかじめ不要な色を削除しておいた方が良いでしょう。

5. パソコンに画像ファイルを置いたままにしないで下さい。断り無く消去する事があります。必ず次項のファイルの転送を行って、無事転送できたら「ごみ箱」アイコンにドラッグして消して下さい。

以下にこのソフトのアイコンの説明をします。

1. 新しい画像を作成します。
2. 既に保存してある画像を呼び出します。
3. 画像をディスクに保存します。
4. 直前に行った操作を取り消します。
5. クリップボード (一時保管場所) にコピーします。
6. クリップボードに移動します。
7. クリップボードに入っていた内容を取り出します。
8. パレットを表示します。もう一度クリックするとパレットの表示を止めます。
9. ペン先の選択ウインドウを表示します。
10. 部分的に拡大されたウインドウを表示します。
11. レイヤ管理のウインドウを表示します。これは動画を作成するときに使用します。
12. スクラップブックと呼ばれる一時的な保管場所のウインドウを表示します。
13. マウスカーソルの先端付近だけを拡大して見せるちびルーペウインドウを表示します。
14. 複数のレイヤとして作成された画像を順次切り替えて見せることにより動画として表示します。
15. 複数のレイヤからなる画像の中から現在表示するレイヤを選択するときにここをクリックします。

1. 表示されている画像を拡大するときにこれで拡大したい場所をクリックします。右ボタンでクリックすると縮小になります。
2. ウインドウよりも画像の大きさが大きい場合、これでドラッグすることにより表示場所の移動ができます。
3. 画像を部分的に切り取ったり、取り込んだりする際の範囲の設定に使います。切り取ったり取り込まれた画像はクリップボードに入ります。単にコピーをしたい場合はそのままドラッグすれば良いのですが、他の画像などに写す場合はちゃんと指示します。この時に「編集」メニューの中の「切り取り」や「コピー」を使いますが、キーボードを使用して Ctrl + x と Ctrl + c を使用する方が良いでしょう。クリップボードに入った画像を取り出すには「編集」メニューの「貼り付け」を選択しますが、キーボードで Ctrl + v を使用する方が良いでしょう。
4. 左ボタンのドラッグにより細い線を描きます。設定したい色と同じところで右ボタンのクリックをすると、以後その色で描くことができます。
5. 線を描くことができます。描くときのペン先を色々選択することができます。
6. クリックした領域を塗りつぶします。
7. 下のレイヤの画像をコピーすることができます。
8. パレットウインドウの左側で指定された色を右側で指定された色で置き換えることができます。
9. 文字を書く時にこれを選択します。
10. 色をぼかすことができます。
11. 色を暗くすることができます。
12. 色を明るくすることができます。
13. グリッド (補助線) を表示することができます。
14. 画面にマンガのコマ枠のようなものを描くことができます。
15. 画面にマンガのふきだしのようなものを描くことができます。
16. 自由な線を描くときにこれを選択します。
17. 直線を描くときにこれを選択します。Shift キーを押しながら描くと水平線や垂直線が描けます。
18. 長方形をを描くときにこれを選択します。Shift キーを押しながら描くと正方形になります。
19. 円や楕円を描くときにこれを選択します。Shift キーを押しながら描くと円になります。
20. 塗りつぶされた長方形をを描くときにこれを選択します。Shift キーを押しながら描くと正方形になります。
21. 塗りつぶされた円や楕円を描くときにこれを選択します。Shift キーを押しながら描くと円になります。

22. 連続して直線を描くことができます。
23. 左ボタンをクリックしていくと直線が、ドラッグすると自由な曲線を描くことができます。
24. 点線を描くことができます。
25. 最初にクリックした点を起点とした放射線を描くことができます。
26. 最初に描いた線と平行な線を描くことができます。
27. 最初に直線を描き、さらに制御点を指定することにより曲線を描くことができます。
28. 現在のペン先の形が表示されています。▲や▼をクリックすることにより、ペン先の大きさを変更することができます。
29. パレットの右の色、左の色、背景色や透明色が表示されます。

1.5.2 画像ファイルの転送

通常パソコンとワークステーションの間でファイルを転送するにはftp (File Transfer Protocol)を利用します。そのためには次のような手順となります。

- ftpの起動とワークステーションへの接続
  1. 画面左下の隅にある「スタート」をクリックする。出てきたメニューで「プログラム」を選択し、さらに出てきたメニューで「MS–DOSプロンプト」を選択する。
  2. 「cd　¥↩」を入力する。
  3. 「ftp 202.35.224.200↩」を入力する。
  4. User(202.35.224.200:(none)):と出たら自分の「登録名@グループ名↩」を入力する。
  5. Password:と出たら自分のパスワードを入力して↩を押す。
  6. Login failedと出たら失敗なので、「bye↩」を入力して再びftpから入れ直す。
  7. User 登録名 logged inと出たら成功なのでさらに「cd www↩」を入力する。
  8. 「binary↩」を入力する。
- パソコンからワークステーションへ

  ワークステーションに転送したいファイルがgazou.gifならば、「put gazou.gif↩」と入力する。
- ワークステーションからパソコンへ

  パソコンに転送したいファイルがgazou.gifならば、「get gazou.gif↩」と入力する。
- ファイル転送の終了

  「bye↩」を入力する。

ちょっと手順がやっかいですが、これでパソコンとワークステーションでファイルのやり取りができます。

演習問題

前章で作成した超簡易CAIの各ページに適切な画像を追加せよ。

2　会話的なWebページ

前章で説明した方法で作成するページは、全てあらかじめ作られたものを見せるのに過ぎません。利用者側からの入力を受けて、これに答えるようなものの作成法について取り上げます。

2.1　プログラムの作成方法

利用者の入力に応じて異なる応答をするためには、それを実行するプログラムを作成する必要があります。このためにWWWのサーバーには設定したプログラムをサーバー上で起動し、利用者からの入力をそのプログラムに渡し、プログラムからの出力をまた利用者に送り返すと言うような機能があります。これをCGI (Common Gateway Interface)と呼んでいます。

このCGI機能を利用するにはプログラムに様々な条件を満たすことが必要です。

- プログラムの拡張子は.cgiにする。通常のプログラムはコンパイルするとa.outと言う名前になりますが、そのままでは使えません。
- 利用者側からの入力は、パラメタの形か標準入力の形でプログラムに渡されます。標準入力とは通常はキーボードからの入力ですので、テストはキーボードで入力することで可能です。
- プログラムの標準出力はそのまま利用者側に送られます。利用者側はそれが単なるテキストなのか、HTMLの記号が付加されたものなのか、画像データなのかわからないので、最初にその目印を送ります。(後の例で出てくるContent-type: text/htmlとその次の空行)通常標準出力は画面ですのでテストの際は画面に正しいものが表示されれば大丈夫です。
- プログラムの実行はサーバーが行います。そのためにファイル等の読み書きを行うプログラムはファイルの許可に注意する必要があります。

