

生活社会科学科  
基礎演習 テキスト

三木 邦弘

平成14年9月25日

目次			
1 はじめに	2	3.5 表示先の指定	19
1.1 演習の受け方	2	3.6 動く画像	19
1.2 HTMLについて	2	3.7 透明な背景色	20
1.3 JavaScriptについて	2	4 JavaScript 入門	21
2 Webページの作成	3	4.1 どこに入れるのか	21
2.1 作成の手順	3	4.2 どのように入れるのか	21
2.1.1 準備作業	3	4.3 画面への出力	21
2.1.2 ファイルの作成	4	4.4 計算式	22
2.1.3 ワークステーションとファイ ルのやりとり	4	4.5 フォーム	22
2.1.4 ファイル名とURL	5	4.6 入力欄への入出力	23
2.2 HTML入門	5	4.7 関数の定義(1)	24
2.2.1 簡単な例	5	4.8 ボタン	24
2.2.2 基本タグ	6	4.9 条件判断	25
2.2.3 文字の形式について	8	4.10 警告・確認画面	26
2.2.4 リンクの付け方	9	4.11 変数	27
2.2.5 画像の指定の仕方	10	4.12 関数の定義(2)	28
2.3 画像の入力と編集	11	4.13 繰り返し	29
3 HTML中級編	15	4.14 選択メニュー	30
3.1 画面の背景の設定	15	4.15 タイマー割り込み	32
3.2 文字の修飾	16	4.16 Windowの操作	34
3.3 表の指定	16	5 Style Sheet 入門	36
3.4 画面の分割	18	5.1 どこに入れるのか	36
		5.2 基本的な形式	37
		5.3 クラス分け	38

# 1 はじめに

平成14年度の私の基礎演習では、現在流行のインターネットで最も良く使われている WWW(World Wide Web)において、利用する側からでなく情報を提供する側になるための技術的なものを取り上げます。具体的にはHTML (Hyper Text Markup Language)とプログラミングの初歩の勉強を兼ねてJavaScriptを取り上げます。来年度開講される私の専門演習ではCを利用してよりまとまなプログラミングやそれをもとにWebサーバーで動くプログラムの作成を行います。そして卒業研究では何らかの情報サービスシステムの試作を行う、と言ったことを考えていますが、卒業研究までお付き合いしなくても、これからの社会に出たらそれなりに役に立つでしょう。

## 1.1 演習の受け方

妙な表題ですが、この演習を受ける皆さんへの要望です。皆さんとしては必修だから仕方がないと言う面も否定できないと思います。ただ他の講義や演習でもそうですが、単位のためだけにこれらを受講するのはもったいないです。この演習は大部分を提出された課題のできで評価しています。単位が取れなかったら困るということでは他の人の答えを丸写しして提出する人が少なくありません。しかしそれではこの演習で扱っているものは身につけません。いくら単位が取れてもそれでは意味がありません。そのあたりをよく考えてやってください。提出された課題の評価では、丸写しをした人はできるだけ低く、できが悪くても自力でがんばったと見られるものはできるだけ高くしようとします。これはなかなか難しいし間違えることもあります。間違った場合は指摘してください。

## 1.2 HTMLについて

WWW (World Wide Web)はWWWサーバーとWWWクライアントがHTTP (HyperText Transfer Protocol)でやり取りをします。その内容がHTML (HyperText Markup Language)で記述されています。ブラウザはサーバーから送られてきたHTMLを解釈して表示を行ったりします。よってWWWサーバーから情報を提供するとなるとHTMLの理解が必要となります。

現在は様々な要求を取り入れてHTMLもかなり複雑になりましたが、この演習ではHTMLが考えられた当初から用いられている基本的な部分だけを扱います。それでも初めての人にとっては結構気持ちの悪いものですが我慢してください。この程度のものにめげるようではプログラミングなんて無理です。

幸いなことにWebページを作成するための様々なソフトが利用可能になっています。例えばWordやExcelなどで作成した文書や表をHTMLの形式で保存ができます。将来Webページを急いで作らなければいけないような場合は、そのようなものを利用すると良いかもしれません。

## 1.3 JavaScriptについて

HTMLの拡張の一つとしてJavaScriptを考えることができます。ブラウザで動く簡易プログラミング言語のようなものです。これも本来多彩な機能がありますが、この演習ではそのごく一部を紹介します。またこれをもってプログラミングの導入にしようと考えています。

ブラウザで本格的なプログラミング言語と言えばJavaです。最近のi-mode端末ではi-アプリと呼ばれるJavaで作成したプログラムが動くようになっています。JavaとJavaScriptは名称が似ていますが、ものとしてはかなり違いますので混同しない方が良いでしょう。

またサーバー側でプログラムを動かす仕組みもあります。これについては来年度の専門演習で取り上げたいと思います。

## 2 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。

### 2.1 作成の手順

Webページを作成するためには、まずHTMLのファイルを作成する必要があります。そして手元で見ただけならば後はブラウザで見るだけです。他の人や外部からも見えるようにするにはできたファイルをワークステーションに転送する必要があります。ただし通常は転送する先がないので、それを準備する必要があります。以下その方法を説明します。

#### 2.1.1 準備作業

以下の作業を行うことにより、ワークステーションの各自の領域の中に外部公開用のディレクトリー(フォルダー)が用意されて、外部から参照できるようになります。この作業は一度行えば二度とやる必要はありません。

1. パソコンを起動してメニューの「WSの端末」をクリックする。
2. 端末プログラムのウィンドウが開いて「login:」と表示されるので、自分のグループ名を入力して[↵]を押す。
3. 「name:」と表示されたら自分の登録名を入力して[↵]を押す。
4. 「password:」と表示されたら自分のパスワードを入力して[↵]を押す。
5. すると「椋山女学園 情報ネットワークメニュー」が表示されるので[7]を押す。
6. 「サブメニュー(その他)」が表示されるので[4]を押す。
7. すると次のようなプロンプトが表示されるので、「w3setup[↵]」と入力する。(miwakoの部分は自分の登録名になる。)

```
cc12[miwako]%
```

8. また同じプロンプトが表示されるので、「exit[↵]」と入力する。
9. 「椋山女学園 情報ネットワークメニュー」に戻るので[0]を押す。

### 2.1.2 ファイルの作成

HTMLで記述した内容を入力してファイルを作成するには、Windowsに付属している「メモ帳」を利用することができます。ただし保存する際に注意しないと「.txt」という拡張子が付いてしまいます。HTMLで記述した内容は「.htm」という拡張子の付いたファイル名にします。そのためには名前を付けて保存する際に、「ファイルの種類」のところを必ず「すべてのファイル(\*.\*)」に変更してからファイル名を入力します。またファイル名に漢字を使用した場合正しくリンクできない可能性が高いので、ファイル名は英数字にしましょう。

また「スタート」メニューから「MS-DOSプロンプト」を起動して、

```
C:\WINDOWS>cd ..  
C:\>edit aaa.htm
```

のような感じでaaa.htmというファイルを作成したり修正することもできます。こちらはカーソルのある位置が表示されるので、JavaScriptで何行目かにエラーがあるような場合、その場所を探すのが簡単になります。

