

現代マネジメント学科
ビジネス情報演習B テキスト

三木 邦弘

平成28年9月26日

目次

1 はじめに	2	2.15 画像のプロパティ	17
2 JavaScript 入門	3	2.16 Window の操作	18
2.1 どこに入れるのか	3	3 PHP 入門	21
2.2 どのように入れるのか	3	3.1 どこに・どのように入れるのか	21
2.3 画面への出力	4	3.2 表示の命令	22
2.4 計算式	4	3.3 変数・計算式	23
2.5 フォーム	4	3.4 繰り返し (1)	23
2.6 入力欄への入出力	5	3.5 配列型変数	25
2.7 関数の定義 (1)	6	3.6 フォームからの入力	25
2.8 ボタン	7	3.7 ファイル操作	27
2.9 条件判断	7	3.8 繰り返し (2)	29
2.10 警告・確認画面	9	3.9 条件判断	30
2.11 ラジオボタンとチェックボックス	10	3.10 文字列関数	31
2.12 選択メニュー	12	3.11 送信前のチェック	33
2.13 変数	14	Appendix: Peditor について	35
2.14 タイマー割り込み	15	A.1 peditor のインストール	35
		A.2 peditor の使い方	35

1 はじめに

この「ビジネス情報演習B」の授業では、前期の「ビジネス情報演習A」の続きとして、現在インターネットで使われているWeb関連技術の理解と習得を目指します。よって後期から受講される方には少し大変なところがあるかもしれません。前期・後期分合わせた全体像は次のようになります。

1. ホームページビルダーを利用したWebページの作成：現在ではWebページを作成するための様々なソフトがあり、それらを利用することにより簡単に綺麗なWebページを作成することができます。そしてできたものをWebサーバーに送ることによって世界中に対して情報発信ができます。ただその内容については十分注意する必要があります。(前期前半)
2. HTMLの基礎：Webページの記述はHTMLと言うものに従っています。利用者に応答するWebページを作成するためにはある程度のプログラミングが必要となりますが、それ以前にこれが理解できていないとできません。(前期後半)
3. スタイルシート：HTMLだけでは美しいWebページはできません。Webページの見た目を設定するのがスタイルシートの役割です。ホームページビルダーなどはこれも自動的に利用しています。(前期後半の最後)
4. JavaScript：お買い物のWebページでは、必要な個数などを入力すると単価や消費税や送料を計算して必要な額を示してくれます。これは通常JavaScriptの機能を利用して実現しています。JavaScriptはWebページを表示するブラウザで動くプログラムです。合計金額の計算などの簡単な処理を、Webサーバーの助けなしに実行することができます。(後期前半)
5. PHPによるプログラミング：みなさんが良く使われるWebでの検索は、Webサーバー上でプログラムの実行によって処理されています。通常検索は大規模なデータベースを元に処理を行うので、ブラウザ上のJavaScriptでは不可能です。Webサーバー上で動くプログラムの作成には様々なプログラミング言語を利用することが可能ですが、ここではPHPと呼ばれるものを利用します。(後期後半)

HTML、JavaScript、PHPと段々プログラミングの世界へと進むという感じです。またこれだけやれば、Web関連の技術の大半について概要を知る、実践することができると思います。ただちょっと心配なのは、一つのファイルの中にこの三者が混在するので、混乱する人が出てくるのではないかと言う点です。そのような事がないように同時に三つはやらないつもりですが、御注意ください。

2 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。それ以外の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 を取り上げます。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。

2.1 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られます。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られます。

2.2 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。

複数の HTML のファイルの中で同じ JavaScript の記述を利用したい場合もあります。そのような場合は、スタイルシートと同様に別ファイル(例えば xxx.js)に JavaScript の記述を入れて、それを次のような記述で取り込むことができます。

```
<Script language="JavaScript" src="xxx.js">
</Script>
```

2.3 画面への出力

JavaScriptでブラウザの画面になにか出力したい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「'」で囲みます。JavaScriptではこのような「”」や「'」で囲まれたものを文字列と呼びます。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(物)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>
```

2.4 計算式

JavaScriptでは通常の四則演算に加えて剰余の計算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の形の数を計算しようとするると自動的に数値に変換されて、結果も数値で得られます。

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。(hello.htm)

```
こんにちは
He said, "You are a beautiful woman."
```

7

2.5 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>
```

多くの方が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。

2.6 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直ります。以下は「aaa」と言う名前のフォームの中にある「x」と言う名前の入力欄に入れられた数字を数値として取り出す時の書き方です。

```
eval(document.forms['aaa'].elements['x'].value)
document.forms['aaa'].elements['x'].value*1
```

演習問題

HTMLで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"/>

2.7 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写したのと同じように実行されます。

2.8 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、

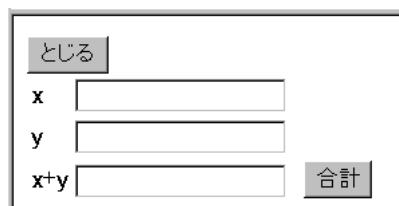
```
<Input type="button" value="ボタンの文字" onClick="関数名 ()">
```

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). Below it are three input fields labeled 'x', 'y', and 'x+y'. To the right of the 'x+y' field is a button labeled '合計' (Total).

3. さらに入力欄を追加し、そこに「平均」のボタンをクリックすると数値の平均が入るようにせよ。

2.9 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。

判断の基本は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。例えば次のような感じになります。

- `x>3 && y<5`

これはxが3より大きくかつ、yが5より小さいという条件を示します。

- `x==y && y==z`

これはxとyとzが等しいことを示します。比較は2つのものしかできないので、`x==y==z`のようには書けません。(書けば別の意味になる。)

このような条件判定を次のようなif文で使用します。

```
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"/>

2.10 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。

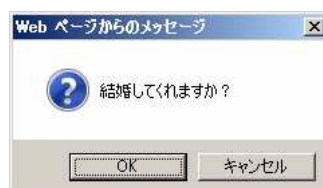


確認画面

間違っってボタンをクリックしたために全てのデータを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、後戻りが簡単にでき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。「==true」は省略可能です。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。そのためには警告画面を出した後で「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。(hantei2.htm)
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。(hantei3.htm)

2.11 ラジオボタンとチェックボックス

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、チェックボックスや次に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、ラジオボタンなど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ことが無く、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。

ラジオボタンとチェックボックスの違いは、複数選択ができるか、できないかにあります。それぞれ用途によって使い分けることになります。例えば、性別を問う場合は、男かつ女は通常ありえないので、複数選択できないラジオボタンを使います。また紅茶に追加するものを選択する場合は、ミルクや砂糖などを好みに合わせて選択することになるので、複数選択できるチェックボックスを使用するのが普通です。

まずラジオボタン自体は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、