実際の簡単な実行例は次のようになります。

1. testcgi.cと言うファイルに以下のプログラムを入力して保存、終了する。

```
#include <stdio.h>

main(){
   puts("Content-type: text/html");
   puts("");
   puts("<HTML>\n<Head>");
   puts("<Title>Test</Title>");
   puts("</Head>\n<Body>");
   puts("<H1>This is a test for cgi.</H1>");
   puts("</Body>\n</HTML>");
}
```

2. testcgi.cをコンパイルする。
3. プログラムを実行して、画面に次のように出力されれば良い。

```
Content-type: text/html

<HTML>
<Head>
<Title>Test</Title>
</Head>
<Body>
<H1>This is a test for cgi.</H1>
</Body>
</HTML>
```

4. a.out と言う名前を test.cgi と言う名前に変更する。

```
cc07[miwako]% mv a.out test.cgi
```

5. URL として http://www.center.sugiyama-u.ac.jp/グループ名/登録名/test.cgi を指定すると画面に、

```
    This is a test for cgi.
```

と表示されます。

cgiプログラムを作成する際には以下のような点に注意します。

- プログラムを修正した場合には再度コンパイルをし、名前を変え、実行できるように設定を変更しなければなりませんが、名前を変えるところで2回目からは、「mv: ～ を上書きしてもよろしいですか (yes/no)?」と確認を求められるので、y と答えます。
- プログラムが別のファイルを読み書きするならば、そのプログラムの設定を変更する必要があります。今そのプログラム名が test.cgi ならば、a.out をこの名前に変更した後で、

```
cc07[miwako]% chmod u+s test.cgi
```

を必ず行わないとテストでは動いても正しく動作しません。

- プログラムの出力が正しい HTML のものになっているかどうかの確認は、名前の変更前ならば、

```
cc07[miwako]% a.out | jweblint -
```

で行えます。

- プログラムからページの内容を出力するのではなく、既存のページの内容を表示するには、“Location: URL”と言う行と空行のみを出力します。URL の部分はもちろん表示させたいページのものです。

演習問題

1. 呼び出した回数を表示するcount.cgiを作成せよ。

   (例：http://cc01.center.sugiyama-u.ac.jp/~miki/count.cgi)

2. 呼び出すたびに前回作成したCAIの問題文のページ、正解のページ、間違いのページを順番に表示するiroiro.cgiを作成せよ。

3. 呼び出す回数をもとに毎回異なる運勢を表示するおみくじプログラムを作成せよ。なお運勢の内容は/homes01a/ss92a/mmiki/omikujiと言う名前のファイルの中にあるメッセージを利用せよ。メッセージの件数は7つで、1行につき一つのメッセージである。

   (例：http://cc01.center.sugiyama-u.ac.jp/~miki/omikuji.cgi)

2.2 定型的なプログラムの起動法

次に説明するFormを利用すると様々な入力データをプログラムに渡すことが可能ですが、場合によっては、

- 単に指定のプログラムを起動すれば良い程度である。
- プログラムの起動を普通のリンクと同じような形式にしたい。
- 一つのページで様々なプログラムを起動したい。

のような要求が出ることもあります。そのような場合には、既に述べた<A href="URL"> ... </A>でURLで起動したいプログラムを指定する事により解決できます。(ただしプログラム名は前節同様に.cgiで終わっている必要がある。)

さらにプログラム名の後に次のような形でプログラムのパラメタを渡すことが可能です。

```
<A href="プログラムのURL?パラメタ1+パラメタ2+パラメタ3"> ... </A>
```

この場合プログラムに渡されるパラメタ1からパラメタ3は、プログラムのmain の先頭を次のようにしていると、argv[1]からargv[3]までにそれぞれ入ります。

```
main(int argc, char *argv[])
```

argcにはパラメタの数+1が入りますので、パラメタの数が色々変化するような場合にも対応できます。注意しなければならないのは、argv[]に入ったパラメタは文字列として扱われていることです。もし数値として扱いたい場合にはatoi()と言う関数を使用して数値に直す必要があります。

```
n=atoi(argv[1]);
```

これで1つ目のパラメタとして指定した数値がnに入ります。端末でまたテストをする場合はパラメタの部分は空白で区切ります。例えばhref="test.cgi?aaa+2222+ccc"をテストするときには、

```
cc07[miwako]% test.cgi aaa 2222 ccc
```

とします。

なおパラメタの中に空白や漢字等が含まれる場合には後で説明するコード化したものを指定します。

演習問題

1. index.html の中で、「おみくじ」を選択すると前の演習問題で作成した omikuji.cgi が起動されるようにせよ。
2. パラメタとして与えられたファイル名をもとに、そのファイルの内容を表示する list.cgi を作成せよ。
3. 前問で作成したプログラムの出力を見ると<>の部分が消えている。そのような事がないようにせよ。

   ヒント：ファイルの内容を出力する際に、一文字ずつ出すようにして、<や>の時は代わりに<や>を出すようにする。

2.3 プログラムの出力の取り込み

あちこちのホームページにカウンターが付いており、何人目のアクセスであるかどうかが判るようになっています。回数を数えるだけならばすでに CGI を利用して実現してみましたが、実際のページではその内容の全てをプログラムで出力するのは実用的ではありません。ページのごく一部にプログラムの出力やファイル等を取り込む SSI (Server Side Include) と言う機能があるのでそれを利用するのが普通です。本来 SSI 自体は様々なものを取り込む事ができるのですが、ここではそのごく一部のみを紹介します。

- SSI を利用したファイルの拡張子は**.shtml** である必要があります。中身は 99%は通常の.html のファイルと同じなのですが、次に述べる取り込みの指示が有効になります。もし拡張子が.html のものに SSI の指示を入れても無視されますので注意してください。
- コマンド (通常のプログラム) の実行結果を取り込みたいときには、

  ```
  <!--#exec cmd="実行したいコマンド" -->
  ```

  を取り込みたいところに挿入します。自作のプログラムの場合は./プログラム名にしないと動きません。
- CGIプログラムの実行結果を取り込みたいときには、

  ```
  <!--#exec cgi="実行したいCGIプログラム" -->
  ```

  を取り込みたいところに挿入します。自作の CGIプログラムの場合は./CGIプログラム名にしないと動きません。

どうしても拡張子が.html のファイルでも SSI を使いたいときはそのやり方がありますがお勧めできません。と言うのは SSI が使えるファイルはサーバーがクライアントに送る際に必ずその内容を調べて必要があれば挿入を行わなければなりません。その結果サーバーに負担がかかるわけですが、.html でも SSI と言うことになると全てのファイルに対して調べる必要が生じるので大変だからです。