### 2.1.3 ワークステーションとファイルのやりとり

作成したデータを外部から見ってもらうためにはワークステーションにファイルを移す必要があります。またワークステーションにあるファイルを取り寄せて何らかの修正を行いたい場合もあるでしょう。そのやり方をここで説明します。

1. ブラウザとしてInternet Explorer(ver.5.5以降)<sup>1</sup>を使用します。Netscapeでは動かないので注意!!
2. WMNを起動して自分のグループ名、登録名、パスワードを入力します。WMNを家庭などの学外から利用する場合は以下のURLを指定してください。

```
http://www.center.sugiyama-u.ac.jp/WMN/
```

3. 画面の左側の「その他」をクリックします。
4. 画面の右側の「ファイルアクセス」をクリックします。
5. 別画面が開いてファイルやフォルダーが表示されますので、その中の「www」という名前のついたフォルダーをダブルクリックします。なお邪魔でもこの後WMNの方は終了してはいけません。
  - ワークステーションへファイルをコピーしたい場合は、該当するファイルのアイコンをドラッグします。
  - ワークステーションへファイルを移動したい場合は、該当するファイルのアイコンをドラッグして、マウスのボタンを離す際にキーボードの「Shift」キーを押しておきます。

<sup>1</sup>設定が違ってもver.5.5でも駄目な場合があります。「ツール」メニューの「インターネットオプション」をクリックして、「詳細設定」のタブをクリックし、「ブラウズ」の項の「FTPサイト用のフォルダービューを使用する」と「ファイアウォールおよびDSLモデムとの互換性を保つためにPASVモードをサポートするFTPを使用する」に印を付けて下さい。

- ワークステーションのファイルをパソコンにコピーしたい場合は、該当するファイルのアイコンを右クリックするとメニューが表示されるので、その中の「フォルダへコピー」をクリックします。すると「フォルダの参照」のウィンドウが開きますので適当な場所を選び「OK」をクリックします。
- ワークステーションのファイルを削除したい場合は、該当するファイルのアイコンを右クリックするとメニューが表示されるので、その中の「削除」をクリックします。すると確認のウィンドウが開きますので「はい」をクリックします。

用が終わった場合は、WMNは左側の「終了」をクリックしてください。アイコンの並んでいるウィンドウは、左上角の[X]をクリックして閉じてください。

#### 2.1.4 ファイル名とURL

wwwフォルダーの下にaaa.htm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.htm`

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またはindex.htm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次に出てくるtitleと言うタグで指定したものが使われます。

## 2.2 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。

### 2.2.1 簡単な例

例えば次のような内容を、aaa.htm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。<Br>
そしてこれは2番目の段落です。
</Body>
</HTML>
```

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。

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。

## 2.2.2 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：この記述はHTMLによるものだとい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。

```
<P>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>
```

```
・ 文章 1
・ 文章 2
```

<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>
```

```
1. 文章 1
2. 文章 2
```

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 -->
```

### 2.2.3 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。

```
<      &lt;
>      &gt;
&      &amp;
"      &quot;
```

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>
```



## 2.2.4 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。

- 他文書へのリンク：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/">椛山女学園大学のホームページ</A>  
<A href="betu.htm">同じフォルダにある betu.htm というファイル</A>  
<A href="/グループ名/登録名/friend.htm">友人の friend.htm というファイル</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="naamae">文</A>文...文
```

naamae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="#naamae">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.htm とい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.htm へ、間違った選択ならば caix.htm へ行くようにリンクを張る。
2. caio.htm とい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.htm へリンクを張る。
3. caix.htm とい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.htm 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。

## 2.2.5 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演習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おう事を考えると、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、

```

```

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、

```

```

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。

```

```

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

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」をクリック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に入れて、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) 画像の上で右クリックをし、出てきたメニューの中の「プロパティ」を選択してそのページの画像のURLを求める。
- (c) 得られた画像のURLをブラウザに入力して、画像のみが表示されるかどうか確認する。
- (d) aaa.htm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。

## 2.3 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。

## 画像ファイルの作成

演習室のメニューにある「D-pixed」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<sup>2</sup>ので後でGIFやJPEGに変換する必要があったり、後で説明する動画GIFに対応していなかったりするので、とりあえずこのソフトで説明します。

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。

<sup>2</sup>Windows98あたりからはGIF形式でも保存可能になったらしい。

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」とい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。ワークステーションに転送するのが容易なように「デスクトップ」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. ワークステーションに画像ファイルを送る方法はHTMLの入ったファイルと全く同じ操作です。
6. 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。ワークステーションにファイルを送ったら、「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ここをクリックします。



### 説明番号

- |    |    |    |    |
|----|----|----|----|
| 1  | 2  | 3  | 4  |
| 5  | 6  | 7  | 8  |
| 9  | 10 | 11 | 12 |
| 13 | 14 | 15 |    |
| 16 | 17 | 18 | 19 |
| 20 | 21 | 22 | 23 |
| 24 | 25 | 26 | 27 |
| 28 | 29 |    |    |

1. 表示されている画像を拡大するとき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。 **Y** 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。

## 演習問題

前章で作成した超簡易 CAI の各ページに適切な大きさと内容の画像を追加せよ。

### 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場合があります。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うまいようです。

#### 演習問題

1. CAI の正解のページ (caio.htm) と誤解のページ (caix.htm) の背景に RGB の指定を用いて薄い色を設定せよ。
2. CAI の問題文のページ (caiq.htm) に対して、背景に画像を設定せよ。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。

## 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）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.htm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

表の指定は、通常の表を示すためと、単に大きさの違うものを綺麗に並べて表示するためによく用いられます。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となります。

表の罫線が必要な場合は、<Table Border>とします。Border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「Border=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<Br>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>dddd</Td><Td>eeee</Td><Td>ffff</Td>
  </Tr>
  <Tr>
    <Td>ggggg</Td><Td>hhhhh</Td><Td>iiii</Td>
  </Tr>
</Table>
```

表のサンプル

aaaaa	bbbbb	ccccc
dddd	eeee	ffff
ggggg	hhhhh	iiii

## 表の指定の追加

表に関してはさらに次のような指定が可能です。

- <Tr>の代わりに<Th>を使用することにより、この行は表の見出し行であることを示せます。この時見た目にはこの行の内容が太字で表示されます。
- <Td nowrap>と指定すると項目の内容がブラウザの表示幅に合わせて改行されないようになります。ただこれを乱用すると表が画面からはみ出して見にくくなります。
- <Td align="right">と指定すると項目の内容が右詰になります。同様に<Td align="middle">と指定すると項目の内容が中央に揃います。
- <Td valign="top">と指定すると項目の内容が上に寄せられます。同様に<Td valign="bottom">と指定すると項目の内容が下に寄せられます。
- nowrap、align、valignなどの指定は自由に組み合わせて使用できます。

## 演習問題

aaa.htm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 i-mode 端末のようなこの機能に対応していないものを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。