```
<Input type="radio" name="sex" value="男"> 男<Br>  
<Input type="radio" name="sex" value="女" checked> 女<Br>  
<Input type="radio" name="sex" value="宇宙人"> その他<Br>
```

のように記述すると、

- 男
- 女
- その他

のようになります。name=のところは、同じグループに属するものは全て同じ名前にします。value=のところは、JavaScriptで選択したものを調べた時に渡したい値を入れます。JavaScriptでは何番目の選択肢を選択したかどうかわかるので、必要としない場合もありますが、授業の後半で扱うPHPによるサーバーでの処理の際には、個々で指定した値しかサーバーに渡らないので必須の項目となります。checkedは同じグループの中で一つだけに付けます。これがある項目が最初に選択された状態になります。これを忘れると、どれも選択されていないままになることがあるので、トラブルの元になります。

そしてInputタグの後に何か書かなければならない事に注意してください。Inputのタグだけでは○しか出てこないのです、何の選択肢なのか利用者にはわかりません。

JavaScript側でどのラジオボタンが選択されたか調べるには次のようにします。

```
if (document.フォーム名.ラジオボタン名[0].checked){  
    1番目のラジオボタンが選択されていた場合の処理内容  
}
```

[]の中がゼロの場合、1番目のラジオボタンを調べることになります。2番目のラジオボタンならば、[]の中は1になります。要するにゼロから数えるようになっています。またvalue=で指定した内容を取り出すには、「document.フォーム名.ラジオボタン名[0].value」のように記述します。

繰り返しを用いて多数のラジオボタンを一気に処理しようとする場合、ラジオボタンがいくつあるかを知る必要があります。そのような場合は、「document.フォーム名.ラジオボタン名.length」でわかります。

チェックボックスもラジオボタンとほぼ同様になります。

```
<Input type="checkbox" name="sugar" checked> 砂糖<Br>  
<Input type="checkbox" name="lemon"> レモン<Br>  
<Input type="checkbox" name="milk"> 牛乳<Br>  
<Input type="checkbox" name="brandy"> ブランデー<Br>
```

のように記述すると、

```
☑ 砂糖  
☐ レモン  
☐ 牛乳  
☐ ブランデー
```

のようになります。グループを構成する必要はないので、name=は異なる名前にします。複数選択可能なので、checkedは複数付けても良いし、全く付けなくても問題ありません。JavaScriptでチェックが付いたかどうか調べる時にはラジオボタンの時と同様に、

```
if (document.フォーム名.チェックボックス名.checked){  
    チェックボックスにチェックが付いている場合の処理内容  
}
```

のようにします。

演習問題

次のようなものをラジオボタンとチェックボックスを利用して作成し、「診断」のボタンをクリックすると、次のようなメッセージが出るようにせよ。ifの数は少ない方がよい。

- 「男」や「その他」を選択した場合は、「まだできていません」というメッセージだけを表示する。
- 「女」を選択した場合には、紅茶に入れるものごとに異なるメッセージを表示する。(例：砂糖⇒甘い性格ですね。)複数選択した場合は、複数のメッセージが出てもよい。

性格診断

あなたの性別は何ですか？

男
 女
 その他

紅茶には何を入れますか？

砂糖
 レモン
 牛乳
 ブランデー

2.12 選択メニュー

ラジオボタンは画面上に予め全ての選択肢が表示されなければならないので、画面上の場所をとると言う問題点があります。ここで説明する選択メニューならば、選択の際にだけ選択肢が表示されるので便利です。ラジオボタンと同様に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」¹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。
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」²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。

```
if (document.フォーム名.セレクト名.options[2].selected){  
    3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」³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()">
```

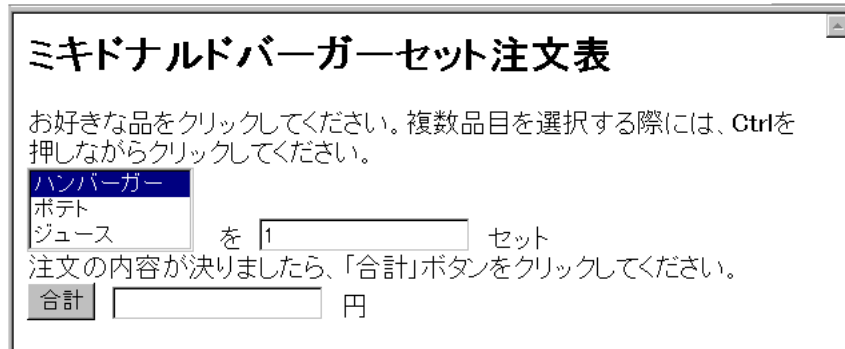
演習問題

1. ミキドナルド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。なおこのバーガーセットは、ハンバーガー(120円)、ポテト(150円)、ジュース(100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)

¹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。

²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。

³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。



2. 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。
3. セット数の部分も選択メニューで1から5まで選択できるようにせよ。

2.13 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ない壁であります。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や文字列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、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は名前の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なります。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。

変数の使い方で判りにくいのは、`=`の両側に同じ名前の変数が出てきた場合です。

```
x=x+3;
```

数学的に考えると x と $x+3$ が等しくなるわけ無いので矛盾していますが、プログラムの世界では、`=`の左側では x という名前の**入れ物**を示し、`=`の右側では x という名前の**入れ物の中身**のことなので矛盾は生じません。ただ両側に同じものが出てくるのは入れるのが面倒？と言った理由から、

```
x+=3;
```

と言った表現が可能になっています。「`+=`」は右側の値を左側の変数に加える、というような働きをします。

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)



```
x=    
x2+3x+4= 
```

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)



```
カウント:  
```

2.14 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、関数実行の機会に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);
```

