

HTML ファイル上から easyBook の指定ページを開く方法について (Safari 対応版)

株式会社光陽社
easyBook サポートセンター

はじめに

easyBook では、リンク元の HTML ファイル上から、easyBook 上の指定されたページを直接開く機能を持ち合わせていません。しかしながら、HTML、JavaScript のコーディングを工夫することで、この機能を実現することが可能になります。本資料では、Internet Explorer、Safari での閲覧に対応した指定のページを開く方法について、3つの方法を紹介いたします。

次ページより、以下の 3つの章に分けて設定方法を解説していきます。

- I. 操作メニューと easyBook を別ウィンドウで開き操作する方法
- II. フレーム内にメニュー表示エリアと easyBook 表示エリアを設けて操作する方法
- III. リンク元の HTML 上から直接 easyBook の指定のページを開く操作方法

本資料で説明する方法は、別途配布する JavaScript ファイルを HTML ファイル上に読み込み、そこに定義されている関数を呼び出す形で、指定したページを開く方法です。この JavaScript ファイル内に定義されている関数を、呼び出し元の HTML ファイル内に直接記述し、ご利用いただいてもかまいません。本資料の末尾に【資料】として、それぞれの HTML、JavaScript ファイルのリストも掲載しておりますので、併せてご覧下さい。

また、動作の確認、組み込みの参考にしていただけるように、この方法で作成したサンプルファイルもダウンロードできるようにしております。次ページからの説明で記載しているファイル名は、このサンプルファイルで使用されているファイル名を掲載しています。組み込み時の参考にしていただければと思います。

なお、各 HTML のタグや JavaScript の詳しい記述方法については、市販の解説書等を参照して下さい。

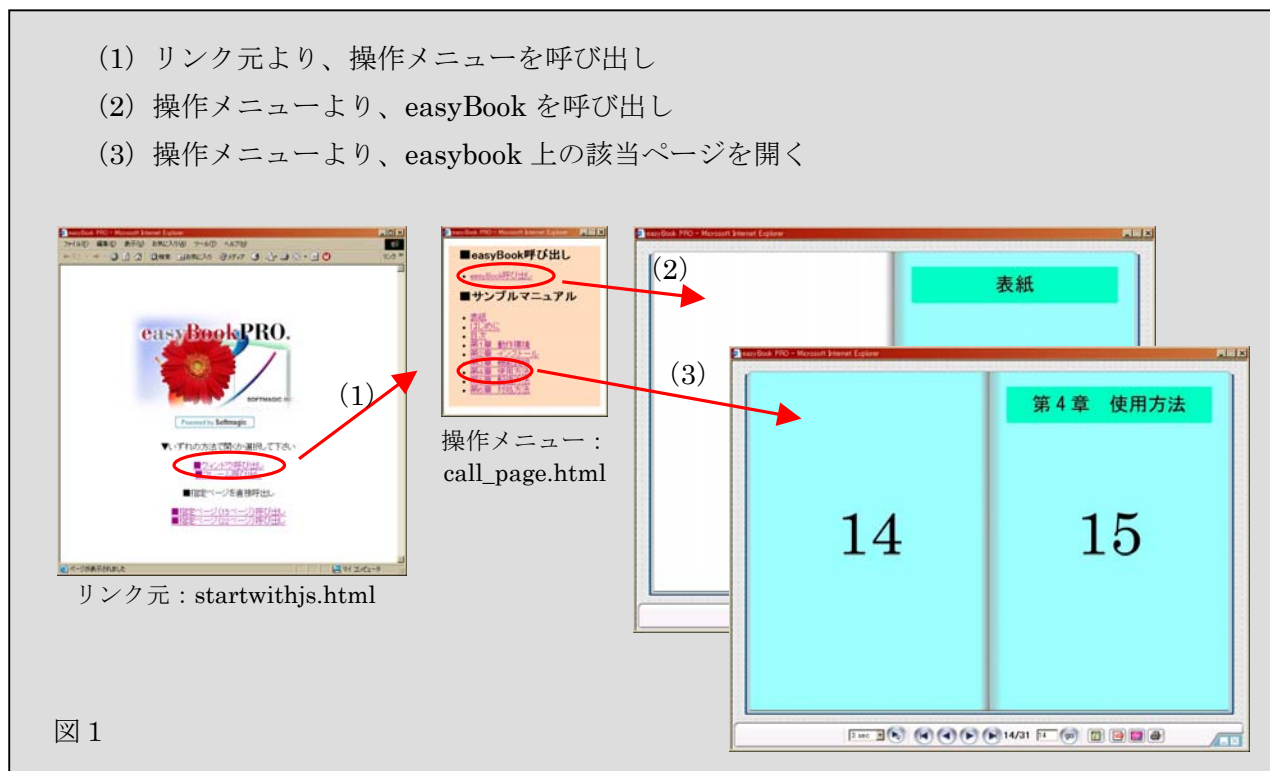
I.操作メニューと easyBook を別ウィンドウで開き操作する方法

1.概要

ここでは、操作メニューと easyBook をそれぞれ別ウィンドウで開き、操作メニュー上から、指定項目に該当するページを開く方法について解説いたします。

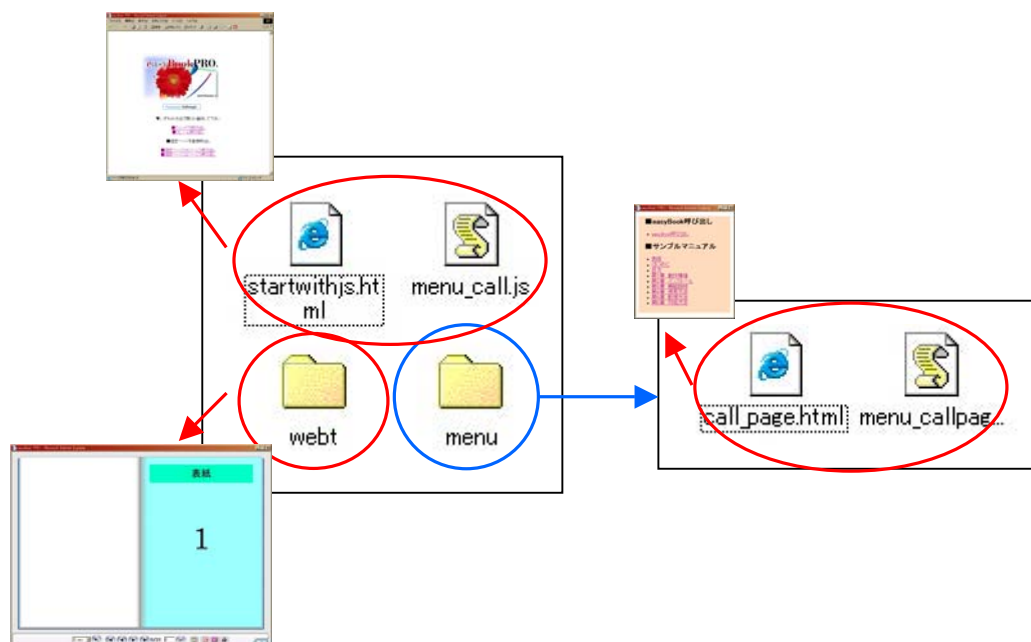
この方法で実現できる easyBook の操作の流れは、以下のようになります。

- (1) リンク元より、操作メニューを呼び出し
- (2) 操作メニューより、easyBook を呼び出し
- (3) 操作メニューより、easybook 上の該当ページを開く