コマンド (普通のプログラム) と CGIプログラムとの違いは最初に Content-type の行を送るかどうかが大きな違いです。ちょっと考えるとその分簡単で良い普通のプログラムだけでも良いように思えますが、Content-type で以下続く内容が画像データであることを示して画像データを出力して表示すると言うことは CGIプログラムでしかできません。

自分のホームページ (index.html) に SSI を使いたい場合は、これを index.shtml に名前を変更すれば可能です。(つまり index.html はなくす) 他の人に URL を教えるときにファイル名を省略したもの (`http://www.center.sugiyama-u.ac.jp/グループ名/登録名/`) で伝えれば、これまで通り今度は index.shtml を見せるようになります。

演習問題

1. date と言うコマンドを使用すると現在の日時が表示される。これを test.shtml の中で呼び出すようにせよ。

2. cal と言うコマンドを使用すると今月のカレンダーが表示される。これを test.shtml の中で呼び出して正しく表示されるようにせよ。

3. test.shtml の中に乱数を使用して、これまで作成した画像の中から一つをランダムに表示するようにせよ。(selgif.c)

2.4 入力用ページの作り方

実際の処理を行うのはプログラムですが、通常はプログラムは利用者から何かデータをもらってからそれを処理します。そのデータをもらうためのページの作るために用いる HTML のコマンドについて説明します。

2.4.1 Form: 形式とプログラムの指定

Form タグを使って、入力されたデータを送る先のプログラムの指定とそのときのデータ形式を指定します。

```
<Form method=POST action="URL">
 .....
</Form>
```

method としては POST 以外に GET と言うのも利用できますが、GET は POST と比べてやや扱いが簡単なものの、入力できるデータ量などに制限があるので、ここでは説明を省略します。URL の部分には実際に作成したプログラムの URL が入ります。また</Form>までの部分に以下で説明するボタンや入力欄の定義が入ります。

2.4.2 Input: ボタン入力

入力が全て終わって、プログラムを起動してくれと言うボタンの定義は次のようにします。

```
<Input type="submit" value="Send">
```

Send の部分はボタンの上に書かれる文字列なので、任意のものが可能です。また似たような形ですが、利用者の入力したものを全て消して最初の状態に戻すボタンは次のように定義します。

```
<Input type="reset" value="Erase">
```

Erase の部分はボタンの上に書かれる文字列なので、任意のものが可能です。
利用者が印を付ける形のボタン (チェックボックス) は次のようにします。

```
<Input type="checkbox" name="info">
```

この場合はボタンしか出てこないのでこの前後に文章を補って、何のボタンかを示す必要があります。info はプログラムに結果を返すときの名前になりますので、同じ Form の中で重複しなければ info 以外の適当なものでも構いません。
幾つかの選択の中で一つだけしか選べないボタン (ラジオボタン) には、次のようにします。

```
<INPUT type="radio" name="service" value="ip" checked>
<INPUT type="radio" name="service" value="uucp">
<INPUT type="radio" name="service" value="ppp">
```

name=で指定しているものが同じもの同士が一つのグループとなり、この中では一つだけしか選択できなくなります。そしてこれはプログラムに返すときの名前になりますので、service 以外の適当なものでも構いません。ip、uucp、ppp はそれぞれのボタンが選択されたときにプログラムに返される値です。これも識別さえできれば適当な値で構いません。なお checked の付いている所が最初に選択された状態になります。

2.4.3 Input: 短い文字列の入力

短い文字列の入力も INPUT で行えます。長い文章などの入力には次の TEXTAREA を用います。

```
<Input name="realname">
```

realname はプログラムに結果を返すときの名前になりますので、適当なものでも構いません。画面上では通常 20 文字程度の幅の入力欄になりますが、入力する内容は、それ以上になっても構いません。しかし入力の際に一部しか見えない状態になるので、場合によっては次のように size=として入力欄の幅を文字数で指定する方が良いでしょう。

```
<Input name="realname" size=100>
```

また、次のようにすると利用者が入力した文字が全て*印で置き換えられて表示されるので (もちろんプログラムには入力した文字がそのまま送られる。) パスワード の入力などに使えます。

```
<Input type="passwd" name="pass">
```

pass の部分は適当なものでも構いません。

2.4.4 Input: 定形データを送る

いくつかのページで同じプログラムを利用する場合、プログラム側からどのページから来たものか判断するための情報が必要となります。また WWW では、サーバーとクライアントは一つのページをやりとりする度に接続を切ってしまうので、プログラムが複数のページを生成するときに何番目までを送ったかを知るためにも使われます。

```
<INPUT type="hidden" name="address" value="miwako">
```

この場合、画面には入力のためのものは何も表示されません。そして後述のように、address=miwako と言うデータがプログラムに渡されます。

2.4.5 Textarea: 長い文字列の入力

長い文章などを入力するには Textarea を使います。

```
<Textarea name="bunsyo" rows=10 cols=50>内容</TEXTAREA>
```

この例では縦 10 行、横 50 文字の入力領域が確保されます。「内容」の部分はこの入力用領域に予め入るものです。不要ならば省略しても構いません。

2.4.6 Select: 選択メニュー

項目を示してその中から一つを選んでもらう方式です。選択をしようとすると項目の一覧が表示されるので、多くの項目でも場所をとりません。

```
<Select name="selone">
   <Option>寝る
   <Option selected>食べる
   <Option value="run">走る
</Select>
```

この例では、「寝る」、「食べる」、「走る」の3つの中から一つ選ぶ事ができます。<Option>の所はいくつでも可能です。selectedは1つだけ指定可能で、この項目には最初から印が付きます。またvalueの指定をすると、プログラムに渡される値はこちらになります。(この例では「走る」の代わりに「run」が渡される。)

なお、複数選択可能な項目とするには、最初の行を次のようにします。

```
<SElect name="selmul" multiple>
```

この場合にはselectedを複数指定しても構いません。複数選択したときには名前=値と言うものが複数繰り返されますので注意が必要です。

また、画面上で幾つかの項目が最初から表示されるようにしたい場合には、最初の行を次のようにします。

```
<Select name="selone" size="3">
```

とすれば、最初から項目が3つ表示されるようになります。項目が3 つ以上ある場合には、残りの項目を表示させるためのスクロールバーが自動的に表示されます。

2.4.7 入力されたデータの表現方法

以上の入力欄に入力されたデータは、Formの所で指定されたプログラムに次のような形で渡されます。

名前=値&名前=値&...&名前=値

名前は今まで説明したタグの中でname=で指定した内容です。そして値が実際に入力された内容になります。名前と値の間には=が、次の名前との間には& が入ります。このデータはプログラムへの標準入力として渡されますが、いくら多くのデータであっても1行です。