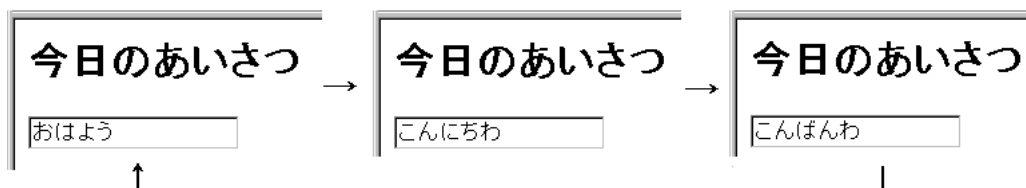
「関数名」のところには時間が経過したときに呼び出す関数を”owari()”のような感じで記述します。「時間」のところには、関数を呼び出すまで待つ時間を1/1000秒単位で記述します。たとえば3秒後に呼び出すならば3000と書きます。ただしこの時間に関しては様々な要因が絡むのでそれほど正確ではありません。「x」は任意の変数で構いませんが、これにタイマーの識別子(ID)が入ります。これを使用して次のようにすると設定したタイマー割り込みを解除することができます。

```
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1回だけ出す。
 - 「yes」をクリックした場合には、「●●●」と言う警告画面を出す。(●●●の部分は何か適当なものを自分で考える。)
 - 「no」をクリックした場合には、「○○○」と言う警告画面を出す。(○○○の部分は何か適当なものを自分で考える。)

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)

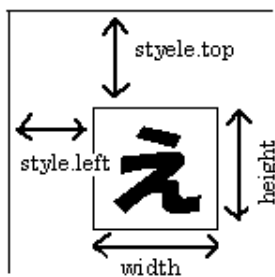
難しい問題

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？

2.15 画像のプロパティ

JavaScriptでは様々なものをオブジェクト (object: 物) として扱います。windowやdocumentもオブジェクトです。オブジェクトはプロパティ(property: 特質)を持ちます。どのようなプロパティを持つかはオブジェクト次第です。ここでは画像のプロパティを例にあげ、それをJavaScriptで扱う方法を説明します。

まず画像のプロパティとして代表的なものに次のようなものがあります。



height	画像の縦幅を画素数で表したもの
width	画像の横幅を画素数で表したもの
style.top	表示している画像の左上角のY座標
style.left	表示している画像の左上角のX座標
name	画像の名前
nameProp	表示している画像のファイル名
src	表示している画像のURL
onclick	画像をクリックしたときに起動される関数

これらのプロパティはHTMLのタグの一部として設定することができます。またstyle.topやstyle.leftはStyle Sheetで設定することができます。

```

```

またJavaScriptでプロパティを変更することにより変わる画像を実現することができます。「position: absolute」という指定が出てきますが、これは位置をWindow内の座標で指定するという意味です。

```
<HTML>
<Head>
<Title>動く・変わる画像</Title>
</Head>

<Body>

<script language="JavaScript">
<!--
window.alert("表示されている画像の縦幅は、"+document.kousha.height+"です。");
window.alert("表示されている画像のURLは、"+document.kousha.src+"です。");
window.alert("表示されている画像のファイル名は、"+document.kousha.nameProp+"です。");
window.alert("表示されている画像のY座標は、"+document.kousha.style.top+"です。");
window.alert("表示されている画像を動かします。");
document.kousha.style.top="50px";
window.alert("表示されている画像を変更します。");
document.kousha.src="http://www.ss.sugiyama-u.ac.jp/web/gif/seis_s.gif";
//-->
</script>
</Body>
</HTML>
```

画像に名前が付いていない場合や他の名前と混同の恐れがある場合は次のように指定します。

```
document.images['kousha'].src="test.gif"; // 混同を避ける
document.images[0].src="test.gif"; // ページに最初に出てきた画像
```

演習問題

1. 「→」と書かれたボタンをクリックすると画像が右に、「←」と書かれたボタンをクリックすると画像が左に動くものを作れ。(image1.htm)
2. 「Start」と書かれたボタンをクリックすると画像が1秒毎に左右に動き、「End」のボタンをクリックすると停止するものを作れ。(image2.htm)
3. 「Start」と書かれたボタンをクリックすると画像が1秒毎に左右に動き、また0.7秒毎に上下にも動くようする。そして「End」のボタンをクリックすると停止するものを作れ。(image3.htm)
4. 上記のものに、「End」のボタンをクリックしたときに左上であれば「あたり！」と警告画面が表示されるようにせよ。(image4.htm)

2.16 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うものもあります。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。次のようにして新しい Window を開くことができます。

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

URL には通常、“http://...”とか“ファイル名”が入ります。windowID は window に付ける名前でも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 A タグの中の 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。)以下に option に使用できるものを示します。

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

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でしょう。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、そのwindowが既にあることもありません。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(URLの場所に""を指定)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("<HTML>");  
w.document.write("<Head><Title>タイトル</Title></Head>");  
w.document.write("<Body><H1>タイトルだぜ</H1></Body>");  
w.document.write("</HTML>");
```

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⁴。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。例えば子のWindowの方で次のような文を実行すると、元のWindowに楢山のトップページを出す事ができます。

```
window.opener.location.href="http://www.sugiyama-u.ac.jp/";
```

そしてJavaScriptが記述されているWindowに対して操作したい場合は、「window.」を付けるか何も付けないようにします。

演習問題

1. 「Open Top」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

⁴document.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の記述が終了したと見なされて変なエラーになります。

うにせよ。さらに元の Window にある「学園センター」と言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。(window1.htm)

2. 「Open Close」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。(window2.htm)
3. ブラウザでファイルを開くと、元の Window には梶山のトップページが、別 Window には自分の好きな画像が表示されるようにせよ。別 Window の大きさは表示される画像に合わせることに。(window3.htm)
4. 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」と言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 に適当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)

3 PHP 入門

前章では JavaScript を利用して、利用者の入力に応答する Web ページを作成する方法について学びました。しかしさらに利用者に対して高度な応答を行うためには、ブラウザ側だけでは無理があります。例えば Yahoo などのキーワード検索を行おうとしても、あらかじめ検索のデータベースの内容を全てブラウザに送ることは無理でしょう。また注文の受付のページでは入力された注文内容をサーバーの方で記録し、発注の処理に入らなければなりません、お客のブラウザからではそのようなことができません。

この章で扱う PHP は、1994 年に Rasmus Lerdorf 氏によって開発が始まったスクリプト言語⁵です。もともとは「Personal Home Page」から名づけられたようですが、現在は個人の手を離れて「PHP: Hypertext Preprocessor」として多くの人々の手によって開発が続けられています。