2.ファイル構成

図 1 の状態を例に解説していきます。ファイル構成は下記のようになっているものとします。



3.組み込み方法

(1) リンク元より、操作メニューの呼び出し

ここでは、リンク元 (startwithjs.html) から、操作メニュー (call_page.html) を呼び出すための組み込みについて解説いたします。

a. リンク元 (startwithjs.html) ファイルへの設定

【手順 1】「menu_call.js」ファイルの組み込み

まず、リンク元 (startwithjs.html) と同じ階層に「menu_call.js」ファイルをおきます。
次に、リンク元 (startwithjs.html) のファイルを、テキストエディタや HTML エディタなど (以下、テキストエディタと表記します) で開き、<head>タグの部分に以下の記述を加えます。

```
<script language="JavaScript" src="/menu_call.js"></script>
```

【手順 2】操作メニュー (call_page.html) へのリンク設定

操作メニュー (call_page.html) の呼び出しに使用したいテキスト、または画像にリンクの設定を行ないます。

■テキストとリンクする場合

- ①リンクしたいテキスト (緑部分) を<a>タグで挟み、href 属性で openMenu 関数を呼び出します (赤部分)。
- ②openMenu 関数の引数として、操作メニュー (call_page.html) ファイルまでのパス情報を相対パスで記述します (青部分)。

```
<a href="javascript:openMenu('/menu/call_page.html'); ">■ウィンドウ呼び出し</a><br/>
```

■画像とリンクする場合

- ①リンクしたいタグ (緑部分) の中に style 属性でポインタの形態を、onclick 属性で openMenu 関数を呼び出します (赤部分)。
- ②openMenu 関数の引数として、操作メニュー (call_page.html) ファイルまでのパス情報を相対パスで記述します (青部分)。

```

```

b. 「menu_call.js」ファイルへの設定

【手順 3】操作メニューの表示位置、サイズの定義

操作メニューのウィンドウの表示位置、サイズの定義は、「menu_call.js」ファイルの openMenu 関数の中で行なっています。これらを変更したい場合は、「menu_call.js」ファイルをテキストエディタで開き、下記の記述 (赤部分) を変更して下さい。

```
CallWind=window.open(openurl, "eMenu", "top=0, left=0, width=320, height=350, fullscreen=no");
```

※この例では、表示開始位置 (0,0)、幅 320 ピクセル、高さ 350 ピクセルでウィンドウを表示します。

(2) 操作メニューより、easyBook を呼び出し、指定ページを開く

ここでは、操作メニュー（call_page.html）から、easyBook を呼び出すための組み込み、ページの指定方法等について解説いたします。

a. 操作メニュー（call_page.html）ファイルへの設定

【手順 1】「menu_callpage.js」ファイルの組み込み

まず、操作メニュー（call_page.html）と同じ階層に「menu_callpage.js」ファイルをおきます。次に、操作メニュー（call_page.html）のファイルを、テキストエディタで開き、<head>タグの部分に以下の記述を加えます。

```
<script language="JavaScript" src="./menu_callpage.js"></script>
```

【手順 2】easyBook へのリンク設定

この方法では、まず easyBook をあらかじめ呼び出ししておくことが前提となります。そのため操作メニュー上に easyBook 呼び出し用のテキスト、または画像を配置しておく必要があります。easyBook の呼び出しに使用したいテキスト、または画像にリンクの設定を行いません。

■テキストからの easyBook の呼び出し

①リンクしたいテキスト（緑部分）を<a>タグで挟み、href 属性で openWindow 関数を呼び出します（赤部分）。

②openWindow 関数の引数として、easyBook を呼び出すためのファイル（「webt」フォルダの中の「selfload.html」ファイル）までのパス情報を相対パスで記述します（青部分）。

※「webt」の部分は、easyBook の形態により、「webr」、「webp」、「webl」の場合もあります。

```
<a href="javascript:openWindow('./webt/selfload.html')">easyBook 呼び出し</a>
```

■画像からの easyBook の呼び出し

①リンクしたいタグ（緑部分）の中に style 属性でポインタの形態を、onclick 属性で openWindow 関数を呼び出します（赤部分）。

②openWindow 関数の引数として、easyBook を呼び出すためのファイル（「webt」フォルダの中の「selfload.html」ファイル）までのパス情報を相対パスで記述します（青部分）。

※「webt」の部分は、easyBook の形態により、「webr」、「webp」、「webl」の場合もあります。

```

```

【手順 3】easyBook 上のページの呼び出し

上記、【手順 2】で画面に表示した easyBook 上の指定ページを開くための記述をします。テキスト、または画像にリンクの設定を行います。

■テキストから easyBook の指定ページを呼び出し

①リンクしたいテキスト（緑部分）を<a>タグで挟み、href 属性で callPage 関数を呼び出します（赤部分）。

②callPage 関数の引数として、開きたいページ数を指定します。この際のページ指定は、後述の【手順 4】の設定を行なうことで、ノンブル上のページ数で指定することができます(青部分)。

```
<a href="javascript:callPage(10);">第 3 章 機能説明</a>
```

■画像から easyBook の指定ページを呼び出し

①リンクしたいタグ(緑部分)の中に style 属性でポインタの形態を、onclick 属性で callPage 関数を呼び出します(赤部分)。

②callPage 関数の引数として、開きたいページ数を指定します(青部分)。

※ページの指定方法は、上記「テキストから easyBook の指定ページを呼び出し」の場合と同様です。

```

```

b. 「menu_callpage.js」ファイルへの設定

【手順 4】 ページ数の指定値調整

素材として利用した PDF ファイルにノンブルが含まれている場合、easyBook 上のページ数の扱いとずれている場合があります。前述の【手順 3】でノンブルどおりの値で指定したい場合は、「menu_callpage.js」ファイル内の変数 JustPage の値(赤部分)を変更し、調整して下さい。

```
var JustPage = 0;
```

例えば、5 ページずれている場合は、「var JustPage = 5;」と設定します。ずれている値の分だけ、この数値を変更することで、【手順 3】の指定の際に、ノンブルどおりの指定が可能になります。

