

2014 年度 メディアプロジェクト演習 1

HTML 講座 発展編 2 ~ CSS と Web アクセシビリティ

立命館大学情報理工学部 メディア情報学科

進捗が遅れている場合に限り、テキスト中で【省略可】マークのついた節は省略してもよい。

1 はじめに

本演習ではこれまで、様々なタグを用いた基礎的な Web ページの作成方法・レイアウト方法について学んできました。ここで、タグは本来、Web ページ上の文章構造を決定するために用いるものです。また、タグでレイアウトを決定すると、HTML 文章の構造が複雑になり、Web ページの運用・保守が非常に困難となります。そのため、Web ページのレイアウトに関しては、Cascading Style Sheets (カスケーディング・スタイル・シート: CSS) を用いてそれを実現することが推奨されてきました。

そこで発展編 2 では、CSS の基礎とその記述方法を説明し、CSS を用いた Web ページ作成について理解を深めることを目標とします。また、世界中の人々が分け隔てなく情報資源にアクセスできるようにすることを目的とした、Web アクセシビリティの考え方についても学びます。

2 Cascading Style Sheets

CSS は、基本的には、各タグで囲まれた文章をどのように表示するかを決定します。CSS を用いる際には、必ず以下の記述が必要となります。

タグ	<code><meta http-equiv="Content-Style-type" content="text/css"></code>
機能	CSS を使用することを宣言します。

(sample34.html)

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML
4.0 Transitional//EN">
<html>
<head>
<meta http-equiv="Content-Style-type"
content="text/css">
</head>
<body>

</body>
</html>
```

なお、ページ幅の都合上改行してありますが、改行はしなくて結構です。しかし、この改行部分に相当するところに、半角スペースを入力してください。

3 CSS の記述方法

3.1 CSS の書式

CSS は、基本的には、以下の書式をとります。

セレクタ {プロパティ: 値;...}

ここで、セレクタは適用範囲、プロパティは何を指定するか（文字の色、形など）、値は具体的な内容（赤、MS 明朝など）を指定します。また、セレクタに示した適用範囲に複数の指定（文字の色と形の両方など）をしたい場合は、「;」で区切ります。このとき、「;」で区切った後に改行すると見やすい場合があります。

3.2 CSS の 3 つの実現方法

CSS には以下に示す 3 つの実現方法があります。

- style 属性を使う方法
- HTML ファイルの<head> ~ </head>間に CSS を記述する方法
- CSS のみを記述した別ファイル(***.css)を用意し、HTML ファイルからリンクを貼る方法

style 属性を使う場合は以下のように記述します。

タグ	<code><要素名 style="プロパティ:値"></code>
機能	要素のスタイルを指定します。

(sample35.html)

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML
4.0 Transitional//EN">
<html>
<head>
<meta http-equiv="Content-Style-type"
content="text/css">
</head>
<body>
<p style="color:#FF0000">赤い文字</p>
</body>
</html>
```

次に、HTML ファイルの<head> ~ </head>間に CSS を記述する場合は、以下のように記述します。

タグ	<code><style type="text/css"> ~ </style></code>
機能	CSS を HTML ファイルの中で記述します

(sample36.html)

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML
4.0 Transitional//EN">
<html>
<head>
<meta http-equiv="Content-Style-type"
content="text/css">
<style type="text/css">
<!--
    p{color:#FF0000}
-->
</style>
</head>
<body>
<p>赤い文字</p>
</body>
</html>
```

ここで、<!-- -->は、CSS に対応していないブラウザで閲覧したときに、この中の部分がそのままテキストとして表示されないようにするための配慮です。

最後に、CSS のみを記述した別ファイルを用意し、HTML ファイルからリンクを貼る場合は、以下のように記述します。

タグ	<link rel="stylesheet" href="***.css" type="text/css">
機能	***.css ファイル内で記述した CSS を、HTML ファイルの中で使用します。

(sample37.html)

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML
4.0 Transitional//EN">
<html>
<head>
<meta http-equiv="Content-Style-type"
content="text/css">
<link rel="stylesheet" href="sample1.css"
type="text/css">
</head>
<body>
<p>赤い文字</p>
</body>
</html>
```