PHP の特徴の一つは、サーバー側での処理を通常の HTML のファイルの中に混在させることができる点です。サーバーで処理した結果をブラウザで表示するためには、これまでどおり HTML による記述が必要です。そしてそのような部分はこれまでどおり記述し、サーバーからの応答の部分だけ PHP の記述するような形になります。慣れれば大変便利ですが、もしかすると HTML、JavaScript、PHP がごちゃごちゃになってしまう人が居るかもしれないのが、少し心配なところです。

PHP の特徴として他に良く上げられるのは、様々なデータベースソフトと簡単に接続できる点です。さきほど例に挙げたキーワード検索や注文受付の処理では、通常データベースが利用されます。検索ならばあらかじめデータをデータベースに入れておき、その中から探します。また注文受付も、受け付けた注文の内容をデータベースに登録するのが普通です。このように大抵の処理ではデータベースの利用が必須となっていますが、PHP から簡単にデータベースを利用できるので、大抵の処理が簡単に記述できるということになります。

Web サーバーで利用できるプログラミング言語として以前は Perl などが良く使われて来ましたが、現在では多くのシステムが PHP で構築されています。

3.1 どこに・どのように入れるのか

PHP 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含めます。入れる場所は特に決まっていません。記述された内容は基本的にはファイルの最初の方に記述された方から実行されます。そして実行された結果がブラウザに送られます。ブラウザ側で「ソースの表示」を行っても PHP の記述は残っていません。

これまでの HTML の記述と PHP の記述が混ざらないように、PHP の記述は `<?php` と `?>` の間に入れます。つまり HTML 的には「`?php`」というタグの形をしています。終わりのタグではありませんから、`/?>` のように「`/`」を入れないでください。

```
<?php
  PHP の記述
?>
```

PHP を利用する際の重要な注意は、

- 拡張子は「.htm」ではなく「.php」にする。

⁵簡易プログラミング言語とも呼ばれるもの。通常実行のための変換を必要とせず、すぐにコンピュータで実行できる。

- 動作を確認する際は必ずサーバーに送らなければならない。

と言う点です。拡張子が従来どおりの「.htm」では、PHPの記述は単なる変なタグと言うことでブラウザで無視されます。またサーバーがPHPの実行を行うので、これまでのようにTeraPadのボタンで表示させたり、パソコンのデスクトップに保存したアイコンをダブルクリックしたのではだめです。

サーバーへのファイル送り方は前期のテキストにあります。演習室メニューにあるmftpを起動します。そして開いたwindowの中の「www」という名前の付いたフォルダーをダブルクリックして開いておきます。できたPHPを含むファイルは、ここにドラッグすることによりサーバーに送ることができます。mftpは授業中は閉じずに、ずっと開いたままにしておく事を勧めます。

そして送ったファイルの動作を確認するためには、ブラウザを起動し自分のトップページのURL⁶の後にサーバーへ送ったファイル名を追加してアクセスします。

3.2 表示の命令

まずはJavaScriptのdocument.writeに相当するものです。これを利用して後述の変数の内容を表示するだけでなく、通常のメッセージなどをHTMLのタグを含めた形で表示することもできます。かつてJavaScriptの演習問題で行ったのと同じものを例として示します。

```
<HTML>
<Head>
<Title>PHPの例</Title>
</Head>

<Body>
<?php
    echo "こんにちは<Br>";
    echo 'He said, "I love you."<Br>';
    echo '<Font size="7">',3+4,"</Font>";
?>
</Body>
</HTML>
```

これを「test.php」のような拡張子の付いたファイル名で一旦パソコンに保存し、サーバーに送って見るとJavaScriptの場合と同様に表示されます。TeraPadで「.php」と言う拡張子のファイルを保存する際には、「ファイルの種類」のところを必ず「すべてのファイル(*.*)」にして下さい⁷。

この例からわかるように何らかの表示を行いたいときは、PHPでは「echo」と言う命令を使います。JavaScriptのdocument.writeは関数だったので()の間に表示したいものを入れましたが、こちらは命令なので()は不要です⁸。また区切りの「;」は必ず必要なので忘れないようにしてください。

PHPで何か間違えるとエラーのメッセージが表示されます。例えば、先ほどの例で9行目にある二つ目のechoの行の最後の「;」を忘れると、「Parse error: parse error, unexpected T_ECHO, expecting ',' or ';' intest.php on line 10」のようなメッセージが表示

⁶<http://www.center.sugiyama-u.ac.jp/users/学籍番号/>

⁷ そうしないと、test.phpがtest.php.txtのようになります。

⁸ PHPにも関数があります。PHPの関数を用いる場合は、JavaScriptと同様に()の中に入れることになります。

されます。10行目で「,」か「;」を期待していたのに、お呼びでない「echo」があったせ、と言うような意味です。このように行の最後で間違えるとエラーの場所が次の行として表示されることもあります⁹。

JavaScriptでは「~」と「`」は全く同じ意味でしたが、PHPでは少し違います。次に出てくる変数を「~」の部分に含んだとき、前者では変数の部分の中身と置き換えられますが、後者ではそのようなことはありません。

```
$x=123;  
echo "x=$x<Br>"; // x=123 と表示される。  
echo 'x=$x<Br>'; // x=$x と表示される。
```

3.3 変数・計算式

PHPでは変数を宣言せずに使用することができます。ただそれではどれが変数なのか見分けが付きにくくなるので、「\$x」のように必ず先頭に「\$」を付けます。また変数名には残念ながら漢字は使用できません。英数字¹⁰にしてください。

変数が「=」の左側に出てきた場合は、変数に何か入れようとしていることを示しています。変数には数値や文字列などを入れることができます。その他の場合は変数に入っている内容を示しています。「123」と言うような文字列は計算式の中にあれば数値の123として扱われます。また必要があれば数値が文字列に自動的に変換されることもあります。

```
$x="123"+456;  
echo "x=", $x; // x=579 と表示される。
```

計算式はJavaScriptと同様に記述することができます。ただ「+」は本当に加算の意味でしかありません。文字列と文字列をくっつけたいときは「.»を使用します。

```
$yen=1234;  
$kekka="お値段は".$yen."円です。<Br>";  
echo $kekka; // お値段は1234円です。 と表示される。
```

3.4 繰り返し (1)

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PHPではfor文と呼ばれる次のような記述で「I love you.»を100回表示することができます。