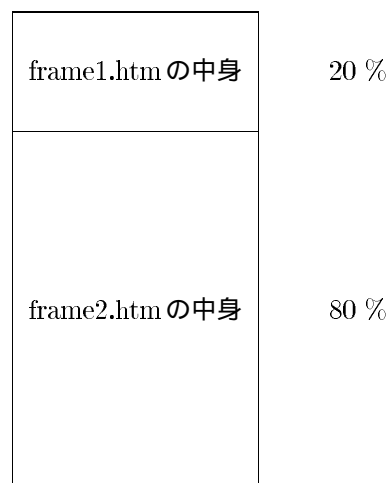
```
<Frame src="URL" name="フレーム名">
```

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。

- <NoFrames>から</NoFrames>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 Navigator 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。また最近では i-mode 端末向けのメッセージをここに入れてある例もあります。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.htm" name="FRAME1">
  <Frame src="frame2.htm" 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.htmの内容とframe2.htm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.htm、右側はfirst.htmを20:80で表示するようにせよ。(index.htm)
2. 目次の中身を作成せよ。これまで作成したファイル名などを並べて、それを選択すると右側の内容がそれに変わるようにせよ。(menu.htm)
3. 最初のページの中身を作成せよ。(first.htm)
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.htmを表示するようにせよ。(menu.htm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ください。

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」というウィ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。

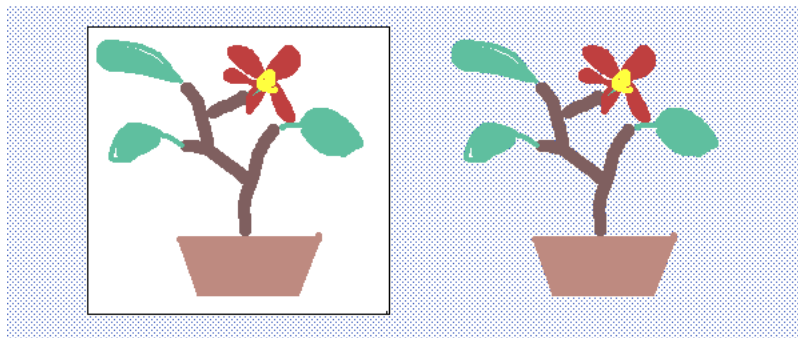
6. 前に作成したレイヤを見るときには「レイヤ管理」の中の「レイヤの選択」の中の見たいレイヤをクリックすれば可能です。
7. 「ウインドウ」メニューの中の「ぱたぱたアニメ」をクリックすると今入力した画像が順番に表示されます。より速い動きを実現したいときには、「レイヤ管理」ウインドウの中の表示時間の所をより小さな値にします。
8. ファイルとして保存する際には必ず GIF 形式を選択し、動画 GIF の部分に印を付けます。

## 演習問題

最低5レイヤからなる動く画像を作成し、first.htm 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) というものが一つあり、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

前節で作成した動く画像の背景を透明にせよ。

## 4 JavaScript 入門

HTML だけでは利用者ができることは自分の好きなリンクをクリックすることだけです。利用者の入力に応じて変化する Web ページを作成するには、

- サーバー側で処理を行う。(CGI: Common Gateway Interface)
- ブラウザ側で処理を行う。(JavaScript、Java)

などの方法があります。CGI を行うにはサーバー側でプログラムを作成する必要があります。またよほど単純な事でない限りブラウザ側だけで片付くことはまず無いので、通常は CGI と JavaScript などが組み合わせて用いられます。Java は本格的なプログラミング言語なので、ここでは JavaScript を用います。ブラウザ側で簡単な処理を行う例によってプログラミングの初歩の勉強も兼ねたいと思います。JavaScript は、C 言語に似ており、Object 指向の部分もかなりあります。

### 4.1 どこに入れるのか

JavaScript による記述はこれまでの HTML の入っていたファイルの中に入れます。Head のタグや Body のタグに囲まれた中に入れます。ファイルは最初から順番に読み込まれますので、関数の定義(後述)以外の部分はその時に順番に実行されます。関数は利用の前に定義されている必要があるため、関数の定義はよく Head タグに囲まれた中に入れます。また表示されたときにすぐ実行される部分は Body のタグに囲まれた中に入れます。

### 4.2 どのように入れるのか

これまでの HTML の記述と JavaScript の記述が混ざらないように、JavaScript の記述は Script というタグに囲まれた中に入れます。さらに JavaScript に対応していない、つまり Script というタグを無視するブラウザの誤動作を防ぐために HTML のコメントのタグも入れるのが普通です。よって通常以下のような形で挿入します。

```
<Script language="JavaScript">
<!--
  JavaScript の記述
//-->
</Script>
```

1 行目と 5 行目は HTML のタグです。このタグの間は language で指定した言語で記述されている事を示します。ただこの Script タグを無視するブラウザのために、<!--と-->という HTML のコメントのタグが入れてあります。JavaScript は<!--は無視することになっています。-->は無視しないのでその前に//という JavaScript のコメントの指示が付いています。JavaScript では//以降の終わりまではコメントと見なして無視するようになっています。

### 4.3 画面への出力

JavaScript でブラウザの画面になにか出力した場合は次のような文を利用します。

```
document.write(出力内容);
```

出力内容としては、"..."、'...' や 3+5 のような式が指定できます。つまり出したい内容を「"」または「'」で囲います。また複数出したい場合はこれらを「,」で区切ります。出したい内容の中に HTML のタグを含めることもできます。

document はブラウザの中の一つの object です。ちょうど今表示されているもの全体として捉えたら良いでしょう。この document に write() をくっつける事により document に write をしてくれと依頼することになります。write() のように () が後に付くものはなんらかの動作を行う関数であることを示します。なお最後の「;」は文の区切りです。忘れないようにしてください。

例えば次のような記述を入れると、入れた場所に「3+5=8」と表示されます。

```
<Script language="JavaScript">
<!--
    document.write("3+5=",3+5);
//-->
</Script>
```

## 演習問題

画面に JavaScript で次のようなものが表示されるようにせよ。ただし最後の 7 は 3+4 の結果として大きく出るようにせよ。(love.htm)

```
こんにちは
He said, "I love you."
```

7

## 4.4 計算式

JavaScript では通常の四則演算に加えて剰余が可能になっています。加減乗除に対してそれぞれ「+」、「-」、「\*」、「/」が Excel などと同じように対応しており、剰余に対しては「%」を使用します。また () も使用可能なので、計算の順番を指定する際に使います。

ちょっと問題があるのは「+」です。通常の数値の加算以外に文字列の連結という意味があります。つまり 2+3 ならば 5 になるのですが、「2"+"3" とすると "23" になってしまいます。他の演算では文字列同士を計算しようとすると結果が「NaN」などになって正しい結果は得られません。

## 4.5 フォーム

フォームは利用者がデータを入れるための HTML のタグです。本来はここへ入力したものが Web サーバーに送られて CGI で処理されますが、Web サーバーに送る前に JavaScript でデータを処理することも可能です。ここで説明するタグは HTML のものなので Script タグの間やコメントのタグの間に入れないようにしてください。また JavaScript で利用する際には必ずフォームのタグで入れ物を出してから JavaScript が実行されるようにしてください。