(sample1.css)

```
p{color:#FF0000}
```

なお、一般に CSS を別ファイルに記述する方法が推奨されています。これは、複数の HTML ファイルから CSS を参照できるようにすることで、HTML 作成・保守・運用を簡単にするためです。本資料でも、CSS を別ファイルに記述する方法に的を絞って説明して行きます。

3.3 Class

これまで、指定した要素にのみ適用可能なスタイルの指定方法について説明してきました。

しかしながら CSS では、class (クラス) と呼ばれるものを用いて好きな名前スタイルを指定することが出来ます。これにより、異なる要素に対して共通のスタイルの指定、また、同じ要素に対して異なるスタイルの指定などが可能になります。まず、クラスを用いた CSS を適用する際の HTML タグの記述方法は以下のようになります。

タグ	<要素名 class="クラス名">
機能	要素にクラス名で指定されたスタイルを適用

なお、クラス名の後にスペースを空けてその他のクラス名を記述することにより、2 つのスタイルを 1 つの要素に適用することが出来ます。次に、CSS 側は、以下のような指定方法になります。

```
.クラス名 (任意に決定) {プロパティ: 値;...}
```

```
要素名.クラス名 (任意に決定) {プロパティ: 値;...}
```

ここで、前者は要素を指定していないので、どのような要素にも適用させることが可能です。また、後者は、指定された要素の一部にのみ、適用させることが可能となります。

定義したクラスを html ファイルに適用する方法は、sample38.html のとおりです。ただし、<p class="red it">のように複数のクラスを適用する場合は、クラス名とクラス名の間にスペースを入れます。

(sample38.html)

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML
4.0 Transitional//EN">
<html>
<head>
<meta http-equiv="Content-Style-type"
content="text/css">
<link rel="stylesheet" href="sample2.css"
type="text/css">
</head>
<body>
<p class="red">赤い文字</p>
<p class="red it">赤いイタリック文字</p>
<h1 class="blue">青い文字</h1>
<p class="blue">青くない文字</p>
</body>
</html>
```

(sample2.css)

```
.red{color:#FF0000}
.it{font-style:italic}
```

```
h1.blue{color:#0000FF}
```

皆さんで調べてみてください。

4 CSS の例

以下に、代表的な CSS について説明します。なお、これら以外にも様々な CSS が存在します。また、それぞれの値の指定方法についても、これ以外のものが沢山あります。それらを勉強する際には、<http://htmlhelp.com/ja/reference/css/> が良い参考ページとして役立つと思います。

4.1 CSS の例 1

タグの範囲の背景色を指定

CSS	background-color:色
機能	背景色を指定
値	数値 16進数で色を指定 色名 色名で色を指定

タグの範囲の文字色を指定

CSS	color:色
機能	文字色を指定
値	数値 16進数で色を指定 色名 色名で色を指定

文字の大きさを指定

CSS	font-size:文字の大きさ
機能	文字の大きさを指定
値	%, em, px, pt など指定
備考	%:標準文字サイズを 100%とした割合 em:親要素の文字サイズを 1 em としたときの比 px:ピクセル数で指定 pt:ポイント数で指定

文字の太さを指定

CSS	font-weight:文字の太さ
機能	文字の太さを指定
値	bold, normal

文字の斜体を指定

CSS	font-style:文字の種類
機能	斜体とするかしないかを指定
値	italic 斜体 normal 標準の文体

文字のフォントを指定

CSS	font-family:フォント名 or 文字の種類名
機能	フォント名や種類名を指定
値	文字の種類名 serif 文字の先端にはねがある. sans-serif 文字の先端にはねが無い cursive 筆記体 monospace 幅が全て同じフォント フォント名 フォント名には様々なものがあります.

行間を指定

CSS	line-height:行間
機能	複数行にわたるブロックの行間を指定
値	%, em, px などの数値

文字の表示位置を指定

CSS	text-align:表示位置
機能	文字の表示位置を指定
値	left 左揃え center 中央揃え right 右揃え justify 両端揃え (半角のみ)

(sample39.html)

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML
4.0 Transitional//EN">
<html>
<head>
<meta http-equiv="Content-Style-type"
content="text/css">
<link rel="stylesheet" href="sample3.css"
type="text/css">
</head>
<body>
<p class="sec">章用スタイル</p><br>
<p class="subsec">節用スタイル</p><br>
<p class="text">本文用スタイル本文用スタイル本
文用スタイル<br>
本文用スタイル本文用スタイル本文用スタイル
</p>
<p>スタイル適用していない本文 (上と行間を比
べてみてください) <br>
スタイル適用していない本文 (上と行間を比べ
てみてください) </p>
<h1 class="center">中央寄せ : 文字色赤 : 背景黒
</body>
</html>
```