また値の中に含まれる記号類は全て%の後に16進数のコードとして表現されます。また空白は+になっています。これらの例を以下に示しますが、漢字などはクライアント側で使用している漢字コードの種類によって変わってきますので対応はかなり難しくなります。

今次のようなファイル(例えばinput.html)を作成して実行すると、

```
<HTML><Head><Title>Test form</Title></Head>
<Body>
<Form method=POST action="show.cgi">
これはTextAreaです。<Br>
<Textarea name="name-1" rows=3 cols=60>Sample Data</Textarea>
<P>これはInputです。
<Input name="name-2">
<p><Input type="submit" value="Send">
   <Input type="reset" value="Erase">
</Form>
</Body></HTML>
```

次の様な画面になります。

これを実行する前にshow.cgiを作成しておく必要があります。例えば、cgiプログラムに渡されたものをそのまま返すプログラムならば次の様になります。

```
#include <stdio.h>

main(){
   char line[300];                /*  十分な大きさの変数にすること      */

   gets(line);                    /*  FORMからの入力                     */
   puts("Content-type: text/html");
   puts("");
   puts("<HTML><Head><Title>Input data</Title></Head>");
   puts("<Body><Pre>");
   puts(line);                    /*  FORMからの入力をそのまま出力する */
   puts("</Pre></Body></HTML>");
}
```

これをコンパイルして前述のinput.htmlを実行して適当なデータを入力してSendボタンを実行すると、

```
   name-1=abc%0def%0ghi&name-2=jkl
```

のような表示がされます。

演習問題

1. 和文英訳の問題のページを作成せよ。入力された英文が正解と一致するかどうかで判定を下す。(eigo.html、eigo.c)

   次の和文を英訳せよ。入力を終えたならO.K.を選択すること。

   「私は少年です。」

   解答 ----------
   O.K.

   (http://cc01.center.sugiyama-u.ac.jp/~miki/eigo.html)

2. 性格判断のページを作成せよ。(seikaku.html、seikaku.c)
   - 答えのパターン数は50以上のこと。
   - 答えの判定にはstrstr()関数を利用するとよい。

   (http://cc01.center.sugiyama-u.ac.jp/~miki/seikaku.html)

3 HTML 中級編

ここではより高度？なHTMLのタグについて説明します。できるだけブラウザ(WWW表示ソフト）に固有の機能は避けています。

3.1 画面の背景の設定

画面の背景に色を付けたり、画像を敷き詰めたりすることが可能です。ただ後者の場合あまり大きな画像を用いると表示の際に時間がかかるので注意しましょう。

まず背景に色を付ける場合は、

```
<Body BGColor="色の指定">
```

のようにします。色の指定の仕方には次のようなものがあります。

- RGB値を#FF0000のように指定する方法があります。光の三原色である赤(R)、緑(G)、青(B)の配分を00〜FFまでの16進数でこの順番に記述します。一般に、値が大きいと明るい色、小さいと暗い色、RGBの値の差が大きいと派手な色、差が小さいと淡い色になります。
- 次の16色については名前で指定可能です。 Black、Gray、Silver、White、Red、Yellow、Lime(黄緑)、Aqua(水色)、Blue、Fuchsia(薄紫)、Maroon(えび茶)、Olive、Green、Teal、Navy、Purple

画面の背景の色を変更した場合、通常の文字の色では見にくくなることがあります。その場合には次のように文字やリンク場所の色を指定することも可能です。

```
<Body BGColor="black" Text="white" Link="red" VLink="yellow">
```

このようにすると画面の背景は黒、通常のテキストは白、リンクの部分は赤、そして一度選択されたことがあるリンクは黄色になります。

また画面の背景に画像を敷き詰める場合には次のような指定をします。

```
<Body Background="画像のURL">
```

指定された画像が画面より小さい場合は繰り返し敷き詰められる形になります。画像の内容に合わせて文字の色の設定も前述同様に行うことが可能です。ただし画像の内容がほとんど黒で文字の色を白にしたような場合、この背景用の画像が正しく転送されなかった場合には、まったく読めない画面になってしまう恐れがあるので、ほぼ同等の色をBGColorで同時に指定しておくと良いようです。

演習問題

1. CAIの問題文のページ(cai1.html)に対して、背景の色を設定せよ。またLinkとVlinkに同じ色を設定することにより正解がばれないようにせよ。
2. 性格判断のページ(seikaku.html)の背景に画像を敷き詰めよ。

3.2 文字の修飾

ワープロなどと同様に文字の大きさを変更したり、色を付けたりすることができます。しかしあまりこのような修飾を乱用するとかえって見にくいものになってしまいます。本当に必要な所に適切なものを使ってください。

- 文字の大きさの設定：n の所には 1 から 7 の数字が入ります。数字が大きいほど大きな文字になります。

  ```
  <Font size=n>文文文</Font>
  ```

- 文字の色の設定：「色」の所には先ほどの背景の色と同じ形式の指定 (#で始まる 16 進数によるものか色の名前にもの) が可能です。また上記の大きさの指定である size=も同時に指定することもできます。

  ```
  <Font color="色">文文文</Font>
  ```

- 肩付き文字の指定：$y = x^2$の 2 のように文字の高さの半分上に字を出したいときに使います。

  ```
  <Sup>字字字</Sup>
  ```

- 下付き文字の指定：H_2Oの 2 のように文字の高さの半分下に字を出したいときに使います。

  ```
  <Sub>字字字</Sub>
  ```

- 中央寄せの指定：文字を画面の中央に表示させたい場合に使います。

  ```
  <Center>文文文</Center>
  ```

演習問題

aaa.html に、画面の真ん中に赤い字で大きく「H_2Oは水のことです」と出るように追加してみよ。

3.3 表の指定

次の画面分割以上にタグがごちゃごちゃと大量に出てきますので混乱しないようにしてください。

1. 表の指定全体を<Table>と</Table>タグで囲みます。なお後に出てくる表の中身として表を用いることも可能ですが、その場合は<Table>が入れ子になることになります。

   表の罫線が必要な場合は、<Table Border>とします。Boder を省略すると枠線がない表になりますが、単に整列させたい場合によく用いられます。さらに「Boder=数値」とすると枠線の太さを変えることができます。大きな数値にすると太い枠線になります。