まずフォームの記述全体を次のようなタグで囲います。

```
<Form name="名前">
    フォームの内容
</Form>
```

多くの人が from と間違えるので注意してください。「名前」の部分には適当な英数字によるものを入れま  
す。違うフォームには違う名前を付けてください。フォームの内容としては色々なものがあるのですが、と  
りあえず入力するためのものは次のような形になります。

```
<Input name="名前">
```

複数の入力欄を設ける場合にはそれぞれ異なる「名前」を付けてください。これによってブラウザによっ  
て適当な大きさの入力欄だけが表示されます。次のようにして大きさの指定も可能です。

```
<Input name="名前" size=文字数>
```

なおここで指定した「文字数」は表示際の入力欄の大きさであり、入力欄には通常これ以上の文字数を入  
れることが可能です。さらにあらかじめ入力欄に何か内容を入れておくことができます。

```
<Input name="名前" value="内容">
```

この場合「内容」に記述したものが、最初から入った形になります。value の指定が無い場合は空欄になり  
ます。

#### 4.6 入力欄への入出力

入力欄へはクリックしてカーソルを出してからキーボードで入力することができます。ここでは JavaScript  
によって入力欄に何か出力したり、逆に入力欄に入っている内容を取り出したりする方法を説明します。  
まず出力するには、

```
document.フォームの名前.入力欄の名前.value=出力内容;
```

とします。大抵のブラウザはこれで動きます。ただし「フォームの名前」や「入力欄の名前」がシステム  
の持っている別のものの名前と偶然一致したような場合に誤動作する恐れがあります。心配な人は、

```
document.forms['フォームの名前'].elements['入力欄の名前'].value=出力内容;
```

のようにしてください。また内容を取り出す場合には=の左側と同じものを使います。ただし、入力欄の  
内容が「123」と言うような数字の場合でも文字列として扱われますので注意します。数値に直したい場合  
は、eval() という関数を利用するのが正式なやり方ですが、数値を掛けることによっても直ります。

#### 演習問題

3つの入力欄を定義し、JavaScript でそのうちの2つにはそれぞれ123と456を入れて、最後の入力欄に  
は他の2つの入力欄の内容の合計が入るようにせよ。(xy.htm)

x	<input type="text" value="123"/>
y	<input type="text" value="456"/>
x+y	<input type="text" value="579"/>

## 4.7 関数の定義(1)

他のプログラミング言語などと比べてJavaScriptでは関数を使えることはかなり重要な意味を持っています。つまり関数がないと利用者からの指示に応じて何かすることができません。例えばどこかのボタンをクリックすると計算をしてくれるというような場合、その計算の内容を関数として定義して、ボタンにはクリックされたらその関数を呼び出せという指示を付けます。とりあえずここでは関数の定義方法を説明し、次章でボタンの方の説明をします。なおJavaScript自体でもかなりの数の関数を既に持っています。それらで済む場合は自分で関数を定義する必要はありません。

関数を定義するには次のように書きます。

```
function 関数名(){  
    関数の中身  
}
```

関数名は英数字です。関数の中身としてはJavaScriptの文を書きます。この記述は必ず関数の呼び出しに先行する必要があるため、よく<Head>のタグの間に書かれます。これまでのJavaScriptの記述と異なりこれを書いただけでは関数の中身は実行されません。

関数の中身を実行したいところに、

```
関数名();
```

を書きます。するとここに関数の中身を書き移したのと同じように実行されます。

## 4.8 ボタン

Webページに見られるボタンはHTMLのタグで作られます。また本来Formタグの中だけで有効なものなので記述する際は、必ず<Form>と</Form>の間に入れます。ボタンを置きたいところに、

```
<Input type="button" value="ボタンの文字" onClick="関数名()">
```

を記述すると、「ボタンの文字」で指定した文字の付いたボタンができ、これをクリックすると「関数名()」で指定した関数が実行されます。ここで指定する関数は通常自分で定義した関数ですが、JavaScriptが持っている関数でもかまいません。

### 演習問題

次のようなボタンの付いたページを作成せよ。(button.htm)

1. クリックすると「window.close()」を呼び出す「とじる」と文字の付いたボタン。
2. クリックすると2つの入力欄に入れた数値の合計を3つめの入力欄に表示する「合計」と文字の付いたボタン。

The screenshot shows a web form with three input fields and two buttons. The first button is labeled 'とじる' (Close) and is positioned above the first input field. The input fields are labeled 'x', 'y', and 'x+y'. The second button is labeled '合計' (Total) and is positioned to the right of the 'x+y' input field.



## 4.9 条件判断

コンピュータは昔は「電子計算機」と呼ばれていました。でもコンピュータと電卓は違います。共に計算ができますが、コンピュータはさらに条件判断が可能です。と言っても曖昧な判断は無理でyesかnoかの論理的な判断しかできません。

判断の基準は2つのものの比較です。次のような比較演算子が利用可能です。

<	未満	>	超過
<=	以下	>=	以上
==	等しい	!=	等しくない

これらを利用することにより2つのものの比較ができますが、さらに多くのものを比較するために、これらの比較を&&(かつ)や|| (または)でつなぐことができます。

このような条件判定を次のようなif文で使います。

```
if (条件) {
    条件が成立したときに実行する内容
}
```

もし条件が成立しなかった場合は何も実行されません。さらに条件が不成立の場合に別の事をしたい場合は次のように記述します。

```
if (条件) {
    条件が成立したときに実行する内容
} else {
    条件が不成立のときに実行する内容
}
```

さらにif文の中にif文を入れることも可能です。次はifを利用した例の一部です。どのような場合にどのようなメッセージが表示されるか考えてください。

```
if (document.aaa.sex.value=="男") {
    if (document.aaa.looks.value=="Good") {
        document.aaa.message.value="携帯の番号教えてくださいませんか? ";
    } else {
        document.aaa.message.value="。。。 ";
    }
} else {
    document.aaa.message.value="こんにちは";
}
```

### 演習問題

1. 上記のifの例において、「。。。」のところも「こんにちは」にするとすると、全体的にもう少し簡単な形に直すことができる。どのようなかたちになるか。
2. 入力欄に身長と体重を入力してから「判定」のボタンをクリックすると、 $(身長 - 100) \times 0.9 < 体重$  ならば判定結果の欄に「でぶ」、そうでなければ「すまーと」と出るようにせよ。(hantei.htm)

身長 (cm)	<input type="text"/>
体重 (Kg)	<input type="text"/>
	<input type="button" value="判定"/>
判定結果	<input type="text"/>

#### 4.10 警告・確認画面

一旦ページが表示された後に、ページの内容で変更することができる場所は限られます。そのうちの一つはFormのタグの中の入力欄でした。ただこれはあらかじめ入力欄を出しておかなければならないのと、あまり目立たないので中身を変更しても気がついてもらえない可能性があるなどの問題点があります。ここで説明する警告・確認画面は、必要なときに画面の中央に指定した内容を表示してくれるのでそのような問題点がありません。

##### 警告画面

次のような記述が実行されると画面に別ウィンドウが開きます。

```
window.alert(表示内容);
```

「表示内容」の部分には「”」や「'」で囲った文字列や式などが「,」で区切れば複数指定することができます。次の画面の例は、表示内容として”本当に好きですか?”を指定した場合です。これを見た利用者が「OK」のボタンをクリックするまでJavaScriptの実行も停止します。