なお、easyBook のスキン(操作バー)やしおり上に表示されるページ数と、ノンブルを合わせたい場合は、各スキンに標準搭載されているページ調整機能を併せて使用して下さい。

※ページ調整機能の詳しい方法については、easyBook ホームページの FAQ T02-013 を参照して下さい。

【手順 5】 easyBook 表示ウィンドウの表示位置とサイズの定義

easyBook 表示ウィンドウの表示位置とサイズの定義は、「menu_callpage.js」ファイルの openWindow 関数の中で行なっています。これらを変更したい場合は、「menu_callpage.js」ファイルをテキストエディタで開き、下記の記述(赤部分)を変更して下さい。

```
CallWind = window.open(openurl, "easyBook", "top=10, left=80, width=920, height=650, fullscreen=no");
```

※この例では、表示開始位置(10,80)、幅 920 ピクセル、高さ 650 ピクセル、フルスクリーン表示オフでウィンドウを表示します。

※書き出された easyBook の表示サイズをあとから変更することはできません。ここで定義した表示ウィンドウサイズに収まるように easyBook の表示サイズを変更したい場合は、easyBook PRO.で再度書き出しを行ってください。「easyBook 書き出し」画面の「メインページ」の「幅」の値を変更することで、サイズ調整をすることができます。

【手順6】easyBookが開いていない場合の処理

この方法では、あらかじめeasyBookが開かれていることが前提となります。開かれていない状態で、ページを指定した場合の処理を、「menu_callpage.js」ファイルのcallPage関数の中で行なっています。処理内容を変更したい場合には、下記の部分を変更して下さい。

```
window.alert("はじめに easyBook を開いて下さい。");
```

※この例では、window.alert 文を使い、「はじめに easyBook を開いて下さい。」というメッセージを表示させています。

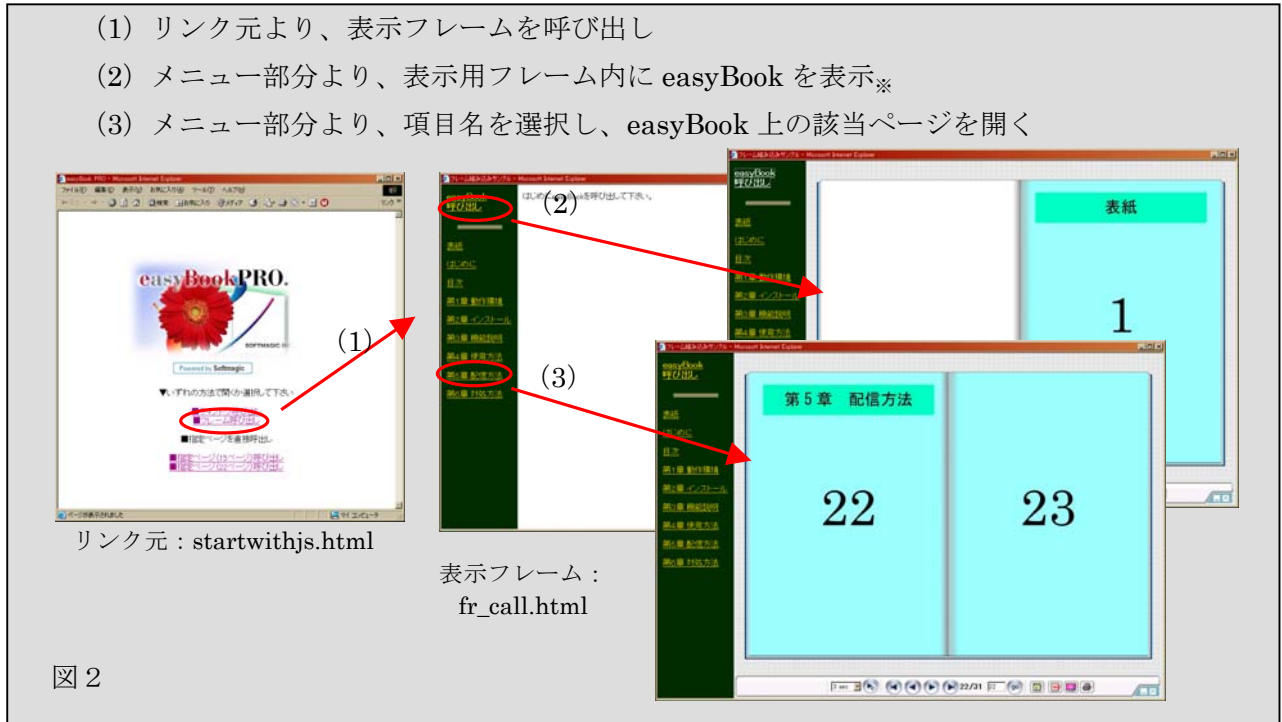
II. フレーム内にメニュー表示エリアと easyBook 表示エリアを設けて操作する方法

1. 概要

表示フレーム内に操作メニューと easyBook をそれぞれ表示し、操作メニュー上から、指定項目に該当するページを開く方法について解説いたします。

この方法で実現できる easyBook の操作の流れは、以下のようになります。

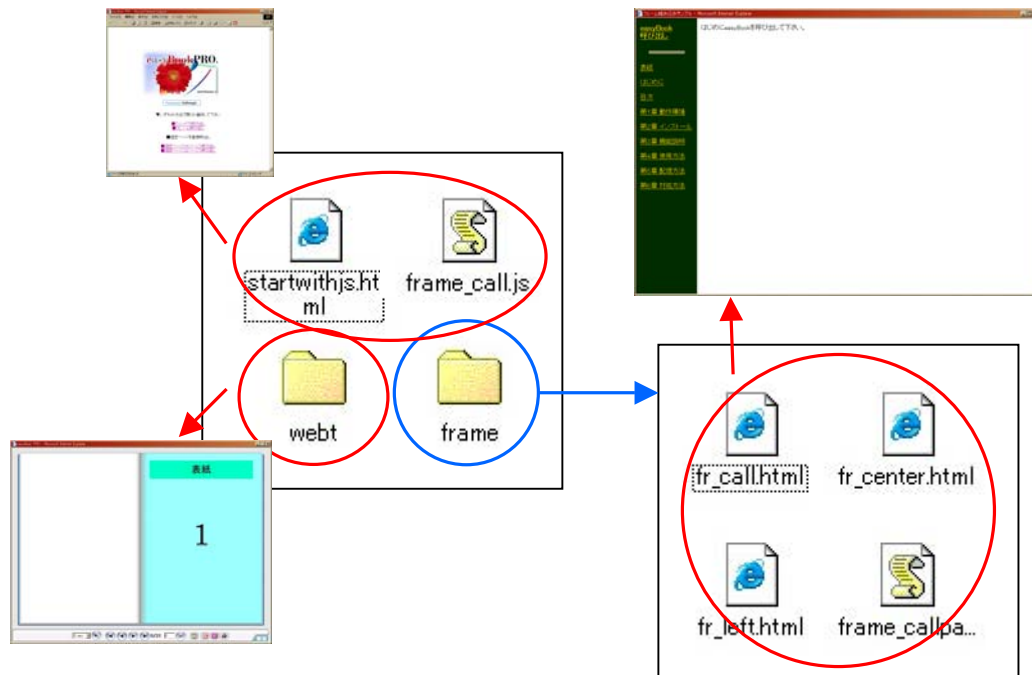
- (1) リンク元より、表示フレームを呼び出し
- (2) メニュー部分より、表示用フレーム内に easyBook を表示※
- (3) メニュー部分より、項目名を選択し、easyBook 上の該当ページを開く