(sample3.css)

```
.sec
{
font-size:16pt;
font-weight:bold;
font-family:"MS Pゴシック";
}
.subsec
{
font-size:14pt;
font-family:"MS Pゴシック";
}
.text
{
font-size:12pt;
```

```
font-family:"MS P明朝";
line-height:1.5em;
}

h1.center
{
    text-align:center;
    color:#FF0000;
    background-color:#000000;
}
```

基本課題 11： 自己紹介ページの背景色やフォントなどのスタイルを別途 CSS ファイルとして作成し、そのスタイルを自己紹介の HTML ページに反映させてください

4.2 CSS の例 2 【省略可】

文字の周囲の余白を指定

CSS	padding:間隔
機能	文字の大きさを指定
値	数値 % , em, px, pt など指定
備考	%:標準文字サイズを 100%とした割合 em:親要素の文字サイズを 1 em としたときの比 px:ピクセル数で指定 pt:ポイント数で指定

文字の周囲にそれぞれ異なる余白を指定

padding:a b c d	文字の上(a)右(b)下(c)左(d)の余白を指定
padding:a b c	文字の上(a)左右(b)下(c)の余白を指定
padding:a b	文字の上下(a) 左右(b)の余白を指定
padding:a	文字の上下左右(a)の余白を指定

上下左右の特定の部分に余白を指定

padding-top	文字の上の余白を指定
padding-right	文字の右の余白を指定
padding-bottom	文字の下の余白を指定
padding-left	文字の左の余白を指定

ブロックの周囲の余白を指定

CSS	margin:間隔
機能	ブロック枠の周囲への余白を指定
値	数値 % , em, px, pt など指定 auto 状況に応じて自動調整
備考	%:標準文字サイズを 100%とした割合 em:親要素の文字サイズを 1 em としたときの比 px:ピクセル数で指定 pt:ポイント数で指定 ブロックとは、テーブルのセルや、h タグによる見出しのように、ブロック状に

	囲まれた部分のことを便宜上指しています。
--	----------------------

ブロックの周囲にそれぞれ異なる余白を指定

margin:a b c d	ブロックの上(a)右(b)下(c)左(d)の余白を指定
margin:a b c	ブロックの上(a)左右(b)下(c)の余白を指定
margin:a b	ブロックの上下(a) 左右(b)の余白を指定
margin:a	ブロックの上下左右(a)の余白を指定

上下左右の特定の部分に余白を指定

margin-top	ブロックの上の余白を指定
margin-right	ブロックの右の余白を指定
margin-bottom	ブロックの下の余白を指定
margin-left	ブロックの左の余白を指定

(sample40.html)

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML
4.0 Transitional//EN">
<html>
<head>
<meta http-equiv="Content-Style-type"
content="text/css">
<link rel="stylesheet" href="sample4.css"
type="text/css">
</head>
<body>
<p class="sec">文字余白確認：章用スタイル</p>
<p class="subsec">文字余白確認：節用スタイル
</p>
<p class="sec_b">ブロック余白確認：章用スタイル</p>
<p class="subsec_b">ブロック余白確認：節用スタイル</p></body>
</html>
```

(sample4.css)