##### 確認画面

間違ってボタンをクリックしたためにデータを全て失うことがあります。そのような不幸なことにならないように、後戻りが簡単に行えないような処理を始める前に確認画面を出すようにしましょう。利用者が表示された内容を見て「OK」または「キャンセル」のボタンをクリックするので、ifを併用してその後に行う内容を切り替えます。

```
if (window.confirm(質問内容)==true){  
    「OK」をクリックした場合にする内容  
} else {  
    「キャンセル」をクリックした場合にする内容  
}
```

「質問内容」の部分が表示されます。指定できるものは警告画面のものと同じです。「キャンセル」をクリックした場合何もしないのであればelse以下を省略することができます。以下は質問内容として”結婚してくれますか?”を指定した場合です。



## 演習問題

1. 確認画面において「キャンセル」をクリックした場合のみ何かをするならばどのように記述すれば良いか。
2. 前節の課題 (hantei.htm) に身長や体重の欄に入力せずに「判定」のボタンをクリックしたときに警告画面が出るようにせよ。さらにその場合に誤った判定結果が出ないようにすること。そのためには表示の後で「return;」を実行するようにする。return文は関数の実行を中断する働きがある。
3. さらに前節の課題 (hantei.htm) で計算の結果「でぶ」となった場合にそれを表示する前に「本当に知りたいですか?」と確認の画面を出すようにせよ。当然「キャンセル」をクリックした場合には結果を表示しない。

### 4.11 変数

プログラミング言語には必ず「変数」というものが出てきます。ただの数ならばいつも同じ値なので問題ありませんが「変数」は字のごとく値が変化します。さらに表計算のセルの値のように直接値を見ることができません。とすることで「変数」はプログラミングを勉強する人にとっての一つの越えなければならぬ壁であります。このテキストでは壁はできるだけ後回しとすることでやってきましたがそろそろ限界なのでやります。

「変数」は名前の付いたメモのようなものです。メモには数値または文字を一つ書くことができます。書いた内容は読むことができます。一つしか書けないので新しい内容を書き込むと以前の内容は失われます。

「変数」を利用するためには通常「宣言」が必要となります。「宣言」によってどのような名前で、どのような形式(数値や文字)の内容が書き込めるかということを示します。プログラミング言語によっては「宣言」を必要としないもの、形式の指定が不要なものもあります。JavaScriptは宣言は必要ですが形式の指定が不要なものに属します。

宣言を行うとある範囲でその変数が使用できるようになります。関数の中で宣言を行うと関数の中でのみ使用可能になります。関数の外で宣言をするとその行以降で使用可能になります。ただどこで宣言しても通常のプログラミング言語では、変数の有効なのはそのプログラムの実行中に限定されますので注意してください。JavaScriptの場合は、その記述があるページが表示されている間になります。

JavaScriptでの変数の宣言の仕方は次のようになります。

```
var 変数名, 変数名, ... ;
```

「変数名」として使用可能なものは英数字と「\_」です。このような形で複数の変数を一度に使えるようにできます。またこのように宣言した変数には数値や文字など一つの変数一つに限られますがなんでも入れることができます。また次のように宣言と同時に変数に値を入れておくことができます。

```
var x=100;
```

通常宣言をただけの変数はどのような値を持つかわからないのでこのように最初から入れておくと安心です。

## 演習問題

1. 次のような  $x$  の値を入力してから「計算」のボタンをクリックすると  $x^2 + 3x + 4$  を計算するものを作れ。(keisan.htm)



A screenshot of a web browser window showing a form. The form has two input fields and one button. The first input field is labeled 'x=' and is empty. To its right is a button labeled '計算'. Below the first input field is a second input field labeled 'x^2+3x+4=' which is also empty.

2. 「+1」のボタンをクリックするとその数字が一つずつ増えていくようなものを作れ。ただし=の右側に document があってはならない。(count.htm)



A screenshot of a web browser window showing a form. The form has one input field and one button. The input field is labeled 'カウント:' and contains the number '0'. To its right is a button labeled '+1'.

3. 最初にゼロが表示されるようにせよ。(count.htm)
4. さらに数字が9まで増えたら次にゼロに戻るようにせよ。(count.htm)

## 4.12 関数の定義 (2)

中学校か高等学校で数学の時間に習った「関数」と JavaScript の関数はかなり感じが違うという印象を受けたと思いますが、普通の「関数」と同じようなこと、つまり何らかの値をもらって計算をして結果を返すようなこと、も JavaScript の関数でも可能です。前述の `window.confirm()` は、指定された内容を受け取ってそれを表示し、クリックされたボタンに応じて `true` または `false` という値を返すので、それを元に `if` 文で処理内容を変えることができました。これが表示できる内容は指定できないとか、どのボタンをクリックされたのかわからないのではちっともありがたくありません。

自分の定義する関数で同様の事を行う場合には次のように記述します。

```
function 関数名(変数名){  
  関数の内容  
}
```

このような形にすると関数を呼び出す側から数値や文字などを渡すことができます。複数のものを渡したい場合は、「変数名」のところを「変数名, 変数名, ...」のようにします。また計算の結果を返すならば、

```
return 返す値;
```

を関数の内容に入れます。ただし `return` は結果を返す作用以外に関数の実行を中断する働きもありますので注意します。

```
function add5(x){ // もらった値は変数 x に入る
    return x+5;
}

....

y=add5(3); // y には 3+5 つまり 8 が入る。
z=add5(y); // z には y+5 つまり 13 が入る。
```

## 演習問題

入力欄の値とメッセージを渡すと、もし入力欄の値が空だともらったメッセージを警告画面で表示し true を返す関数 (check) を作れ。空でないときは単に false を返すこと。(hantei2.htm)

この関数は以下のように使うことを想定している。本来ならば ==true が必要だがこのように省略もできる。

```
if (check(document.aaa.h.value,"身長を入れてください")) return;
if (check(document.aaa.w.value,"体重を入れてください")) return;
```

## 4.13 繰り返し

コンピュータは昔「電子計算機」とも呼ばれたように高速に計算ができます。しかしいくら高速でも、計算すべき式が少ししかなければあまり意味がありません。また式がたくさんあってもそれをいちいち入力しなければならないのでは大変です。変数があるので計算結果をもう一度入力すると言うような事は避けられますが、これだけでは足りません。プログラミング言語では通常繰り返しと言うものが簡単に記述できるようになっており、数行のプログラムで何億回も計算させると言うような事が可能です。

たとえば JavaScript では次のような記述で「I love you.」を 100 回表示することができます。

```
var i;
for (i=0;i<100;i++){
    document.write(i," I love you.<Br>");
}
```

これは for 文と呼ばれるもので類似のものが通常のプログラム言語では必ず存在します。JavaScript では for 文は、for([A];[B];[C]) { [D] } のようなちょっと複雑な形をしており、次のような意味になります。

1. [A] を実行します。ここには通常代入文が入ります。
2. [B] の条件を調べます。もし条件不成立の場合は for 文は終了します。
3. [D] を実行します。ここには任意の複数の文を書くことができます。
4. [C] を実行します。ここには例のような式または代入文が入ります。
5. 2 番目に戻る。