※Windows 版 Internet Explorer で閲覧している場合には、フレームを呼び出すときに、easyBook の先頭ページを開いた状態で呼び出すことが可能です。

2. ファイル構成

図 2 の状態を例に解説していきます。ファイル構成は下記のようになっているものとします。



3.組み込み方法

(1) リンク元より、表示フレームの呼び出し

ここでは、リンク元 (startwithjs.html) から、表示フレーム (fr_call.html) を呼び出すための組み込みについて解説いたします。

a. リンク元 (startwithjs.html) ファイルへの設定

【手順 1】「frame_call.js」ファイルの組み込み

まず、リンク元 (startwithjs.html) と同じ階層に「frame_call.js」ファイルをおきます。次に、リンク元 (startwithjs.html) のファイルを、テキストエディタで開き、<head>タグの部分に以下の記述を加えます。

```
<script language="JavaScript" src="./frame_call.js"></script>
```

【手順 2】表示フレーム (fr_call.html) へのリンク設定

表示フレーム (fr_call.html) の呼び出しに使用したいテキスト、または画像にリンクの設定を行ないます。

■テキストとリンクする場合

- ①リンクしたいテキスト (緑部分) を<a>タグで挟み、href 属性で openFrame 関数を呼び出します (赤部分)。
- ②openFrame 関数の引数として、表示フレーム (fr_call.html) ファイルまでのパス情報を相対パスで記述します (青部分)。

```
<a href="javascript:openFrame('./frame/fr_call.html');">■フレーム呼び出し </a>
```

■画像とリンクする場合

- ①リンクしたいタグ (緑部分) の中に style 属性でポインタの形態を、onclick 属性で openFrame 関数を呼び出します (赤部分)。
- ②openFrame 関数の引数として、表示フレーム (fr_call.html) ファイルまでのパス情報を相対パスで記述します (青部分)。

```

```

b. 「frame_call.js」ファイルへの設定

【手順 3】表示フレームの表示サイズ、位置の定義

表示フレームのウィンドウサイズの定義は、「frame_call.js」ファイルの openFrame 関数の中で行なっています。表示サイズを変更したい場合は、「frame_call.js」ファイルをテキストエディタで開き、下記の記述 (赤部分) を変更して下さい。

```
CallWind=window.open(openurl, "eFrame", "top=0, left=0, width=1012, height=690, resizable=yes, fullscreen=no");
```

※この例では、表示開始位置 (0,0)、幅 1,012 ピクセル、高さ 690 ピクセル、ウィンドウサイズ変更可能、フルスクリーン表示オフでウィンドウを表示します。

【手順 4】 easyBook 表示領域へ easyBook を表示（Windows 閲覧時）

① easyBook 表示領域への表示は、同じく「frame_call.js」ファイルの openFrame 関数の中でフレームを表示した後に続けて行なっています。下記がその部分になります。window.open 文の第 1 引数で、easyBook を呼び出すためのファイル（「webt」フォルダの中の「selfload.html」ファイル）までのパス情報を相対パスで記述します（青部分）。

※「webt」の部分は、easyBook の形態により、「webr」、「webp」、「webl」の場合もあります。

②次に、第 2 引数に easyBook を表示したいフレーム名を指定します。ここでは、「easySpace」（緑部分）を指定しています。なお、このフレーム名は、「fr_call.html」で定義しています。

```
if (agent != "MacPPC" && agent != "MacIntel"){  
    window.open("./webt/selfload.html", "easySpace");  
}
```

※この手法は、Windows で閲覧した場合のみ正常に表示されます。Macintosh で同じ手法を使用した場合、フレームと easyBook がそれぞれ別ウィンドで表示されます。そのため、本サンプルでは、閲覧に使用している OS が Windows、Macintosh いずれかを判断し、Windows で閲覧した場合のみ、この形態が有効になるように設定しています。Macintosh で閲覧している場合は、別途 easyBook を指定フレーム内に表示する手段が必要です。詳細については、次項「(2) メニュー部分より、表示フレーム内に easyBook を表示」を参照して下さい。

(2) メニュー部分より、表示フレーム内に easyBook を表示

ここでは、表示フレームのメニュー部分（fr_left.html）から、easyBook 表示部（fr_center.html）に easyBook を呼び出すための組み込み、呼び出す方法等について解説いたします。なお、今回説明に使用するフレームの構成は、以下のようになっているものとします。