2. 表に表題を付ける場合は、<Table>タグのすぐ後で次のような指定をします。

   ```
   <Caption>表の表題</Caption>
   ```

   ここで指定した表題は表の上に中央寄せされて表示されます。表の下に出したい場合には、<Caption align=Bottom>とします。

3. 各行の内容はそれぞれ、<Tr>と</Tr>で囲う必要があります。

4. 行に含まれる各項目はそれぞれ、<Td>と</Td>で囲う必要があります。

   - 項目の内容が複数行にわたる場合は、改行すべき所に
を入れます。
   - 横隣の項目と合体した横長い項目を作成したい場合には、<Td ColSpan=2>のようにします。(3にすれば3つ連結した形になります。)
   - 下の項目と合体した縦長の項目を作成したい場合には、<Td RowSpan=2> のようにします。(3にすれば3つ連結した形になります。)

以下に簡単な表の例を示します。

```
<Table Border>
  <Caption>表のサンプル</Caption>
     <Tr>
       <Td>aaaaa</Td><Td>bbbbb</Td><Td>ccccc</Td>
     </Tr>
     <Tr>
       <Td>ddddd</Td><Td>eeeee</Td><Td>fffff</Td>
     </Tr>
     <Tr>
       <Td>ggggg</Td><Td>hhhhh</Td><Td>iiiii</Td>
     </Tr>
</Table>
```

表のサンプル

aaaaa	bbbbb	ccccc
ddddd	eeeee	fffff
ggggg	hhhhh	iiiii

演習問題

aaa.html に、次のような形の表が出るようにせよ。

	男とは		
女にとって	愛人	恋人	友人
	父親	夫	息子
	オス	ひも	粗大ごみ

3.4 画面の分割

画面を分割することが可能です。これによって画面を目次とその内容のように分けて、いちいち利用者が目次のあるページの戻らなくても良いようにできます。ただこれを利用すると当然画面が分割されるので一つあたりの面積は少なくなりますので見にくくなります。また lynx のようなこの機能に対応していないもので見る利用者にとっては不便なものになります。

画面を分割する場合は、分割の仕方を指定するファイルとその分割された内容のファイルの両方が必要になります。後者は2分割ならば2つ、3分割ならば3つ必要になります。ただ基本的には後者のファイルはこれまで説明してきた普通の HTML で記述された内容のものです。

1. 画面分割の指定のファイルは<HTML>で始まり</HTML>で終わるところと、その次に<Head>の部分がある所はこれまでのものと同じです。

2. 通常ならば<Body>と</Body>が来るところに、<Frameset> と</Frameset>が来ます。

3. <Frameset>のタグの間には、次に3種類のものが入ります。

   - <Frameset>を入れるとさらに画面を細かく分割することができます。分割の方向としては上下と左右が可能です。画面を上下に分割する場合は、後の例のように Frameset の中で rows=を指定します。「20%,*」とすれば画面が上下に 20%と残りに分けられます。「33%,33%,*」とすればほぼ同じ高さに上中下と分割されます。左右に分割したい場合は cols=を指定します。

- <Frame>を用いて分割された画面に表示する内容が入ったものを指定することができます。

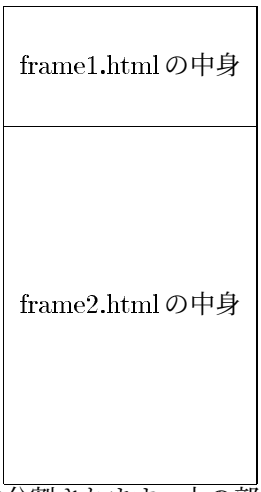
  ```
  <Frame src="URL" name="フレーム名">
  ```

  URLの部分に表示する内容のURLを、「フレーム名」の所には分割された画面を識別するための適当な名前を入れます。後の例では普通のファイルですが、画像ファイルでも、CGIプログラムでも構いません。

- <NoFrame>から</NoFrame>の間に、このような画面分割に対応していないブラウザの利用者に対するメッセージを記述するのに使用します。よくあるのが、「このページはNetscape Navigetor ver.2以降かInternet Explorer ver.3以降でご覧ください。」と言う感じのものです。内容の前後は<Body>と</Body>タグで囲う必要があります。

以下に実際にFramesetを使用した例を示します。

```
<HTML>
<Head>
  表題など
</Head>
<Frameset>
<Frameset rows=20%,*>
  <Frame src="frame1.html" name="FRAME1">
  <Frame src="frame2.html" name="FRAME2">
</Frameset>
<NoFrames>
<Body>
  フレーム未対応ブラウザ用メッセージ
</Body>
</NoFrames>
</Frameset>
</HTML>
```



これを画面分割に対応したブラウザで見ると、画面が上下に分割されます。上の部分が20%、残りが下の部分となり、それぞれframe1.htmlの内容とframe2.htmlの内容が表示されます。

さて一旦このように画面が分割されてしまうと以後それぞれ分割された画面の中で変化するようになります。つまり別のファイルにリンクを張った場合にそれを選択すると、今選択を行った画面の中にそのリンク先が表示されます。大抵はこれで構わないのですが、左の画面で目次を示して、選択された内容は右の画面に出ると言うような場合は困ります。

3.5 表示先の指定

通常のAタグによるリンク先の指定では、選択された場合、このタグが表示されていた画面にそのまま表示されます。ここでAタグにtarget=と言う指定を追加することにより別の場所に表示することが可能になります。

target=”フレーム名”	指定されたフレームに表示
target=”_blank”	新しい画面を作成してそこに表示
target=”_top”	画面分割を全て解除して表示
target=”_parent”	画面分割を一つ解除して表示

演習問題

1. 左右に分割した目次付きのページを作成せよ。左側(目次)はmenu.html、右側はfirst.htmlを20:80で表示するようにせよ。(index.html)

2. 目次の中身を作成せよ。これまで作成したファイル名やプログラム名を並べて、それを選択すると左側の内容がそれに変わるようにせよ。(menu.html)

3. 最初のページの中身を作成せよ。(first.html)

4. さらに「次の人」を選択すると画面の分割を止めて次の人の index.html を表示するようにせよ。(menu.html に追加)

3.6 動く画像

GIF 形式の画像ファイルは、一つのファイルの中に複数の画像を含み、それらを決められた間隔で切り替えて表示するように指示できます。この機能を利用すると表示されたときにぱらぱらと動く画像を作ることができます。

既に紹介した「D-pixed」では複数の画像を別のレイヤとして扱います。各レイヤごとに少しずつ異なる画像を用意すれば動画ができます。以下の手順に従って作成してみて下さい。

1. 「お絵かきツール」を起動して適当な大きさの新規画面を用意します。

2. 最初の絵を描きます。

3. 「表示」メニューの中に「編集中のレイヤのみを表示」というものがあるので印がついていなければクリックして印を付けます。

4. 「ウィンドウ」メニューの中の「レイヤ管理」をクリックすると「レイヤ管理」というウインドウが新たに開かれます。その中の「新規レイヤ」ボタンをクリックすると新しいレイヤが作られます。