[B] が条件であること、[C] と [D] が書いてある順番と逆に実行されることに注意してください。上記の実例の場合次のような感じで実行されます。

1. 「i=0」を実行するので変数iの値がゼロになる。
2. 「i<100」の条件を調べると、変数iの値はゼロなので条件は成立する。
3. 「document.write()」を実行するので「0 I love you.」と表示する。
4. 「i++」を実行する。これは「i=i+1」と同じ意味なので、変数iの値は1になる。
5. 「i<100」の条件に戻る。変数iの値は1なので条件は成立する。
6. 「document.write()」を実行するので「1 I love you.」と表示する。
7. 「i++」を実行する。変数iの値は2になる。
8. 繰り返すたびに変数iの値は増加していく。
9. 「99 I love you.」と表示した後変数iの値は100になる。
10. 「i<100」を満たさなくなるのでfor文は終了する。

もし1000回「I love you」と出したい場合は、「i<1000」に変更します。iの値をどんどん減らしたい場合は、「i++」の代わりに「i--」を使用します。これは「i=i-1」と同じ意味です。

## 演習問題

1. 画面上に「1 3 5 7 9 ... 999」と表示するもの。数字ごとに改行しなくても良い。(kisuu.htm)
2. 画面上に「1000 999 998 997 ... 3 2 1 0 ドカン」と表示するもの。数字ごとに改行しなくても良い。(countdown.htm)
3. 画面上に次のように10ごとに行を変えた1から100までの数を表示するもの。for文を複数使う手もあるが、for文は一つでやること。(10x10.htm)

```

1 2 3 4 5 6 7 8 9 10
11 12 13 14 15 16 17 18 19 20
21 22 23 24 25 26 27 28 29 30
31 32 33 34 35 36 37 38 39 40
41 42 43 44 45 46 47 48 49 50
51 52 53 54 55 56 57 58 59 60
61 62 63 64 65 66 67 68 69 70
71 72 73 74 75 76 77 78 79 80
81 82 83 84 85 86 87 88 89 90
91 92 93 94 95 96 97 98 99 100

```

4. 前問と同様のものをfor文を2つ利用し、かつ数字の先頭が揃うようにせよ。(10x10f.htm)

## 4.14 選択メニュー

入力される内容があらかじめ数種類に限定される場合にはラジオボタンやここで説明する選択メニューが用いられます。入力欄を用いた入力では想定外の入力が避けられませんが、選択メニューではこちらが用意したものしか選べないのでそのような事がなく、また入力者にとってもマウスで選択するだけなので楽です。ただ県名のように多数あるものとなると、マウスで選択するのも大変になるので注意が必要です。

まず選択メニュー自体はHTMLのタグです。Formタグの有効範囲の中で使用できます。例えば、

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

のように記述すると、



のように表示されます。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、この項目が最初から表示されます。またvalueの指定をすると、あとでJavaScriptで値を取り出すことができます。(この例では「走る」の項目のみ「run」と言う値を持つ。)

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>
```

2つめ以降の項目を選択する際にはCtrlを押しながらクリックします。この場合にはselectedを複数指定しても構いません。またJavaScriptの方で複数選択した場合に対応できないといけません。

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。

さて、このようにして作成した選択メニューでどのような選択が行われたかは次のようにしてJavaScriptで調べることができます。

1. 選択肢の数は、「document.フォーム名.セレクト名.options.length」<sup>3</sup>でわかります。なお、「セレクト名」はSelectタグで指定した名前のことです。通常これを元にして次のようにfor文を回します。

```
for (i=0;i<document.フォーム名.セレクト名.options.length;i++){
```

2. 選択されたかどうかは、「document.フォーム名.セレクト名.options[数字].selected」<sup>4</sup>がtrueであるかどうかでわかります。「数字」の部分には、ゼロから選択肢の数-1が入ります。複数選択されている場合は複数のtrueが、全く選択されていない場合は、trueが一つも無いこともあるので注意します。通常は次のようにこれをif文の条件のところに入れて、選択された項目のみ処理するようにします。

<sup>3</sup>きちんと書くならば、document.forms[フォーム名].elements[セレクト名].options.lengthになります。

<sup>4</sup>きちんと書くならば、document.forms[フォーム名].elements[セレクト名].options[数字].selectedになります。

```
if (document.フォーム名.セレクト名.options[i].selected){
    i 番目の項目が選択されている場合の処理内容
}
```

3. 選択肢に設定されている値は、「document.フォーム名.セレクト名.options[数字].value」<sup>5</sup>で知ることができます。「数字」の意味はselectedと同じです。
4. 選択を変更したら処理をしたい場合には、Selectのタグの中にonChangeという指定を入れます。

```
<Select name="selone" onChange="関数名">
```

## 演習問題

1. Mバーガーのバーガーセット注文画面を以下のように作成してみよ。なおMバーガーのバーガーセットは、ハンバーガー(59円)、ポテト(150円)、ジュース(160円)のそれぞれ1個または0個の組み合わせである。同じ組み合わせのものを複数セット一度に注文できるものとする。(barger.htm)

ミキドナルドバーガーセット注文表

お好きな品をクリックしてください。複数品目を選択する際には、Ctrlを押しながらクリックしてください。

ハンバーガー  
ポテト  
ジュース

を 1 セット

注文の内容が決まりましたら、「合計」ボタンをクリックしてください。

合計  円

2. 前問と同様のものをfor文を利用し、かつチーズバーガー(79円)をメニューに追加する際に、selectタグのsizeのところを直し、Optionタグの行を追加するだけで、JavaScriptの記述には手を加えなくても良いようにせよ。
3. 選択内容が変更されるたびに合計金額も更新されるようにせよ。
4. セット数の部分も選択メニューで1から10まで選択できるようにせよ。なおこの選択メニューを記述するOptionタグの部分は規則的なのでJavaScriptで出すようにせよ。

## 4.15 タイマー割り込み

コンピュータの分野で「割り込み」と言うのは結構重要な機能です。コンピュータは誕生以来かなり初期の頃から回りの人や機械と比べてかなり高速でした。その結果、人からの入力、テープ装置からの読み込み、プリンターへの印字などの際には速度を合わせるための待ち時間が実行時間全体の99%以上を占めると言うことも珍しくありませんでした。そこでその待ち時間を有効に利用するために他の仕事も並行して行おうとしましたが、他の仕事の方に夢中になってしまうと困ります。「割り込み」機能はちょうど我々が

<sup>5</sup>きちんと書くならば、document.forms[フォーム名].elements[セレクト名].options[数字].valueになります。



用いる電話のように、本来の仕事をする必要が生じたときに、割り込むことができるようなものです。これによって他の仕事に夢中になっていても構わなくなります。

JavaScriptのonClickやonChangeも広い意味では割り込みです。他の仕事は特にしていませんが、これらの指示をしておくことでマウスの操作で関数の実行を割り込ませることができました。ここで説明するタイマー割り込みも似たようなものですが、関数実行の機会は指定した時間が経過するとやってきます。この機能を利用して実務的には時間切れの処理、趣味的には画面をどんどん変化させるのに用いられます。

タイマー割り込みの設定には次のような関数を用います。