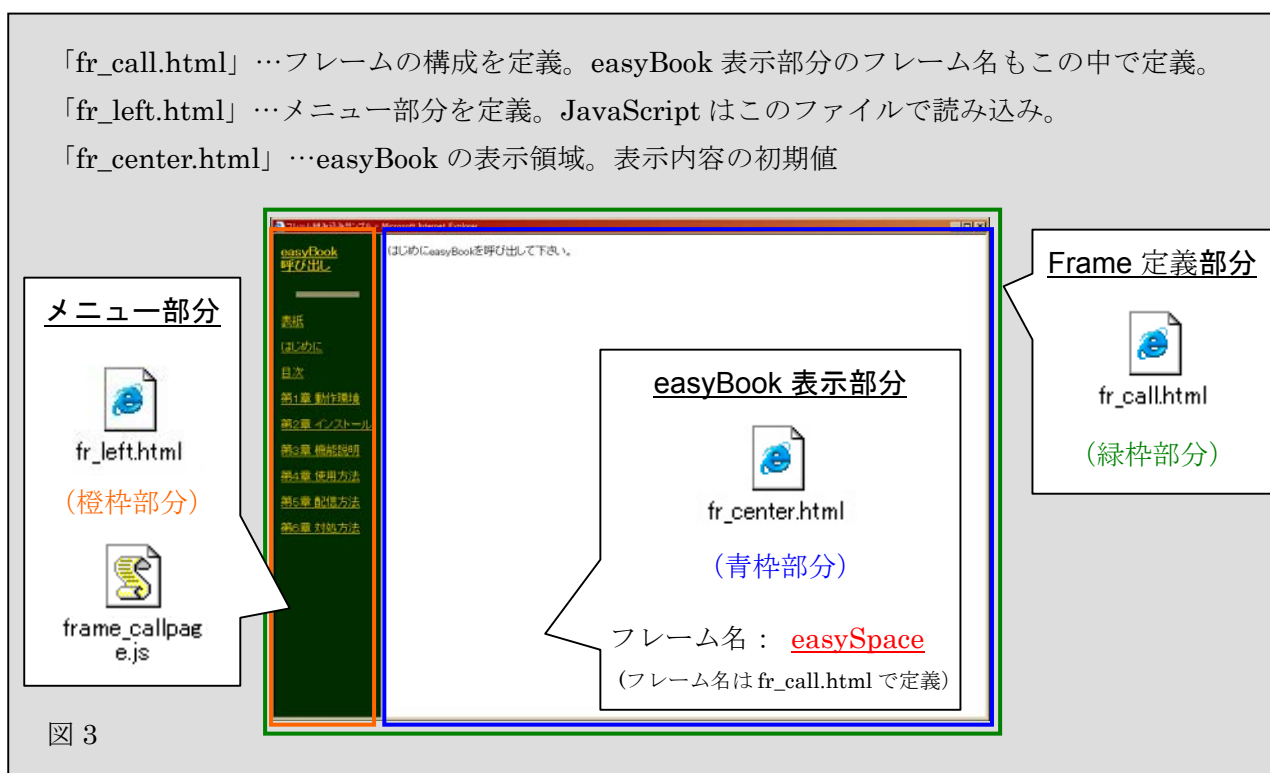


図 3

a. メニュー表示部 (fr_left.html) ファイルへの設定

【手順 1】「frame_callpage.js」ファイルの組み込み

まず、メニュー部分 (fr_left.html) と同じ階層に「frame_callpage.js」ファイルをおきます。次に、メニュー部分 (fr_left.html) のファイルをテキストエディタで開き、<head>タグの部分に以下の記述を加えます。

```
<script language="JavaScript" src="/frame_callpage.js"></script>
```

【手順 2】easyBook 表示領域へ easyBook を表示

この方法では、まず easyBook をあらかじめ easyBook 表示部に呼び出しておくことが前提となります。そのためメニュー部分に easyBook 呼び出し用のテキスト、または画像を配置しておく必要があります。

easyBook の呼び出しに使用したいテキスト、または画像にリンクの設定を行いません。

■テキストからの easyBook の呼び出し

①リンクしたいテキスト (緑部分) を<a>タグで挟みます (赤部分)。href 属性で easyBook を呼び出すためのファイル (「webt」フォルダの中の「selfload.html」ファイル) までのパス情報を相対パスで記述します (青部分)。

※「webt」の部分は、easyBook の形態により、「webr」、「webp」、「webl」の場合もあります。

②同じく<a>タグの内部に target 属性で、easyBook をフレーム内のどの部分に表示するかを定義します。ここでは、フレーム名「easySpace」(黒部分)の部分に表示します。なお、このフレーム名は、「fr_call.html」で定義しています。

```
<a href="/webt/selfload.html" target="easySpace">easyBook<br/>呼び出し</a>
```

■画像からの easyBook の呼び出し

①リンク元にしたタグ (緑部分) を<a>タグではさみます (赤部分)。href 属性で easyBook を呼び出すためのファイル (「webt」フォルダの中の「selfload.html」ファイル) までのパス情報を相対パスで記述します (青部分)。

※「webt」の部分は、easyBook の形態により、「webr」、「webp」、「webl」の場合もあります。

②テキストの場合と同様に、<a>タグの内部に target 属性で、easyBook を表示するフレーム名を定義します (黒部分)。下記の例では、フレーム名「easySpace」の部分に表示します。

```
<a href="/webt/selfload.html" target="easySpace"></a>
```

【手順 3】easyBook 上のページの呼び出し

上記、【手順 2】で画面に表示した easyBook 上の指定ページを開くための記述をします。テキスト、または画像にリンクの設定を行いません。

■テキストから easyBook の指定ページを呼び出し

①リンクしたいテキスト (緑部分) を<a>タグで挟み (赤部分)、href 属性と onclick 属性を併用して、callFramePage 関数を呼び出します (青部分)。

②callFramePage 関数の引数として、開きたいページ数を指定します。後述の【手順 4】の設定を行なうことで、ノンブル上のページ数でも指定することができます（黒部分）。

```
<a href="javascript:void(0)" onclick="javascript:callFramePage(4);">目次</a>
```

■ 画像から easyBook の指定ページを呼び出し

①リンクしたいタグ（緑部分）の中に style 属性でポインタの形態を、onclick 属性で callFramePage 関数を呼び出します（赤部分）。

②callFramePage 関数の引数として、開きたいページ数を指定します（青部分）。

※ページの指定方法は、上記「テキストから easyBook の指定ページを呼び出し」の場合と同様です。

```

```

b. 「frame_callpage.js」ファイルへの設定

【手順 4】 ページ数の指定値調整

上記、【手順 3】でノンブルどおりの値で指定したい場合は、「frame_callpage.js」ファイル内の変数 JustPage の値（赤部分）を変更し、調整して下さい。

※詳しくは、「I.操作メニューと easyBook を別ウィンドウで開き操作する方法」の説明の中の「3.組み込み方法」の【手順 4】（本資料 P.5）を参照して下さい。

```
var JustPage = 0;
```

【手順 5】-補足- easyBook が開いていない場合の処理

この方法は、あらかじめ easyBook 表示部に easyBook が開かれていることが前提となります。そのため、「frame_callpage.js」ファイルでは、easyBook が開かれていない状態で項目（ページ）を指定した場合、警告画面を開くよう記述されています。処理内容を変更したい場合には、「frame_callpage.js」ファイルの callFramePage 関数の下記の部分を変更して下さい。

```
window.alert("はじめに easyBook を開いて下さい。");
```

※この例では、window.alert 文を使い、「はじめに easyBook を開いて下さい。」というメッセージを表示させています。

【手順 6】-補足- 任意のフレーム名を設定したい場合

今回の例では、easyBook 表示部のフレーム名を「easySpace」と設定した場合で紹介していません（本サンプルでは、「fr_call.html」ファイル内でフレーム名を定義しています）。作成時、任意のフレーム名を設定していただいてもかまいませんが、その場合には、「frame_callpage.js」ファイルの callFramePage 関数の記述も一部変更する必要があります。

callFramePage 関数の中の下記で示す「easySpace」の部分（赤部分）を設定したフレーム名と同じ名称に書き換えて下さい。

```
parent.frames['easySpace'].frames['Frame_Screen'].GotoPage(cpage);
```