```
.sec
{
    font-size:16pt;
    font-weight:bold;
    font-family:"MS Pゴシック";
    background-color:#000000;
    color:#FF0000;

    padding:7pt;
}
.subsec
{
    font-size:14pt;
    font-family:"MS Pゴシック";
    background-color:#000000;
    color:#FF0000;

    padding:7pt 7pt 1pt 24pt;
```

```

}
.sec_b
{
    font-size:16pt;
    font-weight:bold;
    font-family:"MS Pゴシック";
    background-color:#000000;
    color:#FF0000;

    margin:0;
}
.subsec_b
{
    font-size:14pt;
    font-family:"MS Pゴシック";
    background-color:#000000;
    color:#FF0000;

    margin-top:0pt;
    margin-left:24pt;
}

```

CSS	border-color:色
機能	枠線の太さを指定
値	px, em

上下左右に、異なる枠線の色を指定

border-color:a b c d	上(a)右(b)下(c)左(d)
border-color:a b c	上(a)左右(b)下(c)
border-color:a b	上下(a)左右(b)
border-color:a	上下左右(a)

枠線の種類、太さ、色を一括指定

CSS	border:種類 太さ 色
機能	枠線の種類、太さ、色を指定
値	border-style/border-width/border-color と同じ

特定の部分に枠線の種類、太さ、色を指定

border-top	上
border-right	右
border-bottom	下
border-left	左

4.3 CSS の例 3 【省略可】

文字の周囲に枠線を表示

CSS	border-style:種類
機能	枠線の種類を指定
値	solid 枠を1本線に表示 dotted 枠を点線に表示 dashed 枠を破線に表示 double 枠を二重線に表示 groove 枠がへこんでいるように表示 ridge 枠が出っ張っているように表示 inset 枠の内側がへこんでいるように表示 outset 枠の内側が出っ張っているように表示 none 枠無し。(初期値)

上下左右に、異なる枠線を指定

border-style:a b c d	上(a)右(b)下(c)左(d)
border-style:a b c	上(a)左右(b)下(c)
border-style:a b	上下(a)左右(b)
border-style:a	上下左右(a)

枠線の太さを指定

CSS	border-width:太さ
機能	枠線の太さを指定
値	px, em

上下左右に、異なる枠線の太さを指定

border-width:a b c d	上(a)右(b)下(c)左(d)
border-width:a b c	上(a)左右(b)下(c)
border-width:a b	上下(a)左右(b)
border-width:a	上下左右(a)

枠線の色を指定

(sample41.html)

```

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML
4.0 Transitional//EN">
<html>
<head>
<meta http-equiv="Content-Style-type"
content="text/css">
<link rel="stylesheet" href="sample5.css"
type="text/css">
</head>
<body>
<span class="solid">solid</span>
<span class="dotted">dotted</span>
<span class="dashed">dashed</span>
<span class="double">double</span><br><br><br>
今までのCSSを下にまとめてみました。
<p class="solid2">border-style:solid;
border-width:1px; border-color:#666666;
padding:5px;<br>
background-color:#CCCCCC; color:#666666;
font-family:"Arial"; font-weight:bold; font-size:11pt;
line-height:1.5em;</p>
</body>
</html>

```

(sample5.css)

```

.solid{border-style:solid}

.dotted{border-style:dotted}

.dashed{border-style:dashed}

.double{border-style:double}

.solid2
{
    border-style:solid;
}

```

```
border-width:1px;
border-color:#666666;

padding:5px;

background-color:#DDDDDD;
color:#666666;