⁹「}」を忘れると、はるかかなたでエラーになることもあります。

¹⁰英字または数字で、先頭は英字にしてください。

```
for ($i=0;$i<100;$i++){
    echo $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¹¹。PHPではfor文は、for(**A**;**B**;**C**) {**D**}のようなちよつ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. 「echo」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. 「echo」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。

¹¹既に扱ったJavaScriptでももちろん使用可能です。

演習問題

1. 画面上に「1 3 5 7 9 ... 97 99」と表示するもの。数字ごとに改行する必要はない。(kisuu.php)
2. 画面上に「99 98 97 ... 3 2 1 ドカン!」を全てPHPで表示するもの。数字ごとに改行する必要はない。(countdown.php)
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。for文を二つ使用する。(10x10.php)

```
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
```

3.5 配列型変数

コンピュータで大量の情報を処理したい場合、それに応じて変数も大量に必要になります。そのような場合にそれぞれの変数に別の名前を与えるのは大変です。そこで配列型の変数と言うものがプログラミング言語では使えるようになってきました。PHPでは配列型の変数も変数ですので「\$」から始まります。そして変数の名前の後に「[]」がありその中に添え字と呼ばれるものを指定します。普通のプログラミング言語では添え字は数値ですが、PHPでは文字列も使用できます。

```
$data[1]=123;$data[2]=456;$data[3]=789;
for ($i=1;$i<=3;$i++) {
    echo "$data[$i]<Br>";
}
$data['name']="三木";$data['address']="名古屋市";
echo $data['name'],"の住まいは",$data['address'],"です。";
```

3.6 フォームからの入力

フォームの入力欄に利用者が入力した内容をPHPで処理するためには、一旦サーバーに入力した内容を送り返してもらわなければなりません。そのためにはクリックしたらフォームの内容を送信するボタンが必要になります。このボタンはHTMLで次のように記述します。

```
<input type="submit" value="送信">
```

valueで指定した内容はボタンの上にかかれる文字ですので任意のものが可能ですが、利用者が最後にこれをクリックしなければならないことがわかるようなものにします。またこれだけでは

サーバーに送った内容をどのPHPのファイルが処理をすれば良いのかわかりません。この処理するPHPのファイルの指定は、Formのタグのところで行います。

```
<Form method="POST" action="prog.php">
```

methodについてはPOST以外にGETも指定可能ですが、GETの方はサーバーに送れるデータ量に制限があるのでとりあえずはPOSTで良いでしょう。actionでサーバーで処理するプログラムを指定します。

指定されたPHPのファイルの方では、特別な配列型変数を用いることにより、簡単にフォームの中で入力した内容を取り出すことができます。例えば「namae」と言う名前を付けた入力欄の値は、「\$_POST['namae']」で取り出すことができます。

```
<HTML>
<Head>
<Title>入力欄の例</Title>
</Head>

<Body>
<Form method="POST" action="prog.php">
名前 : <Input name="namae"><Br>
<Input type="submit" value="送信">
</Form>
</Body>
</HTML>
```

この入力欄の例にはPHPの記述は何も含まれて居ませんので、通常の拡張子が.htmのファイルに入れます。(例えば「input.htm」)そして次のは必ず「prog.php」と名前を付けて保存してください。(上記の例でaction=でこの名前を指定しているので。)

```
<HTML>
<Head>
<Title>入力欄処理の例</Title>
</Head>

<Body>
<?php
    echo "名前には、",$_POST['namae'],"が入っていました。<Br>";
?>
</Body>
</HTML>
```

演習問題

次のような入力画面になるページ(cyuumon.htm)を作成し、そこから注文内容を表示する(uketuke.php)を作成せよ。なお入力画面の入力欄には上から順番に、「namae」、「bangou」、「tanka」、「kosuu」という名前がついている。

通信販売注文受付

ご注文内容を入力してから、「注文」のボタンをクリックしてください。

- お名前:
- 商品番号:
- 単価:
- 個数:

通信販売注文内容

美希様 ご注文ありがとうございました。
商品番号:123456の商品を30個お届けします。
金額は4500円(税抜き)です。

3.7 ファイル操作

利用者からの入力を保存するためにはファイルを作成して、そこに入力内容を入れなければなりません。また逆にファイルに入っている情報を取り出して表示するようなこともしばしば行われます。ここではそのようなファイルに関する操作をどのようにPHPでは記述するのかについて説明します。

ファイルを開く

ファイルを利用するためにはまず「ファイルを開く」という操作が必要になります。このとき、これから扱うファイルの名前とファイルに対してどのような操作を行いたいかを指定します。

```
$file=fopen("aaa.txt","w");
```

fopenが「ファイルを開く」関数です。fopenの最初に指定しているのがこれから操作するファイルの名前です。この例では「aaa.txt」という名前のファイルが対象となります。次に“w”という指定がありますが、これはファイルに情報を書き込む(write)ことを示します。もし「aaa.txt」がなければ、ここで「aaa.txt」と言う空のファイルが作成されます。もし既に存在した場合は、これまで入っていた内容は消されてしまうので注意します。

これまで入っていた内容に追加(append)をしたい場合は、“w”の代わりに“a”を指定します。ファイルがなかった場合は“w”を指定したのと同じ動作になります¹²。

ファイルの内容を読み出し(read)をしたい場合は、“r”を指定します。もし存在しないファイルに対してこの指定をするとfopenは「false」という特殊な値を返し、以下の読み書きはできませんのでご注意ください。

¹²PHP本来の機能ではこれで説明終わりなのですが、実際に使用してみると「Permission denied」というエラーが生じることがあります。これは許可がないと言う意味で、WebサーバーのソフトがPHPを動かしてファイルを作成しようとしたところ、利用者の領域だったのでファイルを作成できなかったためです。そのような事が生じないようにsuPHPと言うソフトでこの問題を回避するように設定してあります。

fopen の返す値をこの例では \$file という変数に入れています。これは後で使用しますので、適当な名前の変数に必ず入れてください。なお、同時に複数のファイルを扱うことができますが、その場合は fopen の結果をそれぞれ異なる変数に入れてください。

ファイルへの書き込み

ファイルへ何かデータを書き込む場合は、次のように fwrite を使用します。