Ⅲ. リンク元の HTML 上から直接 easyBook の指定のページを開く操作方法

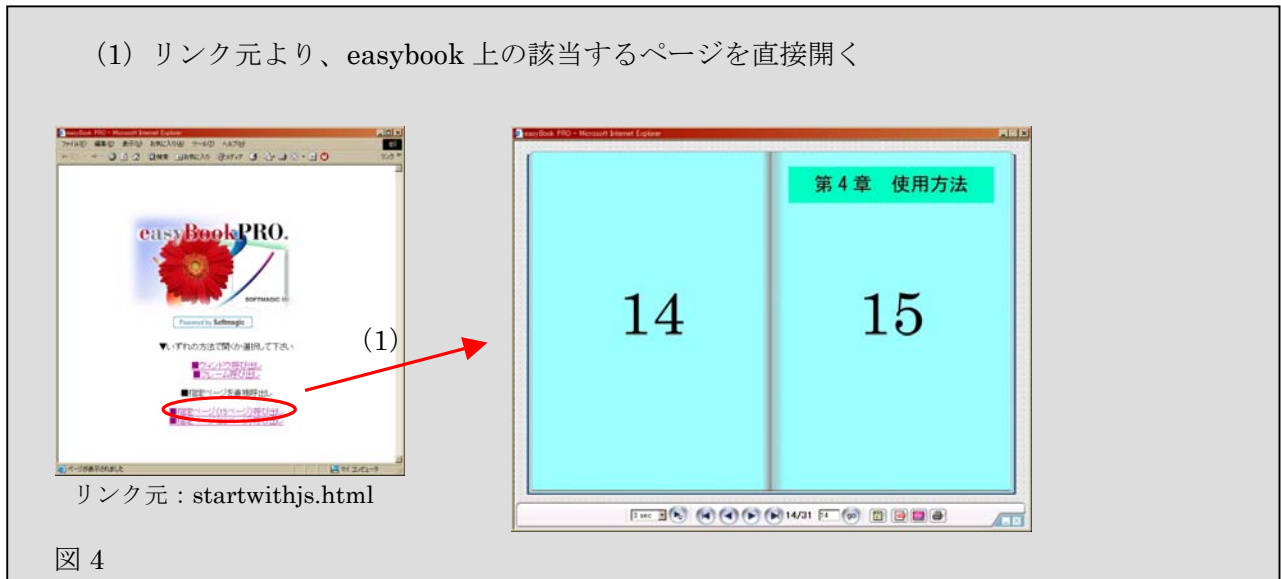
1.概要

ここでは、リンク元の HTML 上のリンクから直接指定のページを開く方法について解説します。

ただし、この方法は、動作の安定性に欠ける場合があります（特に Internet Explorer で閲覧している場合）。
ご使用の際はあらかじめご注意ください。

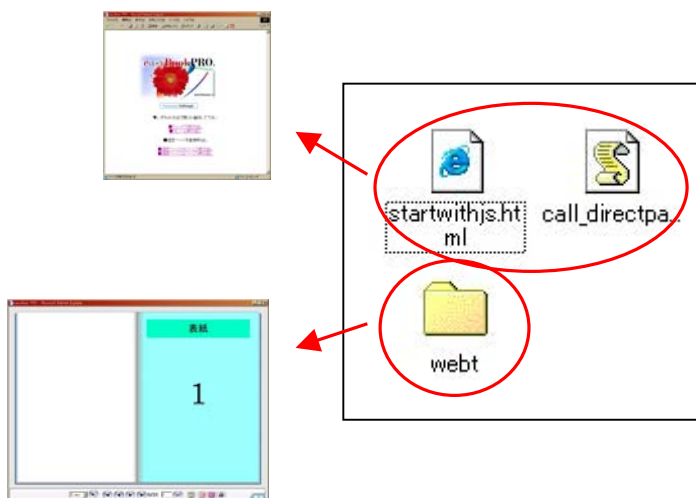
確実に動作させたい場合は、前述の「操作メニューを表示して操作」、または、「フレーム内にメニューを表示しての操作」のご使用もご検討下さい。

この方法で実現できる easyBook の操作の流れは、以下のようになります。



2.ファイル構成

図 4 の状態を例に解説していきます。ファイル構成は以下のようになっているものとします。



3.組み込み方法

(1) リンク元より、操作メニューの呼び出し

ここでは、リンク元 (startwithjs.html) から、easyBook 上の該当するページを直接開くための組み込み方法について解説いたします。

a.リンク元 (startwithjs.html) ファイルへの設定

【手順 1】「call_directpage.js」ファイルの組み込み

まず、リンク元 (startwithjs.html) と同じ階層に「call_directpage.js」ファイルをおきます。次に、リンク元 (startwithjs.html) のファイルを、テキストエディタで開き、<head>タグの部分に以下の記述を加えます。

```
<script language="JavaScript" src="./call_directpage.js"></script>
```

【手順 2】easyBook 上のページを直接呼び出し

この方法では、easyBook 上のページを直接呼び出すことが可能です。easyBook の呼び出しに使用したいテキスト、または画像にリンクの設定を行ないます。

■テキストからの easyBook の呼び出し

- ①リンクしたいテキスト (緑部分) を<a>タグで挟み、href 属性で directOpen 関数を呼び出します (赤部分)。
- ②directOpen 関数の第 1 引数として、easyBook を呼び出すためのファイル (「webt」フォルダの中の「selfload.html」ファイル) までのパス情報を相対パスで記述します (青部分)。
※「webt」の部分は、easyBook の形態により、「webr」、「webp」、「webl」の場合もあります。
- ③directOpen 関数の第 2 引数として、開きたいページ数を指定します。この際の指定は、後述の【手順 3】の設定を行なうことで、ノンプル上のページ数で指定することができます (黒部分)。

```
<a href="javascript:directOpen('./webt/selfload.html',15);">■指定ページ (15 ページ) 呼び出し</a>
```

■画像からの easyBook の呼び出し

- ①リンクしたいタグ (緑部分) の中に style 属性でポインタの形態を、onclick 属性で directOpen 関数を呼び出します (赤部分)。
- ②directOpen 関数の第 1 引数として、easyBook を呼び出すためのファイル (「webt」フォルダの中の「selfload.html」ファイル) までのパス情報を相対パスで記述します (青部分)。
※「webt」の部分は、easyBook の形態により、「webr」、「webp」、「webl」の場合もあります。
- ③directOpen 関数の第 2 引数として、開きたいページ数を指定します (黒部分)。
※ページの指定方法は、上記「テキストから easyBook の指定ページを呼び出し」の場合と同様です。

```

```

b. 「call_directpage.js」ファイルへの設定

【手順 3】 ページ数の指定値調整

上記、【手順 2】 でノンブルどおりの値で指定したい場合は、「call_directpage.js」ファイル内の変数 JustPage の値（赤部分）を変更し、調整して下さい。

※詳しくは、「I.操作メニューと easyBook を別ウィンドウで開き操作する方法」の説明の中の「3.組み込み方法」の【手順 4】（本資料 P.5）を参照して下さい。

```
var JustPage = 0;
```