font-family:"Arial";
font-weight:bold;
font-size:11pt;
line-height:1.5em;
}
```

5 Web アクセシビリティ

WWWにおいては、世界中の人々が分け隔てなく世界中の情報資源にアクセスできるようにしようという理念があります。この事を、アクセシビリティと呼んでいます。アクセシビリティの向上のために、Webの表現方法を直接左右するHTMLの規格について、いろんな改良、変更が加えられてきました。HTMLの要素の中には、アクセシビリティについての考慮の足りなかった要素も含まれており、現在、そのような要素については非推奨という指定がされています。

5.1 非推奨な要素や属性

HTMLの仕様の中には、色の指定や、文字の装飾やサイズの指定、テキストや画像の配置など、見た目の構造を制御する要素や属性がかなりあります。WWWでは、楽しいページや目立つページ、見やすいページを目指して、そういった様々な物理的な構造の工夫をしています。

しかし、実は、そのような物理的な構造を表現する要素や属性は、現在のHTMLの規格においては、ほとんどのものが「非推奨」となっているのです。ただ、非推奨だからと言って、見た目の工夫を否定してしまったら、そのページの目的によっては意味のないものとなったり、楽しくないものになってしまう可能性もあります。

これは、物理的な構造を否定するのではなく、言語仕様の上で、HTMLは論理的な構造を表現する役割に専念し、物理的な構造については別の言語仕様で面倒をみるという、役割分担を目指した結果だと考えられます。そこで、物理的な構造を表現するために、本資料で勉強したCSS (Cascading Style Sheets) という言語仕様提案され、実際に使われています。

ただ、現実的にはまだまだ見た目を表現するための要素や属性は多く使われており、TransitionalなDTDを用いる限り、禁止されているわけではありません。扱いが容易なので、本講座でも、フォントの指定や色の指定などの見

た目の表現についてのHTML要素や属性についても学びました。

5.2 Webアクセシビリティ

Webのアクセシビリティを押し進めている活動として、Web Accessibility Initiative (WAI) が挙げられます。Webアクセシビリティについてのガイドラインの日本語訳がありますので、そのページのURLを載せておきます。

<http://www.zspc.com/documents/wcag10/index.html>

また、本資料最後の付録ページに、チェックリストを載せておくので、自分のホームページをチェックしてみてください。

さて、なぜHTMLの仕様の中から物理的な構造についての要素や属性を分離しなければならなかったかと言うと、これはアクセシビリティの問題があったからです。例えば、読み上げソフトを使ってWWWを閲覧している目の不自由な人たちもいます。文字の大きさや文字の色を直接指定してあっても、そのような人にとっては、その事自体にはあまり意味がありません。

しかし、文字の大きさを直接指定するのではなく、h1要素のように、「見出し」とであるという論理的な構造を、HTML要素として指定しておけば、読み上げソフトはそれに則してその部分が見出しであると言う事を閲覧者に伝える事ができるでしょう。そのような意味で、HTMLにおいては論理的な構造を表現する方法を確立させておき、物理的な構造の表現については別の手段を用意するという役割分担の考え方が、アクセシビリティの向上にとってとても有効であると言えます。

参考文献

[1] HTMLとスタイルシートによる最新Webサイト作成術—ホームページでなにを伝える?どう作る?, エクスナレッジ, December 2002.

付録: アクセシビリティガイドライン

1	優先度[1] 必ず配慮したい項目
2	優先度[2] できるだけ配慮したい項目
3	優先度[3] できれば配慮したい項目

番号	ガイドライン	優先度
全体		
A-1	ページの制作サイズはブラウザの横幅800 pxを超えないこと。	1
A-2	HEAD に付ける<TITLE>は内容が解る	1

	ように適切なタイトルを付けること.	
A-3	背景色・文字色等は, コントラストが十分にあり, 見やすいこと.	2
A-4	緑と赤の区別の難しい人がいることを配慮し, この色の組み合わせは極力避けること.	2
A-5	1 ページは適切な長さにし, やむを得ず長くなるときは, アンカーをつけること.	2
A-6	ページのレイアウトのためにテーブルを利用するのは極力避けること.	2
A-7	フォントサイズ・色・表示位置等の制御にはできるだけスタイルシートを使うこと.	2
A-8	色の指定には色名 (red や blue) ではなく, 数値 (#ffffff) などを利用すること.	2
A-9	ステータスバーには, できるだけメッセージを流さないこと.	2
A-10	タグの属性値は引用符 (") で囲むこと.	2
A-11	できれば Web サイトの概略を記述した英文ページを用意すること.	3