```
x=setTimeout(関数名,時間);
```

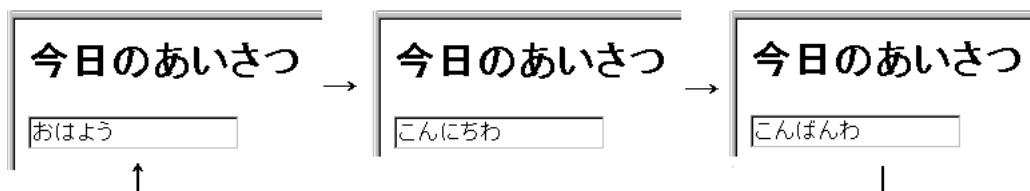
「関数名」のところには時間が経過したときに呼び出す関数を「owari()」のような感じで記述します。「時間」のところには、関数を呼び出すまで待つ時間をミリ秒単位で記述します。たとえば3秒後に呼び出すならば3000と書きます。ただしこの時間に関しては様々な要因が絡むのでそれほど正確ではありません。「x」は任意の変数で構いませんが、これにタイマーの識別子(ID)が入ります。これを使用して次のようにすると設定したタイマー割り込みを解除することができます。

```
clearTimeout(x);
```

注意しなければならないのは一度関数が呼び出されるとこのタイマー割り込みの設定がなくなることです。繰り返し呼び出す場合は、呼び出された関数の中で再びsetTimeoutの指示を行う必要があります。

## 演習問題

1. 同じ入力欄に3つのメッセージを2秒間隔で表示するものを作れ。なお最後のメッセージを出した後はやはり2秒後に最初のメッセージに戻り、永遠に続くようにする。(display.htm)



2. 次のような感じで問題を出し、

- 3秒以内に答えなかった場合には、「時間切れです」と言う警告画面を出す。
- 「yes」をクリックした場合には、「                    」と言う警告画面を出す。
- 「no」をクリックした場合には、「                    」と言う警告画面を出す。

もちろん時間以内に答えた場合は、「時間切れ」はでないようにする。(quiz.htm)

### 難しい問題

3秒以内にお答え下さい。  
三木先生はかっこいいと思うか？

## 4.16 Window の操作

アクセスすると本体以外に別の Window が開くページがあります。大抵は CM のページなので邪魔なだけですが、場合によってはそれを閉じるとさらに別の CM のページが出てくるといったものもあります。ここでは Window に関する様々な操作法について説明します。

まず別 Window の作成の仕方です。次のような感じで新しい Window を開くことができます。

```
var w;  
w=window.open(URL,windowID,option);
```

URL には、http://... とかファイル名が入ります。windowID は window に付ける名前前で英数字で適当な名前を付けます。複数の Window を開く場合には違う名前にします。またこれを target で指定することにより、クリックすると別 Window に表示する様なこともできます。option の部分には、様々な指示を""で囲って、', ' で区切って複数指定することができます。(例えば、"width=200,height=100" のような感じ。)

指示名	内容	例
menubar	メニューバーを付けるかどうか	menubar=yes
toolbar	ツールバーを付けるかどうか	toolbar=yes
location	URL の表示を付けるかどうか	location=yes
directories	リンクのところを付けるかどうか	directories=yes
status	window の下に状況表示を出すかどうか	status=yes
scrollbars	スクロールバーを付けるかどうか	scrollbars=yes
resizeable	大きさを変更できるようにするかどうか	resizeable=yes
width	Window の幅	width=300
height	Window の高さ	height=300

何か option の指定をした場合、yes にしなかったものは表示されません。ただブラウザによっては出てくることもあるので no と指定した方が良いかもしれません。なお、URL、windowID、option の指定は省略可能です。

上記の例で w という変数には生成された window が設定されます。これを利用して新しくできた window を操作することができます。例えば、

```
w.close();
```

を実行するとその window を閉じる (消す) ことができます。window は手動で消すこともできるので、操作しようと思ってもものが既にあることもあります。そのような場合に対処するには、

```
if (w.closed) {  
    そのwindowが閉じられていた場合の処理  
}
```

のように記述します。

URL を指定する場合は新しい window の中味は別ファイルに入れることになりますが、URL を指定しない場合は、open したあとに続けて document.write でその内容を送ることもできます。

```

var w;

w=window.open("", "", "width=300,height=200");
w.document.write("<Head><Title>タイトル</Title></Head>");
w.document.write("<Body><H1>タイトルだけ</H1></Body>");

```

この例からもわかるように、先頭に「w.」を付けることにより、これまでやったことが大抵そのまま新しく作った Window に対して行うことができます。逆に新しく作られた Window から逆に元の Window に対して指示を出す事ができます。その場合には「window.opener.」を付けることとなります。例えば次のような関数を実行すると、元の Window に楢山のトップページを出す事ができます。

```

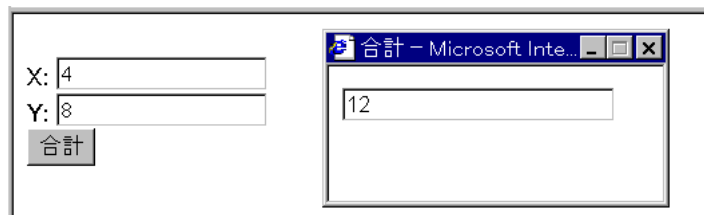
function change(){
    window.opener.location="http://www.sugiyama-u.ac.jp/";
}

```

注意：document.write で指定する文字列の中に</Script>がある場合は、<\Script>のように記述してください。そうしないとここで JavaScript が終了したと見なされて変なエラーになります。

## 演習問題

- 「Open 1」と書かれたボタンをクリックすると、大きさが600×400で縦横のスクロールバーのみが付いた別 Window に楢山のトップページ (<http://www.sugiyama-u.ac.jp/>) が出るようにせよ。(window1.htm)
- 「Open 2」と書かれたボタンをクリックすると別 Window に楢山のトップページが出るようにし、さらに元の Window にある「学園センター」というリンクをクリックすると別 Window に表示されているものが学園センターのトップページ (<http://www.center.sugiyama-u.ac.jp/>) に変るようにせよ。(window2.htm)
- 「Open 3」と書かれたボタンをクリックすると別 Window が開き、そこに「閉じる」と書いたボタンが表示されるようにせよ。なおこのボタンをクリックすると別 Window は閉じられる。また元の Window にも「閉じる」と書いたボタンを置き、これをクリックするとやはり別 Window が閉じるものとする。(window3.htm)
- 入力欄を2つ設けて、そこに数値を入力した上で「合計」と書かれたボタンをクリックすると、「合計」というタイトルの付いた別 Window が開き、その中に数値の和が表示されるようにせよ。なお既に別 Window がある場合は改めて別 Window は開かないようにする。ヒント:あらかじめwに適切な値を入れておくとwindow.openに使用したか、まだ使用していないかどうかの判定ができる。(window4.htm)



## 5 Style Sheet 入門

ここではWebページの体裁を整えるために1996年末に導入されたCSS (Cascading Style Sheet)について簡単に説明します。当初研究成果の自由な交換を考えて作られたWWWでしたがインターネットで広く使われるに従い、「より美しく」、「どのような環境でも作成者の意図したとおり」に表示させたいという要求は強くなりました。そのために当初はHTMLのタグの拡張と言う形で行われてきましたが、情報内容の構造的なものを示すタグで行うのは適切でないということになり、Style Sheetで行うことになりました。

「より美しく」と言うようなデザイン的なことになると、細かく実際のものを見ながらやるのが一番です。よってホームページビルダーなどの作成用ソフトを使用するのが良いでしょう。ただマウスできっちり同じ形を描くと言うのも結構面倒なものです。数字で大きさを指定した方が速いことも少なくありません。

「どのような環境でも」と言うのも悪くはありません。せっかくのデザインがブラウザの設定によってがたがたになってしまうことも少なくありません。しかし、例えば文字の大きさをきっちり決めてしまうと、目の良い人はもう少し小さい字ならば全体が見えて良いと思うし、目の悪い人は字を大きくして見ることができたら読みやすいと思うでしょう。画面の大きさも様々ですので、高性能・大画面の開発用マシンでデザインをしたら、普通のパソコンで見る時に画面をスクロールしないと文章が読めないとかがよくあります。

### 5.1 どこに入れるのか

後述のStyle Sheetの様々な指定(スタイル)は3種類の入れ方があります。

- 別ファイルに入れる：指定だけを別ファイルに入れて、それを読み込んで利用することができます。同じように形式を整えたいWebページが複数ある場合にはこれが一番良いでしょう。なぜならばファイルの一つ直すだけで、それを取り込んでいる全てのファイルに形式の変更が及ぶからです。

各WebページのHeadのタグの間に次のような内容を入れます。