5. 次の絵を描きます。そしてまた「新規レイヤ」ボタンを押して次のレイヤを作ることを繰り返します。

6. 前に作成したレイヤを見るときには「レイヤ管理」の中の「レイヤの選択」の中の見たいレイヤをクリックすれば可能です。

7. 「ウインドウ」メニューの中の「ぱたぱたアニメ」をクリックすると今入力した画像が順番に表示されます。より速い動きを実現したいときには、「レイヤ管理」ウィンドウの中の表示時間の所をより小さな値にします。

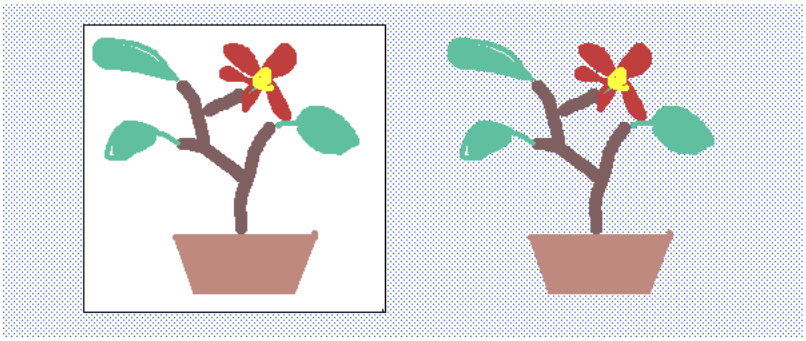
8. ファイルとして保存する際には必ず GIF 形式を選択し、動画 GIF の部分に印を付けます。

演習問題

最低 5 レイヤからなる動く画像を作成し、first.html に入れてみよ。

3.7 透明な背景色

GIF 形式の画像ファイルでは背景色 (通常は白色) と言うものが一つあり、それを透明にすることも可能です。透明にしない場合 (左側) は、背景色が有効になり、描いた絵の回りは背景色になり、もしこれが貼り付けられたページのバックが違う色であると画像全体が浮き上がった感じになります。一方透明した場合 (右側) には、描いた絵のみがページに貼り付けられた感じになります。



背景色を透明にするには次のような操作が必要になります。

- 画像の形式はGIFにします。
- 透明になる色は、パレットのウインドウの中の色の中で点線で囲まれたものです。
- 保存の際に「透明化GIF」のところに印を付けます。
- 動く画像の場合は、レイヤごとに「透明色が有効」に印を付け、さらに「描画方法」を「書き直す」にする必要があります。

演習問題

1. 前節で作成した動く画像の背景を透明にせよ。
2. 次章の課題で作成する「みきっち」に必要な動く画像を作成せよ。できた画像はichiran.htmlで見ることができるようにする。(参考:`http://cc01.center.sugiyama-u.ac.jp/~miki/Mikichi/ichiran.html`)

通常体型の小	normal1.gif	通常体型の中	normal2.gif	通常体型の大	normal3.gif
やせ型の小	yase1.gif	やせ型の中	yase2.gif	やせ型の大	yase3.gif
でぶ型の小	debu1.gif	でぶ型の中	debu2.gif	でぶ型の大	debu3.gif
うんち	unchi.gif	お墓	ohaka.gif		

4 CGI中級編

ここではより高度？な会話的なページを作成する方法について説明します。

4.1 ページ内容の自動更新

通常のWebページは一旦表示されると利用者が何かしない限り動くことができません。例えば街の風景などの画像を定期的に自動更新していても、そのままでは利用者の方に表示された画面は変化しません。この問題を解決するには3 つほど方法があります。

- Javaを利用する。
- サーバーから継続的にデータを送る。
- ブラウザから自動的に更新の要求が出るようにする。

ですが、ここでさらにJavaまで勉強する元気はないでしょうし、サーバーから継続的にデータを送るのも少しややこしいので省略して、最後の例だけ示します。ブラウザから5秒毎に更新の要求を出させるには、Headタグと/Headタグの間に次のような指示を入れます。

```
<META HTTP-EQUIV=Refresh CONTENT=5>
```

またCGIの出力ではContent-typeの次に出力する方法もあります。

```
Content-type: text/html
Refresh: 5

<HTML>
......
</HTML>
```

Refresh:の次の行は空行でなければなりません。最後の5が秒数ですがあまり短い間隔で更新をすると、更新に必要な時間を越えてしまうこともあるので注意します。特に学外から見てもらう場合には、そのページの表示に思いがけないほどの時間がかかることもありえますから。

なお、これの応用として、次のようにすると5秒後に別のページを見せることも可能です。

```
<META HTTP-EQUIV=Refresh CONTENT="5;URL=別のページのURL">
```

別のWebサーバーに引っ越したときに、元のページにこの指定をして、元のページを見に来た人を自動的に新しい引っ越し先に飛ばすのに使われます。

演習問題

1. test.shtmlファイルに10秒毎に更新を行う指示を追加して確認せよ。
2. test.shtmlとaaa.htmlに指示を追加し、ほっておくと、それぞれ10秒毎に入れ替わるようにせよ。

4.2 Cookieによる利用者の識別

WWWのサーバーと利用者の間は毎回接続されては切れると言う特徴があります。そのために同じ利用者が再び接続してきた場合に何か特別な待遇をしようとすると、それなりの工夫が必要となります。

その一つとしてtype="hidden"のinputタグを用いる方法があります。利用者に表示するページに必ずこのタグを使用して何らかの識別内容を設定しておき、再び利用者からアクセスがあったときには、設定しておいた内容で判断します。ただこのやり方では、直接利用者には設定した内容は見えませんが、ちょっと操作すれば見えてしまいます。よって偽造などがされやすいのと、inputタグを利用するので必ずFormを使用しなければならない等の問題があります。

ここではCookieと呼ばれるこれとは別のやり方を紹介します。これを利用すると、

- 利用者に設定した内容を見られない。
- 指定した期限まで設定した内容がブラウザ(パソコン)に残る。
- 設定したディレクトリーの中のファイルに対しての応答の場合には、常に設定した内容が返ってくる。

などの特徴があります。なおCookieの具体例とそのプログラムは、

http://cc01.center.sugiyama-u.ac.jp/~miki/Cookie/

を参照してください。

4.2.1 Cookieの設定

Cookieは通常のHTMLのファイルやCGIのプログラムで設定することができます。設定されたCookieは期限の指定があればパソコンのディスクに、期限が無ければパソコンのメモリーに記憶されます。後者の場合、利用者がブラウザのプログラムを終了すると消えてしまいます。Cookieの設定には次のような2つのやり方があります。