【手順 4】 easyBook 表示ウィンドウの表示位置とサイズの定義

easyBook 表示ウィンドウサイズの定義は、「call_directpage.js」ファイルの directOpen 関数の中で行なっています。表示ウィンドウサイズを変更したい場合は、「call_directpage.js」ファイルをテキストエディタで開き、下記の記述（赤部分）を変更して下さい。

```
CallWind = window.open(openurl, "easyBook", "top=10, left=80, width=920, height=650, fullscreen=no");
```

※この例では、表示開始位置（10,80）、幅 920 ピクセル、高さ 650 ピクセル、フルスクリーン表示オフでウィンドウを表示します。

※書き出された easyBook の表示サイズをあとから変更することはできません。ここで定義した表示ウィンドウサイズに収まるように easyBook の表示サイズを変更したい場合は、easyBook PRO.で再度書き出しを行ってください。「easyBook 書き出し」画面の「メインページ」の「幅」の値を変更することで、サイズ調整をすることができます。

【手順 5】 easyBook を開いてから指定ページを開くまでの時間の変更

easyBook を開いてから、指定ページを開くまでの時間は、「call_directpage.js」ファイルの directOpen 関数の中での定義しています。この時間を変更したい場合は、「call_directpage.js」ファイルをテキストエディタ等で開き、下記の記述（赤部分）を変更して下さい。

```
if((userinfo.indexOf('MSIE') >= 0) && (userinfo.indexOf('Opera') < 0) && (userinfo.indexOf('Firefox') < 0)) { // For Internet Explorer
  //Waiting for display of easyBook
  openNum = num;
  setTimeout("movePage(openNum)",3000);
} else { // For other browsers
  cpage = num;
  CallWind.location.href = "./webt/selfload.html?" + cpage;
  CallWind.focus();
}
```

※値の指定時の単位は、ミリ秒です（1000 ミリ秒 = 1 秒）。

※この例では、3 秒後に指定されたページを表示します。

※値は任意に変更してかまいませんが、指定する値を小さくしすぎると指定したページが表示されず、easyBook が起動しただけで動作が終了してしまう場合があります。

※easyBook は、フレームで構成されているため、それぞれのパーツが表示された後にページ指定を行なう必要があります。そのため setTimeout 関数を使用して時間調整を行なっています。

【資料】

本資料の各項目で使用した HTML、JavaScript ファイルのリストを掲載します。本文中で説明した箇所を赤枠で囲い項目番号を表記していますので参考にして下さい。

■起動用共通ファイル：startwithjs.html

```
<html>
<head>
  <meta http-equiv="Content-Type" content="text/html; charset=Shift_JIS">
  <script language="JavaScript" src="/menu_call.js"></script>    … I-3-(1)-a
  <script language="JavaScript" src="/frame_call.js"></script>  … II-3-(1)-a
  <script language="JavaScript" src="/call_directpage.js"></script> … III-3-(1)-a
  <title>easyBook PRO</title>
</head>
<body bgcolor="#FFFFFF">
<p align="center">&nbsp;</p>
<p align="center">&nbsp;</p>
<table width="392" border="0" cellspacing="0" cellpadding="0" align="center" >
  <tr>
    <td width="221" align="center"></td>
  </tr>
  <tr>
    <td align="center">
      <br/>
      <p>▼いずれの方法で開くか選択して下さい</p>
      <p>
        <a href="javascript:openMenu('/menu/call_page.html');">■ウィンドウ呼び出し</a><br/>    … I-3-(1)-a
        <a href="javascript:openFrame('/frame/fr_call.html');">■フレーム呼び出し </a><br/>    … II-3-(1)-a
      </p>
      <p>■指定ページを直接呼出し</p>
      <p>
        <a href="javascript:directOpen('/webt/selfload.html',15);">■指定ページ（15 ページ）呼び出し</a><br/>
        <a href="javascript:directOpen('/webt/selfload.html',22);">■指定ページ（22 ページ）呼び出し</a><br/>
      </p>
    </td>
  </tr>
</table>
</body>
</html>
```

「I.操作メニューと easyBook を別ウィンドウで開き操作する方法」で使用したファイル

■リスト A : menu_call.js

```
//Declaration of variable
var openurl;
var CallWind;

// Menu Window & Main Window
function openMenu(openurl) {
    CallWind=window.open(openurl, "eMenu", "top=0, left=0, width=320, height=350, fullscreen=no"); ... I-3-(1)-b
    CallWind.focus();
}
```

■リスト B : call_page.html

```
<html>
<head>
    <meta http-equiv="Content-Type" content="text/html; charset=Shift_JIS">
    <script language="JavaScript" src="/menu_callpage.js"></script> ... I-3-(2)-a
    <title>easyBook PRO</title>
</head>
<body bgcolor="#FFFFFF">
<table bgcolor="#FFdab9" width="300" height="320">
    <tr>
        <td >
            <h2> ■easyBook 呼び出し</h2>
            <ul>
                <li><a href="javascript:openWindow('../webt/selfload.html')">easyBook 呼び出し</a></li>
            </ul>
            <h2> ■サンプルマニュアル</h2> ... I-3-(2)-a
            <ul>
                <li><a href="javascript:callPage(1);">表紙</a></li>
                <li><a href="javascript:callPage(2);">はじめに</a></li>
                <li><a href="javascript:callPage(4);">目次</a></li>
                <li><a href="javascript:callPage(7);">第1章 動作環境</a></li>
                <li><a href="javascript:callPage(8);">第2章 インストール</a></li>
                <li><a href="javascript:callPage(10);">第3章 機能説明</a></li>
                <li><a href="javascript:callPage(15);">第4章 使用方法</a></li>
                <li><a href="javascript:callPage(22);">第5章 配信方法</a></li>
                <li><a href="javascript:callPage(27);">第6章 対処方法</a></li>
            </ul>
            </td>
        </tr>
    </table>
</body>
</html>
```

■リスト C : menu_callpage.js

```
//Declaration of variable
var openurl;
var CallWind;
var num;
var cpage;
var JustPage = 0; ... I-3-(2)-b

var userinfo = navigator.userAgent

// (Menu Window) & Main Window
function openWindow(openurl) {
    CallWind = window.open(openurl, "easyBook", "top=10, left=80, width=920, height=650, fullscreen=no");
    CallWind.focus(); ... I-3-(2)-b
}

//Call Main Window's Page
function callPage(num){

    if((userinfo.indexOf('MSIE') >= 0) && (userinfo.indexOf('Opera') < 0) && (userinfo.indexOf('Firefox') < 0)) {
        if (CallWind && !CallWind.closed) {
            if (CallWind.startRight && CallWind.spreadPage) { // 1st page ON
                cpage = num + (JustPage + 1);
            } else { // 1st page OFF
                cpage = num + JustPage;
            }
            CallWind.frames['Frame_Screen'].GotoPage(cpage);
            CallWind.focus();
        } else {
            window.alert("はじめに easyBook を開いて下さい。"); ... I-3-(2)-b
        }
    } else {
        cpage = num;
        if (CallWind && !CallWind.closed) {
            CallWind.location.href = "../webt/selfload.html?" + cpage;
            CallWind.focus();
        } else {
            window.alert("はじめに easyBook を開いて下さい。"); ... I-3-(2)-b
        }
    }
}
}
```