```
fwrite($file,"書き込むデータ¥n");
```

この例では先ほど fopen で開いたファイルに「書き込むデータ」という文字列と改行 (¥n) が書き込まれます。echo と同様に変数を指定すれば変数の内容を書き込むこともできます。fwrite の () の中には一つしか書き込みたい内容を指定できないので、文字列と変数の内容を両方書きたい場合は、文字列の結合を示す「.」でつなぎます。書き込みたい内容が多数ある場合は fwrite を必要な数だけ繰り返し記述します。複数のデータを書き込む場合は、データとデータの間空白や改行が入るようにします。例えば「123」と「456」を続けてファイルに書き込むとファイル中には「123456」が残るので、後でこれを読み込んでどこで区切れれば良いのかわからなくなります。次のようにファイルに書き込むと、

```
$a=123;
fwrite($file,$a."¥n");
fwrite($file,"ありがとう¥n");
```

ファイルには次のような内容が入ります。

```
123
ありがとう
```

ファイルからの読み出し

ファイルからデータを読み出すには、次のように fgets を用います。

```
$data=fgets($file,256);
```

これで先ほどの fopen で開いたファイルより 1 行分読み出され、変数 \$data に入ります。256 はデータの最大長です。もしファイルにこれより長い行があった場合は、とりあえずここで指定した分だけ変数に入ります。この fgets を繰り返すことによりファイルの先頭から順番に 1 行ずつ読み出すことができます。

ファイルを閉じる

ファイルに関する操作が終わったら次のようにしてファイルを閉じます。ファイルの読み出しの際は大きな問題になることは少ないですが、書き込みの際に忘れると色々問題を生じることがあります。

```
fclose($file);
```

ファイルの内容の確認

editor でファイルを編集することにより、ファイルに書き込んだ内容を確認することができます。また mFtp では、内容を確認したいファイルのアイコンをダブルクリックすると、そのファイルがパソコンのデスクトップ上にサーバーから送られ、パソコンにあるソフトウェアによって開かれます。内容を確認した後、ソフトウェアを閉じて、デスクトップに送られたファイルはそのまま残りますので不要な場合は削除してください。また内容を修正した場合は、サーバーに送り返さないと、サーバ上のファイルの内容は変更されませんのでご注意ください。

なお、拡張子が txt のファイルは通常「メモ帳」で開かれますが、改行が無視されて1行になったり、漢字が文字化けすることもあります。これは、サーバーとパソコンでは、改行コードや文字コードが異なっているためです。TeraPad はこの辺りの違いを自動的に検出して対応してくれるので、デスクトップにある表示が変になったファイルを、TeraPad で開きなすと大抵正しく表示されます。

演習問題

前節の演習問題で作成した注文内容を表示する (uketuke.php) に各入力内容を別々の行にしてファイル (data.txt) に追加する部分を追加せよ。なお、ファイルに入れる順番は、名前、商品番号、単価、個数の順とする。

3.8 繰り返し (2)

for 文以外にも while 文と言われ繰り返し文があります。

```
while (条件) {  
    条件が成立している間に繰り返し実行する内容  
}
```

for 文でも時々終了しない繰り返しになって困りますが、while 文は繰り返される内容の中で条件が変化するようなことをしないと簡単に終わらないものになってしまうので注意します。また、do-while 文と呼ばれる繰り返し文もあり、次のような形をしています。

```
do {  
    条件が成立している間に繰り返し実行する内容  
} while (条件);
```

while 文と異なるのは条件を調べるのが実行をしてからと言う点です。つまり繰り返し実行される内容は少なくとも一回は実行されます。

あらかじめ何行分のデータがファイルにあるとわかっている場合は、その数だけ fgets を書けば良いのですが、実際はどのくらい入っているかわからない例も多数あります。そういう場合はファイルから読み出しができなかった場合に fgets が「false」という値を返すのを利用して次のような while による繰り返しを使います。

```
while ($data=fgets($file,256)) {  
    $dataに読み出されたデータの処理  
}
```

演習問題

1. for 文の例を while 文を利用して記述せよ。
2. data.txt ファイルの内容を全て次のように表の形で表示するものを作成せよ。ただしデータが何件入っているかは事前にはわからないものとする。(hyouji.php)



お名前	商品番号	単価	個数
美希ちゃん	123456	150	30
りかちゃん	aa123	1000	10
りさちゃん	pp7878	300	20

3.9 条件判断

PHP の条件判断も JavaScript と全く同じ形をしています。() 中の条件の書き方も同様です。例えば「変数 a の内容が 10 に等しくない」は「`$a!=10`」のように書きます。

```
if (条件) {  
    条件が成立したときの内容  
} else {  
    条件が成立しなかったときの内容  
}
```

条件が不成立の場合の内容がないときは、else 以下は省略できます。PHP が JavaScript と大きく異なるところは内容のところに HTML などの内容を直接記述する方法があることです。例えば次のような感じです。

```
if ($tokuten==100) { // 得点の変数($tokuten)が100ならば画像を表示する。  
?>  
      
<?php  
}
```

つまり一度?>でPHPの記述を終わらせて通常のHTMLのタグなどを記述することができます¹³。ただ閉じる「}」などを忘れるとエラーになります。

¹³JavaScriptではこのような場合、document.writeを使用する必要がありました。

演習問題

1. 前章の演習問題2番 (hyouji.php) に追加をして、注文件数が表の下に「現在の注文件数：3件」のように表示されるようにせよ。なお件数がゼロの場合は、「注文はありません」と表示されるようにする。

ヒント：繰り返しに入る前に「`$count=0;`」をし、繰り返しの中で「`$count++;`」をしておけば、繰り返しが終了したときに`$count`に件数が入っている。

2. 前章の演習問題2番 (hyouji.php) に追加をして、注文金額(単価と個数の積)が10,000以上の注文者の名前が赤で表示されるようにせよ。前章の演習問題2番の画面例ならば、「りちゃん」のみが赤く表示される。

3.10 文字列関数

コンピュータは、数値の計算をするから電子計算機と呼ばれたのですが、現在の使われ方は計算以外の仕事の方が多いようにも思われます。数値以外のデータとして例えば文字列があります。ここではPHPで文字列を扱う際に使われる関数のいくつかを紹介します。

- `strlen`関数：文字列の長さを求める関数です。文字列の長さは、文字列を構成する文字の数です。

```
echo strlen("abcdefg"); // 7と表示される
```

- `strpos`関数：文字列の中に、指定した文字列が含まれるかどうか調べる関数です。最初に指定した文字列の中に、次に指定した文字列が含まれれば、何文字目からかを答えます。もし含まれて居なければ`false`を返します。