- Headと/Headタグの間に、

  ```
  <META HTTP-EQUIV="Set-Cookie" CONTENT="設定内容">
  ```

  を入れる。これはHTMLのファイルでもCGIプログラムでも可能です。

- 「Content-type: text/html」や「Location: URL」を出力する前に、

  ```
  Set-Cookie: 設定内容
  ```

  と言う形で出力する。これはCGIプログラムでしか利用できませんが、Cookie で設定した内容が利用者から見えないので偽造の恐れが少なくなります。

どちらのやり方にしても「設定内容」は次のようなものになります。

```
変数名=値; expires=値; domain=値; path=値; secure
```

「変数名=値」以外は省略可能です。それぞれ、次のような意味があります。

- 変数名=値

  好きな変数名に任意の値を指定します。セミコロン(;)、カンマ(,)、空白や日本語を使用する際にはそれぞれ、%3B、%2C、%20のようにコード化する必要があります。

- expires=値

  クライアント側のディスクに記録されるCookieの有効期限を指定します。値は次のような形式で指定します。

  ```
  Fri, 31-Dec-1999 23:59:59 GMT
  ```

  この項目を省略した場合、Cookieはクライアント側のメモリーにのみ残ります。また過去の値を指定すると直ちにCookieを削除します。

- path=値

  Cookieを発行するページを指定します。このパスを含むページを開いた時に、ブラウザからサーバーへCookieが送信されます。通常、「http://WWWサーバー/ディレクトリ/ファイル」の「/ディレクトリ」の部分になります。省略した場合は今開いたファイルと同じディレクトリーにあるファイルにアクセスした場合にCookieが送られます。

- secure

  これを記述しておくと、Cookie情報が暗号化されて送信されるようになります。

4.2.2 Cookieの読み取り

Cookieの読み取りはCGIプログラムでなければできません。ブラウザが設定したCookieの値は環境変数と言うところに設定されるので、CGIプログラムではこれをgetenvと言う関数で読み取ります。そのためには以下のような手順をふみます。

1. プログラムの先頭にgetenvを使うために、

   ```
   #include <stdlib.h>
   ```

   を追加します。

2. 環境変数の値は文字列になります。これを参照するための変数として、次のような文字ポインター変数を宣言します。

   ```
   char *p;
   ```

3. getenv()を使用して「HTTP_COOKIE」と言う環境変数の値を取り込みます。

   ```
   p=getenv("HTTP_COOKIE");
   ```

4. Cookieがなかった場合には、pの値がNULLになります。そうでない場合は、例えば次のようにしてCookieの内容を表示することができます。

   ```
   printf("Cookie=%s\n",p);
   ```

演習問題

アクセスするたびに何回目かを指摘するCGIプログラムを作成せよ。ただしこの回数はその場限り、利用者ごとのものとする。(countc.c)

ヒント：まずCookieがあるかどうか確認し、なければ「初めてですね」と挨拶して内容が2のCookieを設定する。もしCookieがあり、その内容がnならば「n回目ですね」と表示した上で内容がn+1のCookieを設定する。文字列を数値に直すにはatoi()と言う関数が利用できる。

```
p="n=123"のとき、
   atoi(p)   → ゼロ (n=123は数字でないので)
   atoi(p+2) → 123  (123なので)
   atoi(p+3) → 23   (23なので)
```

4.3 応用課題「みきっち」

数年前に「たまごっち」と言う携帯ゲーム？が流行しました。誕生から死ぬまで、えさをやったりウンチの片づけをしたり、色々世話の焼き方によって様々な成長をすると言うものです。年中ぴーぴー注文を付けてくるので、子供が学校に持ち込んだりして社会問題になったりもしました。これに似たものをWebで実現することを考えます。

基本的な筋書きは次のようになります。

- みきっちの状態としては、
  - 小型、中型、大型、死んでいる
  - やせ、正常、でぶ
  - ウンチがある、ない

  は見た目でもわかりますが、ウンチが出るかどうかはえさを過去にやったかどうかによって決るので、このような外から見える状態だけではわかりません。よって内部では次のような変数を用いてみきっちの状態を表します。

変数名	値の範囲	変数の内容
day	0,1,2...	これまでの経過時間。
status	1,2,3	それぞれ「やせ」、「正常」、「でぶ」であることを示す。
level	1,2,3,4	それぞれ「小型」、「中型」、「大型」、「死亡」であることを示す。
food	0,1,2...	体内にあるエサの数。食べ過ぎると死ぬ。
weight	0,1,2...	体重。これによってstatusが変る。軽すぎも重過ぎも死ぬ。
age	0,1,2...	年齢。これによってlevelが変る。ある値まで来たら死ぬ。
unchi	0,1,2...	未処理のウンチの数。ある値を越えたら死ぬ。

- 「誕生させる」で開始する。開始時の状態は、小型、やせ、ウンチなし、age、day、foodはゼロ、体重は10とする。
- 開始以後できる操作とその結果状態変数に及ぼす影響は次の通り。

操作	操作の及ぼす影響
放置する、次の日	day++、weight−−、やせならば1/4、正常ならば1/6、でぶならば1/2の確率でage++、foodが正ならば1/10の確率でunchi++してfood−−とする。
えさをやる	food++、weightを10増やす。
ウンチを片づける	unchiが正ならばunchi−−。

各操作に共通なものとして、

- いつ死ぬか：「weightがゼロ未満」、「weightが100以上」、「ageが100以上」、「unchiが5個以上」、「foodが10個以上」
- 体型の変化：weightが10以下で「正常→やせ」、weightが30以上で「やせ→正常」、weightが50以下で「でぶ→正常」、weightが70以上で「正常→でぶ」
- 大きさの変化：ageが20以下で「小型」、ageが60以下で「中型」、それ以降は「大型」

とする。以下はこの「みきっち」を少しずつ作成していくことにする。

1. 全体のフレームの定義のファイル(mikichi.html)を作成する。画面を左右に30%と残りに分割する。左側にはメニューのファイル(mmenu.html)、右側にはaction.cgiが表示されるようにする。せっかくだからmenu.htmlにこのファイルへのリンクを追加する。

2. 左側のメニューのファイル(mmenu.html)にはとりあえず、「次の日」、「一覧表示」、「次の人へ」、「トップページに戻る」の4つのリンクを作る。それぞれ、自分のaction.cgiを右側に表示する、自分のichiran.htmlを右側に表示する、次の人のmikichi.htmlを画面全体に、自分のindex.htmlを画面全体に表示するようにする。

3. 最初の段階として「みきっち」を表示するだけのものを作成する。(action1.c)