```
<Link rel="stylesheet" type="text/css" href="ファイル名.css">
```

そして指定は「ファイル名.css」という名前のファイルに入れます。

- ファイル全体に対して指定する：Headタグの間に次のような形で指定を入れます。その指定はBodyタグの間全体に有効となります。JavaScriptと同じような感じです。

```
<Style type="text/css">
<!--
スタイル
-->
</Style>
```

- 個々のタグに対して指定する：HTMLのタグに対して指定することができます。この場合指定が及ぶ範囲は、そのタグの範囲に限られます。これは次のような形になります。

```
<タグ style="スタイル">
```

これらの指定は任意に組み合わせて使用することができます。後のものほど有効範囲が狭いので、最初のもので各Webページ共通部分を指定し、最後のものでも個別の違いに対応と言った感じの使い分けをします。

## 5.2 基本的な形式

スタイルの基本的な形は次のようになっています。

```
セクタ{属性:値}  
セクタ{属性:値; 属性:値}
```

セクタは、スタイルを設定する対象(各種タグ名、クラスなど)です。属性は、適用したいスタイルの属性を示し、値はその属性をどうするのか(例えば大きさ、色など)を示します。また複数の属性について指定する場合は、;で区切ります。具体的には、

```
H1{color: red}
```

によってH1のタグで囲まれた文の色を赤にすることができます。具体的には次のように指定します。

```
<HTML>  
<Head>  
<Title>Style のテスト</Title>  
<Style type="text/css">  
  <!--  
    H1{color: red}  
  -->  
</Style>  
</Head>  
  
<Body>  
<H1>これは赤になる</H1>  
<H1 style="color: blue">これは青</H1>  
<H1>またまた赤になる</H1>  
</Body>  
</HTML>
```

color以外の属性としては次のようなものがあります。 には数値が入ります。

属性	説明	値
background-color	背景色を指定	#RRGGBB または色の名前
background-image	背景画像を指定	画像ファイル名
font-style	フォントの書体を指定	italic など
font-weight	フォントの強調を指定	100 ~ 900(100 step)
font-size	フォント大きさを指定	px、 pt、 %、 cm、 mm、
text-decoration	テキストの装飾方法を指定	none、 underline、 line-through など
text-align	文字の位置を指定	left、 right、 center
text-indent	テキストの字下げを指定	px、 pt、 %、 em、 cm、 mm、
line-height	行間の幅を指定	px、 pt、 cm、 mm、

なお、長さの指定の所で使用できる単位の意味は、

em	フォントサイズを1とした場合の相対指定
%	フォントサイズや画面サイズを100%とした場合の相対指定
mm	ミリメートルで指定
cm	センチメートルで指定
in	インチで指定 (1in = 2.54cm)
px	ピクセル(画像の点)で指定
pt	ポイントで指定 (1pt = 1/72in)
pc	パイカで指定 (1pc = 1/6in)

となっていますが、いくらブラウザやパソコンが1cmにしようとしてもディスプレイの大きさが違ってしまふとなんともなりません。よってem、%、pxなどが良く使われます。

### 演習問題

1. なぜpxが良く使われるのだろうか？
2. ブラウザの画面いっぱい「謹賀新年」と赤い字で出るようにせよ。(gasho.htm)
3. p.17の演習問題で作成した表の各セルの背景に次のように色を設定せよ。「男とは」には青、「女にとって」には赤、他のセルには黄色。(hyou.htm)

### 5.3 クラス分け

HTMLのタグにそれぞれStyleの指定が可能です。もう少し細かく指定することができます。つまり同じタグをクラス分けしてそれぞれについて異なる指定が可能です。例えばAタグで指定するリンクについては、

A:link	未訪問のリンク
A:visited	訪問済みのリンク
A:active	選択中のリンク
A:hover	マウスが上にある時のリンク (IE4.0以上)

のように分けて指定することができます。これによってtext-decoration属性はnoneと言うように指定して下線が出ないようにしたり、A:hoverに特別な指定をすることにより、マウスがリンクの上に来ると形が変るようなものが実現できます。

通常のタグに対して次のようにしてクラス分けをすることができます。

```
セレクトA.クラス名{属性:値}
.クラス名{属性:値}
```

クラス名としては任意の英数字が使用できますが先頭は英字です。「.」の前にクラス名を指定しない場合は、どのようなタグでも使えるクラスになります。一方これを利用するタグの方では、

```
<タグ class="クラス名">
```

のように適用したいクラスの名前を指定します。このようなクラス分けのメリットとしては、例外的な指定もクラスとして指定することにより、統一化できるというものがあります。以下の例ではsmallredというクラスをH1やAで利用しています。

```
<HTML>
<Head>
<Title>クラスのテスト</Title>
<Style type="text/css">
  <!--
  H1{color: yellow}
  H1.special{color: blue}
  .smallred{color: red; font-size: 8pt}
  -->
</Style>
</Head>

<Body>
<H1>これは黄色になる</H1>
<H1 class="special">これは青</H1>
<H1>またまた黄色になる</H1>
<H1 class="smallred">えらい小さい</H1>

さらに他のタグでも使える。たとえば、
<A href="http://www.sugiyama-u.ac.jp/" class="smallred">梶山</A>
などでも。
</Body>
</HTML>
```

## 演習問題

マウスを上を持って行くと赤から緑に変化するリンクを作ってみよ。