```
echo strpos("abcdefg","cde"); // 2と表示される
```

実は先頭をゼロと数えるので上記の例では3ではなく2が表示されます。PHPではゼロと`false`は同等に扱われます。そのために含まれなければ何かすると言う場合は次のような書き方をします。

```
if (strpos($x,"abc")===false) {  
    // $xにabcが含まれなかった場合の処理  
}
```

等号が2つでも妙な感じでしょうが、等号が3つ必要となります。これの否定の条件では「`!==`」になります。

- `substr`関数：文字列の一部を切り出す関数です。何文字目から切り出すのかと、何文字分切り出すかを指定します。何文字分の指定が省略された場合は、指定された文字以降全てが切り出されます。

```
echo substr("abcdefg",2,3); // cde が表示される
echo substr("abcdefg",4); // efg が表示される
```

何文字目かの指定の際も先頭の文字がゼロなので注意します。また何文字目かのところを負の数にすると、文字列の後ろから指定することもできます。

```
echo substr("abcdefg",-3,2); // ef が表示される
```

また以下の操作は既にやりました。

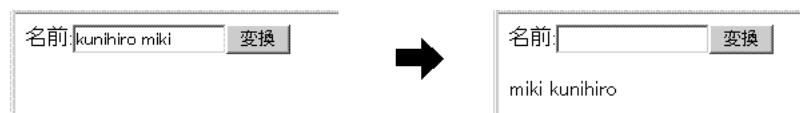
- 文字列の連結：文字列と文字列を連結したい時には「.」を使います。
- 文字列の比較：二つの文字列が等しいかどうか調べる時は「==」、等しくないかどうかを調べる時は「!=」を使用します。アルファベット順で前にあるものが、より小さいと判断されるので、「<」や「>」も使用可能です。

なお、処理する対象の文字列が、半角の英数字だけでなく、日本語の漢字などを含む場合はそういう文字に対応できる関数を用います。例えば「strlen("あいう")」の結果は6になります。一方以下の表にある「mb_strlen("あいう","SJIS")」ならば結果は3になります。正しい結果を求めるためには、正しいencodingの設定(前述の例では"SJIS"の部分)を行わないとならないので、日本語限定でも結構面倒です。

内容	半角英数字用	日本語にも対応
文字列の長さ	strlen	mb_strlen
文字列の位置	strpos	mb_strpos
文字列の切り出し	substr	mb_substr

演習問題

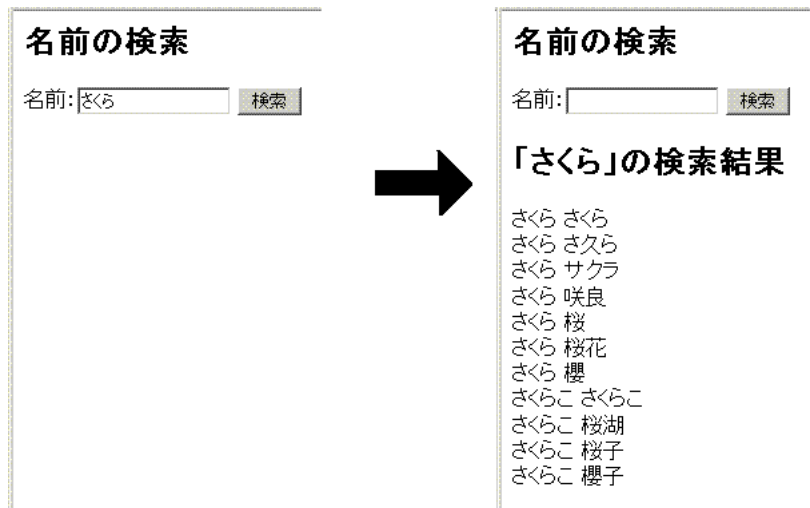
1. 欧米では姓名を日本の逆順にする。入力欄に入力された姓名(ただし姓と名の間には空白が一つあるものとする)を「変換」ボタンをクリックすると逆順に表示するプログラムを作成せよ。



2. 次のような名前の検索プログラムを作成せよ。

- ファイルは kensaku.php のみである。
- 検索するデータは fname.txt というファイルに入っている¹⁴。入力されたキーワードを含む行をそのまま画面に表示すれば良い。

¹⁴このファイルは教員から提供するので、各自 mftp または peditor で web サーバーに転送すること。



3.11 送信前のチェック

人は誰でも間違いを犯します。Web ページの入力に誤りがあった場合、その誤りが機械的に検出可能であっても、PHP は送信ボタンをクリックして入力内容がサーバーに来ない限り対応できません。そして誤りがあったことをまた送り返さないと利用者には伝わりません。ネットワークの往復をする代わりに、送信ボタンをクリックすると、まず JavaScript でチェックを行い、問題が無ければ本当にサーバーに送信することが考えられます。ネットワークを使用しない分即座に応答を返すことができます。

そのためには、色々手直しの必要があります。元は次のようになっていたとします。

```
<Form method="POST" action="syori.php">  
... 中略...  
<Input type="submit" value="送信">  
... 後略...
```

Form のタグには、「name="bbb"」のように名前を付けます。これは後で JavaScript からこのフォームに対して送信の指示ができるようにするためです。また、「type="submit"」のボタンを「type="button"」に変更し、クリックしても即座に送信されないようにします。さらに onClick を利用して内容をチェックする関数を指定します。そうすると次のような感じになります。

```
<Form method="POST" action="syori.php" name="bbb">  
... 中略...  
<Input type="button" value="送信" onClick="check()">  
... 後略...
```

ここでは内容をチェックする関数として check() を指定していますが、もちろん check() の内容も定義しなければなりません。入力欄などの内容を調べて、何か問題などがあれば、そのことを警告画面などを使用して利用者に知らせてから return 分を実行するか、関数を終了します。入力の内容に問題がなければ、次のようにしてフォームの submit() 関数を呼び出すことによって、入力された内容をサーバーへ送ることができます。