   ファイル名は順番に変更していくが、コンパイルした後で名前を変更する際は常にaction.cgiにすること。

```
#include <stdio.h>

int day,status,level,food,weight,age,unchi;

void display(){
    今の変数にあった画像を含むページを表示する。
    変数の内容もついでに表示する。
}

main(){
   day=12;status=2;level=1;food=3;weight=15;age=22;unchi=4;
   display();
}
```

4. Cookieがなかった場合の関数を追加する。(action2.c)

```
void setCookie(){
   Cookie を設定する (複数の変数を設定するには Set-Cookie の行を複数出力する)
}

void noCookie(){
   変数に初期値を設定をする
}

main(){
   noCookie();setCookie();display();
}
```

5. Cookie があった場合に変数の読み込みを行う関数を作成する。(action3.c)

   複数の変数を設定して Cookie を読み込むと次のような形になる。

```
   "x=1; y=2; z=3;"
```

   ここで sscanf と言う関数を使用すると一発で変数 a、b、c に読み込める。

```
   p=getenv("HTTP_COOKIE");
   sscanf(p,"x=%d; y=%d; z=%d;",&a,&b,&c);
```

   これを利用して以下を実現する。更新をする度にウンチが増えれば良い。

```
void getdata(){
   Cookie がなければ noCookie() を呼ぶ。
   そうでなけれ Cookie から変数に値を読み込む。
}

main(){
   getdata();
   unchi++;
   setCookie();
   display();
}
```

6. 「みきっち」に対して行える操作として、「次の日」の他に「エサをやる」、「ウンチを捨てる」、「誕生させる」が考えられる。これに対してそれぞれ別のプログラムを作成する方法も考えられるが、内容の違いは変数の操作の部分に限られるので同じプログラムで対応することとする。同じプログラムで異なる動作をさせるために一番簡単な方法として、パラメタを使用する。

動作	パラメタ
次の日	パラメタ無し
エサをやる	1
ウンチを捨てる	2
誕生させる	3

   よって呼び出す側は、

```
<A href="action.cgi?3">誕生させる</A>
```

となり、呼び出される側は、

```
main(int argc, char *argv[]){
   ...
   if (argv[1][0]=='3') { 誕生させる処理 }
   ...
```

のような形となる。以上をもとに、mmenu.html を修正し、プログラムは「次の日」以外に対応できるようにせよ。(action4.c)

7. パラメタがなかった場合や「次の日」の処理 (nextDay()) をプログラムに追加せよ。ただし死亡状態であればこの関数を main() から呼び出さないこと。確率的な動作が必要な部分は、getpid() で得られた値をもとに実現せよ。(action5.c)

8. 「次の日」、「えさ」、「ウンチ」の操作の後に共通して行う部分を main() に追加せよ。また死亡状態でない場合には、20 秒毎に自動更新がかかるようにせよ。(action6.c)

9. 死亡したときに day の値と日付を record と言うファイルに追加するようにせよ。また mmenu.html に「飼育記録」と言う項目を追加してクリックすると右側に record.cgi が表示されるようにし、record.cgi では record と言うファイルの中身を table タグを使用して表示するものとする。(action7.c、record.c)

   ヒント：日付を調べるには time() と言う関数を用いる。しかしこの関数はグリニッジ標準時の 1970 年 1 月 1 日午前 0 時からの秒数を返すだけなので、通常この結果を我々の扱いやすい形式に直す関数を利用する。その中でも一番簡単なのは ctime() だが、使い方にちょっと難しい点があるので、次のサンプルプログラムを参考にせよ。

```
#include <stdio.h>
#include <time.h>              /* time() や ctime() を使用する際に必要  */

main(){
   time_t t;                   /* time_t は秒数を入れるための変数の型 */

   t=time(NULL);
   printf("%s",ctime(&t));     /* ctime() は\n 付きの文字列を返す。     */
}
```

これをコンパイルして実行すると次のような感じの出力が得られます。

```
Wed Dec 15 11:53:37 1999
```

4.4 応用課題「簡易チャット ルーム」

Webを利用したチャットと言うのが結構流行しているようです。要するに複数の利用者が書き込みをする伝言板のようなものです。最近過激な書き込みをして大学にクレームが付くと言う事件がありました。別に名乗っていなくても、どこのパソコンから書き込んだかはすぐわかりますので、調子に乗らないようにしてください。

この応用課題では簡易版と言うことで通常のものとちょっと違う形になっています。
(例が`http://cc01.center.sugiyama-u.ac.jp/~miki/chat.html` にあるので必要があれば参照せよ。)

- chat.html: 利用者受け付けページ。ここで利用者名を入力してもらう。「Start」をクリックしたらchat.cgiにstartと言うパラメタを付けて呼び出す。
- chat.cgi: パラメタの有無で動作が変ります。
  - パラメタ＝startの場合：利用者名をCookieとして設定してからチャットの画面を表示する。
  - パラメタ＝endの場合：Cookieを消去してからchat.htmlを呼び出す。
  - パラメタ＝inputの場合：入力された文をファイルに追加してからチャットの画面を表示する。
  - パラメタ無しの場合：チャットの画面を表示する。

  チャットの画面では、ページの上部にこれまでのやり取りを順番に並べ、一番下にメッセージの入力欄と「書き込み」、「更新」、「終了」のボタンやリンクがあるものとする。

以下の手順で作成してみよ。

1. chat.htmlを作成せよ。名前の入力欄にはnameと言う名前を付けること。またmenu.htmlに「簡易チャット」と言う項目を追加して、画面の右側に呼び出せるようにせよ。
2. 最後の入力欄と「書き込み」、「更新」、「終了」のボタンやリンクを表示するだけのchat.cgiを作成せよ。ただしこの表示の部分はdisplay()と言う関数にすること。(chat.c以下同様)
3. パラメタが無い場合は単にdisplay()を呼ぶようにせよ。
4. パラメタがstartの時に、その内容をCookieとして設定する関数setCookie()を呼び、次にdisplay()を呼ぶようにせよ。
5. 「終了」をクリックするとCookieを消去した上でchat.htmlに戻るようにせよ。
6. パラメタがinputの時に、入力された内容を「/var/tmp/グループ名.登録名.chat」と言うファイルに追加する関数input()を呼び、次にdisplay()を呼ぶようにせよ。
7. 前述のファイルの内容を表示する部分をdisplay()に追加せよ。
8. このままでは更新や入力時にページの表示が先頭までもどってしまうので次のようなJavaScriptの記述を`</Body>`タグの直前に出るようにせよ。

```
<Script language="JavaScript">
<!--
   document.fo.message.focus();
//-->
```

   foと言うのはFormに付けた名前である。Formのタグの中で`name="fo"`のような指定を追加する。messageと言うのは入力欄に付けた名前である。