「Ⅱ. フレーム内にメニュー表示エリアと easyBook 表示エリアを設けて操作する方法」で使ったファイル

■ リスト D : frame_call.js

```
//Declaration of variable
var openurl;
var CallWind;

// Frame Window
function openFrame(openurl) {
    var agent = navigator.platform;
    CallWind=window.open(openurl, "eFrame", "top=0, left=0, width=1012, height=690, resizable=yes, fullscreen=no");
    //easyBook open for Windows
    if (agent != "MacPPC" && agent != "MacIntel" ){
        window.open("./webt/selfload.html","easySpace");
    }
    CallWind.focus();
}
```

■ リスト E : fr_call.html

```
<html>
  <head>
    <meta http-equiv="Content-Type" content="text/html; charset=Shift_JIS">
    <title>フレーム組み込みサンプル</title>
  </head>
  <frameset cols="102,910" frameborder="NO" framespacing="0" border="0" >
    <frame src="./fr_left.html" scrolling="NO">
    <frame src="./fr_center.html" name="easySpace" scrolling="AUTO">
  </frameset>
</html>
```

■ リスト F : fr_center.html

```
<html>
  <head>
    <meta http-equiv="Content-Type" content="text/html; charset=Shift_JIS">
    <title></title>
  </head>
  <body>
    <p>はじめに easyBook を呼び出して下さい。 </p>
  </body>
</html>
```

■ リスト G : fr_left.html

```
<html>
  <head>
    <meta http-equiv="Content-Type" content="text/html; charset=Shift_JIS">
    <script language="JavaScript" src="./frame_callpage.js"></script> ... II-3-(2)-a
    <title></title>
  </head>
  <body bgcolor="#003300" text="#cccc99" link="#999900" vlink="#ffcc00" alink="#ffffff">
    <h3><a href="../webt/selfload.html" target="easySpace">easyBook<br/>呼び出し</a></h3>
    <hr width="90" size="8" noshade /> ... II-3-(2)-a
    <p><a href="javascript:void(0)" onclick="javascript:callFramePage(1);">表紙</a></p>
    <p><a href="javascript:void(0)" onclick="javascript:callFramePage(2);">はじめに</a></p>
    <p><a href="javascript:void(0)" onclick="javascript:callFramePage(4);">目次</a></p>
    <p><a href="javascript:void(0)" onclick="javascript:callFramePage(7);">第 1 章<br/>動作環境
</a></p>
    <p><a href="javascript:void(0)" onclick="javascript:callFramePage(8);">第 2 章<br/>インスト
ール</a></p>
    <p><a href="javascript:void(0)" onclick="javascript:callFramePage(10);">第 3 章<br/>機能説
明</a></p>
    <p><a href="javascript:void(0)" onclick="javascript:callFramePage(15);">第 4 章<br/>使用方
法</a></p>
    <p><a href="javascript:void(0)" onclick="javascript:callFramePage(22);">第 5 章<br/>配信方
法</a></p>
    <p><a href="javascript:void(0)" onclick="javascript:callFramePage(27);">第 6 章<br/>対処方
法</a></p>
  </body> ... II-3-(2)-a
</html>
```

■リスト H : frame_callpage.js

```
//Declaration of variable
var num;
var cpage;
var JustPage = 0;    ... II-3-(2)-b

var userinfo = navigator.userAgent;

// Call Frame Window's Page
function callFramePage(num){
    var whName = parent.easySpace.location; //Getting of target space's URL
    var txName = whName.toString();        //Converting Object into String
    var cfThere = txName.indexOf("wysiweb"); //Confirming whether the URL contains 'wysiweb'.

    if((userinfo.indexOf('MSIE') >= 0)  && (userinfo.indexOf('Opera') < 0) && (userinfo.indexOf('Firefox') < 0)) {

        if (cfThere > 0) {
            if ((parent.frames['easySpace'].startRight) && (parent.frames['easySpace'].spreadPage)) {
                // 1st page ON
                cpage = num + (JustPage + 1);
            } else {
                // 1st page OFF
                cpage = num + JustPage;
            }
            parent.frames[easySpace].frames['Frame_Screen'].GotoPage(cpage);
        } else {    ... II-3-(2)-b
            window.alert("はじめに easyBook を開いて下さい。");    ... II-3-(2)-b
        }

    } else {
        cpage = num;
        if (cfThere > 0) {
            parent.frames['easySpace'].location.href = "../webt/selfload.html?" + cpage;
        } else {
            window.alert("はじめに easyBook を開いて下さい。");    ... II-3-(2)-b
        }
    }
}
}
```

「Ⅲ. リンク元の HTML 上から直接 easyBook の指定のページを開く操作方法」で使用したファイル

■リスト1: call_directpage.js

```
//Declaration of variable
var openurl;
var CallWind;
var num;
var cpage;
var JustPage = 0;    ... Ⅲ-3-(1)-b

var userinfo = navigator.userAgent
// Declaration of variable for Function movePage()
var whichBook;
var openNum;

// Open direct page
function directOpen(openurl, num) {
    var agent = navigator.platform;    ... Ⅲ-3-(1)-b
    CallWind = window.open(openurl, "easyBook", "top=10, left=80, width=920, height=650, fullscreen=no");
    if((userinfo.indexOf('MSIE') >= 0) && (userinfo.indexOf('Opera') < 0) && (userinfo.indexOf('Firefox') <
0)) { // For Internet Explorer
        //Waiting for display of easyBook
        openNum = num;
        setTimeout("movePage(openNum)",3000);    ... Ⅲ-3-(1)-b
    } else { // For other browsers
        cpage = num;
        CallWind.location.href = "./webt/selfload.html?" + cpage;
        CallWind.focus();
    }
}

function movePage(openNum){
    if (CallWind.spreadPage) { // For webt and webr
        whichBook = CallWind.startRight;
        if (whichBook == 0) {
            cpage = openNum + JustPage;
            CallWind.frames['Frame_Screen'].GotoPage(cpage);
        } else if (whichBook) {
            cpage = openNum + (JustPage + 1);
            CallWind.frames['Frame_Screen'].GotoPage(cpage);
        } else {
            alert('表示速度の関係で正しいページが表示されていない可能性があります。ご確認下さい。');
        }
    } else { // For webp and webl
        cpage = openNum + JustPage;
        CallWind.frames['Frame_Screen'].GotoPage(cpage);
    }
}
```