A-12	どうしてもページをアクセシブルにできない場合は, 同等の情報を持つテキストページを用意する.	2
A-13	サイト全体の構成を示すために, サイトマップや目次を提供すること.	2
A-14	サイトで使用する文章の基本となる言語を指定しておくこと.	3
A-15	レイアウト, ナビゲーション, 文章表現などは, サイト内で一貫性のあるルールで作ること.	3
画像		
B-1	画像には ALT 属性を付けること.	1
B-2	リンクボタンになっている画像の全てに ALT 属性を使ってリンク先を明確にする事.	1
B-3	意味の持たないワンポイント的画像には ALT にスペースを入れること.	1
B-4	画像は極力軽く作ることを心がけ, 指定には width, height 属性を付けること.	2
B-5	イメージマップのリンクに ALT 属性を付ける.	1
B-6	重要な画像 (図表やグラフ) には, 解説をリンクさせること.	3
B-7	文字の透過画像は, 反転させても見えるように配慮すること.	2
B-8	画像中の文字は見やすくすること.	2
B-9	点滅やスクロールなど変化する画像の	2

	使用は, コントラスト・点滅スピードに注意すること.	
リンク		
C-1	リンクの下線はむやみに消さないこと.	2
C-2	リンクするボタン画像などは, リンクである事が解るように配慮すること.	2
C-3	キーボード操作だけで適切なリンクができること.	2
C-4	「戻る」等のサイト内基本操作は, 解りやすいように同じ位置・形状にすること.	2
C-5	リンクする画像が隣接する場合は, 誤操作をしないように間を開けること.	2
C-6	リンクを張ったテキストは幅が狭すぎないこと.	2
C-7	リンクによって, 新しいウィンドウを開くことは必要最低限とすること.	2
C-8	リンクの文字は, 文章の前後関係を見なくても意味が通り, あまり長くないこと.	3
C-9	連続したリンクが 1 つのものと誤認されないように, 間に仕切り記号を入れる等配慮をする.	3
C-10	文頭等にナビゲーションを付ける場合は, 読み飛ばせるように, 本文へのアンカーを付けること.	2
音声・映像		
D-1	音声情報に対してはできるだけ文字でも情報を提供すること.	2
D-2	ビデオ情報に対しては, できるだけそれと同期した音声の情報も提供すること.	2
D-3	重要な情報を警告音等で知らせる場合は, 同等のメッセージを表示すること.	2
D-4	音声情報は, 再生・停止・音量調整などのコントロールを可能にすること.	2
テキスト		
E-1	日本語のページでは外国語の乱用はせず, 誰にでも解る様に配慮すること.	1
E-2	色によって伝えられる情報は, 色がなくても情報が伝わるようにすること.	1
E-3	略語や一般的でない言葉には, 理解しやすいように説明を付けるなどして配慮すること.	3
E-4	日付の表記には「/」を使わず, できるだけ日本語を使うこと.	3
E-5	全角と半角の使い方は, できるだけサイト内で統一すること.	3

E-6	文字サイズ・行間はスタイルシートを使用し、読みやすい物にすること.	2
E-7	見出し要素は正しい順番を守ること.	2
E-8	一単語内にスペースや強制改行を入れないこと.	2
E-9	重要な意味を持つ物には記号を用いないこと.	2
E-10	点滅したりスクロールしたりするテキストの使用は極力避けること.	2
E-11	リストは正確に構造を記述し、見栄えのためには用いないこと.	2
E-12	リストの番号表示等は音声では読み上げられないので注意すること.	2
E-13	丸数字等の機種依存文字は使わないこと.	1
E-14	PDF での情報提供は極力避けること.	2
テーブル		
F-1	表組は、行と列の関係を考慮してなるべく解りやすくすること.	2
F-2	表の始まりには「表題」を付けること.	2
F-3	行や列の見出し項目には<TH>を使って指定すること.	2
F-4	セルの結合は、必要最小限にすること.	2
F-5	レイアウトのテーブルは、情報が適切に音声ブラウザで読み上げられること.	1
フレーム		
G-1	フレームは必要最小限にすること.	1
G-2	それぞれの FRAME 要素にタイトルを付けること.	2
G-3	画面が小さくても表示できるようにスクロールバーの非表示には注意すること.	2
G-4	フレームに直接画像を入れずに別の HTML ファイルを使うこと.	2
フォーム		
H-1	submit ボタンに画像はできる限り使わないこと.	2
H-2	入力ボックス, テキストエリアにはできるだけデフォルト値を持たせること.	2
H-3	フォームによる入力をさせる場合には、できるだけ代替手段を用意すること.	3
H-4	プルダウン形式の選択メニューはキーボードからのアクセスを可能にすること.	2
JavaScript		

I-1	JavaScript には対応していない音声ブラウザを考慮して代替手段を用意すること.	2
I-2	Java アプレットには対応していない音声ブラウザを考慮して代替手段を用意すること.	2
I-3	利用者の意図に関わらず、ページの更新や移動を自動的に行わないこと.	2