```
function check(){
  ... 中略...
  document.bbb.submit(); // bbbはフォームにつけた名前
  ... 後略...
}
```

演習問題

1. p.32の「検索プログラム」を、「検索」ボタンをクリックすると、入力内容が空でなければ送信するようにせよ。もし入力内容が空であれば、警告画面を用いてその旨を知らせるようにする。
2. p.26の演習問題のcyuumon.htmにおいて、「注文」ボタンをクリックされると、確認画面が表示され、そこで「OK」ボタンをクリックすると送信されるようにせよ。なお、確認画面には入力された単価と個数から金額を計算し、「～円になりますが、注文処理をしても良いですか?」と表示されるようにする。

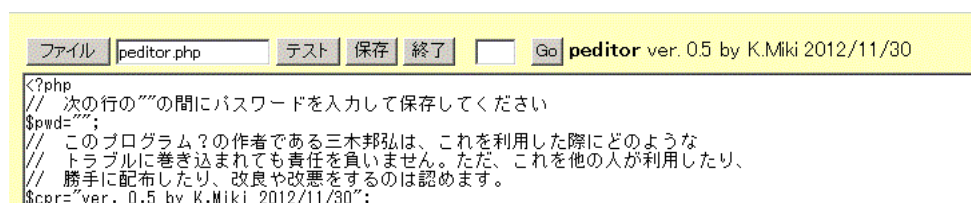
Appendix: Peditor について

PHP を含んだ Web ページのファイルはテストが面倒です。いちいち修正したファイルをサーバーへ送らなければなりません。そこで、サーバー上でファイルを編集できるようにしたものが peditor です。これを使用すれば修正したファイルはサーバー上にあるので、すぐにテストが可能になります。

A.1 peditor のインストール

peditor のホームページ (<http://www.mgt.sugiyama-u.ac.jp/miki/Peditor/>) から peditor.php の最新版をダウンロードして、これを自分のホームページにアップロードするだけです。アップロードには mftp を使用すればよいでしょう。mftp を起動し、自分の ID とパスワードを入れ、「www」フォルダを開き、そこへ peditor.php をドラッグするとインストール終了です。逆に peditor が不要になれば、mftp など peditor.php を削除します。

ブラウザで peditor.php に初めてアクセス¹⁵すると「パスワードが設定されていません」とダイアログが表示されて、次のような編集画面になり、peditor.php 自身が編集できる状態になります。そこで 3 行目の、「\$pwd=""」の「""」の間にパスワードを入れます。それ以外のところは変更しないように、「”」は消さないように注意します。



```
<?php
// 次の行の""の間にパスワードを入力して保存してください
$password="";
// このプログラム?の作者である三木邦弘は、これを利用した際にどのような
// トラブルに巻き込まれても責任を負いません。ただ、これを他の人が利用したり、
// 勝手に配布したり、改良や改悪するのは認めず。
$cpr="ver. 0.5 by K.Miki 2012/11/30";
```

パスワードを入れたら「保存」ボタンをクリックします。するともう一度「パスワードが設定されていません」とダイアログが表示されてから、「ファイルを保存しました」のダイアログが表示されて、パスワード入力画面になります。以上でインストールは終わりです。

もし自分で設定したパスワードを忘れた場合は、もう一度インストールしてください。パスワードを変更したくなったら、peditor で peditor.php を編集してください。

A.2 peditor の使い方

アクセスすると、次のようなパスワード入力画面になりますので、パスワードを入力した上で「Start」ボタンをクリックします。



パスワード: Start

すると次のようなファイル一覧画面になります。編集画面からこの画面に行くときは、「ファイル」ボタンをクリックします。

¹⁵ブラウザで <http://www.center.sugiyama-u.ac.jp/users/学籍番号/peditor.php> を指定する。



- 新規作成の場合：「新規作成(ファイル)」ボタンの左側の入力欄にファイル名を入力し、「新規作成(ファイル)」ボタンをクリックします。編集画面に変わるので、内容を入力してから「保存」ボタンをクリックします。
- 既存のファイルを編集する場合：該当するファイルの「編集」ボタンをクリックします。「保存」ボタンをクリックする前にファイル名を変更すれば、元のファイルを以前のまま残しておくこと¹⁶ができます。

TeraPadと違い、全角の空白を□で表示する機能がありません。全角の空白はよくPHPなどのエラーの原因になるのですが、通常の空白と見た目が変わらないので発見に苦労します。入力の際にうっかり入れないように注意しましょう。

保存した後で「テスト」ボタンをクリックすると、ブラウザの別の画面¹⁷で編集したファイルを確認することができます。PHPなどでエラーが発生した場合、何行目かわかれば、画面の一番上の行の「Go」のボタンの左側の小さな入力欄に行数を入れてから「Go」ボタンをクリックすると、カーソルがその行の先頭に移動します。

さらに次のようなことがpeditorでできます。

- 別のウィンドウを開き、そちらでも作業ができるようになります。(「別Windowを開く」ボタン)ただし、別のウィンドウの方で「終了」ボタンをクリックすると元のウィンドウの方も終了してしまうので、閉じたいときはブラウザの×ボタンを使用してください。
- ファイルやディレクトリ(Winodwsでフォルダと呼ばれるもの)の名前の変更ができます。(「名前変更」ボタン)
- ファイルやディレクトリのモード(読み書きの許可)の変更ができます。(「モード変更」ボタン)
- ファイルやディレクトリを作成することができます。「新規作成(ファイル)」ボタンの左にある入力欄に、新しいファイル名やディレクトリ名を入力してから、「新規作成(ファイル)」ボタンまたは「新規作成(ディレクトリ)」ボタンをクリックします。
- ファイルやディレクトリの削除ができます。(「削除」ボタン)

¹⁶ 既存のファイルを編集して保存すると、元のファイルは拡張子をbakに変更して残されます。ただこれで残るのは一つ前の状態だけなので、あまり当てになりません。また、編集前のファイルに用がなければ、~.bakと言う名前のファイルは削除しても構いません。

¹⁷ ブラウザの設定によっては別のタブに表示されることもあります。

- サブディレクトリの中を見ることができます。(「開く」ボタン)親ディレクトリに戻る時は、「..」と言うディレクトリの「開く」ボタンをクリックします。
- パソコンで作成したファイルをサーバーに送ることができます。「参照」ボタンをクリックし、出てきた「ファイルのアップロード」のウィンドウで送りたいファイルを選択して「開く」ボタンをクリックします。そして「ファイルをアップロード」ボタンをクリックします。するとファイルがサーバーに送られて、ファイルの一覧の中に追加されます。

なお、peditorの使用終了時には、必ず「終了」のボタンをクリックしてください。