



まえがき

本書は、これから新しくHTMLとCSSを学びはじめる人向けの入門書です。いま現在の、このタイミングで学びはじめるわけですから、最新バージョンのHTML5とCSS3をベースとした総合的な内容となっています。

この本の企画を進めるにあたっては、クリアすべき課題が2つありました。ひとつは、全体のボリュームの問題です。普通にHTML5とCSS3の両方を解説すると、500ページは軽く越えてしまうことが予想されました。しかし、初心者向けの本を、見ただけで「ちょっと無理」と感じてしまうほど分厚くするわけにはいきません。もうひとつは、構成上の流れの問題です。普通に考えられる順序で解説すると、先にHTMLを学習してからそれを前提としてCSSを学ぶ、という流れになり、前半はタグの意味を覚えることが中心の退屈な内容になってしまうという点です。学習を開始した早々、眠くなるような説明が延々と続いたのでは前向きに頑張る気持ちも失せてしまうかもしれません。

ひとつめの課題については、「現時点でほとんど使われていないもの、仕様が不安定なものについては掲載しない」という方針にすることでクリアしました。ご存知の方も多いかと思いますが、HTML5とCSS3はその大部分がまだ未完成であり、現在でも少しずつ変更されています。まだ正式に決まってもいない内容を入門書で事細かに解説することは、あまり意味のあることだとは思えません。入門者が覚えるべきことは、「将来そう決まるかもしれない仕様書の内容」ではなく、「今現在の実践的な制作のときに必要となる知識」であるからです。

もうひとつの課題に関しては、はじめは仕方がないと考えていました。CSSはHTMLに対して適用するものであり、どうしても先にHTMLを覚える必要があるからです。しかしあるとき、べつにHTMLのすべてを覚えてからCSSに進む必要もないと気づき、執筆の途中で構成全体を作り直しました。HTMLとCSSを少しずつ同時進行させることにしたのです。このような進め方になると、覚えた内容がすぐにブラウザでの表示結果としてあらわれるため、少しずつできるようになっていく感覚を積み重ねながら学習を進めていくことができます。ただしその分、各段階ではHTMLもCSSも限定された範囲までしか使えないことになるため、新しい構成を考える作業はまるでパズルでも解くかのような複雑な作業となることが予想されました。しかし今回はあえてそれに挑戦してみることにしたのです。その甲斐もあって、HTMLとCSSが同時進行で学べる新しいタイプの教科書が完成しました。

本書を、これからHTMLとCSSを覚える人が“楽しみながら学ぶ”ための入門書としてご活用いただけましたら幸いです。

2012年7月

大藤 幹

HTMLの役割、CSSの役割

まずはWebページを構成するHTMLとCSSについて、簡単にその概要と役割を押さえておきましょう。

▶▶ HTMLってどんなモノ？

HTMLとは、ごく簡単に言えば、テキストに**タグ**と呼ばれる印を付けて、**それぞれの部分が何であるのか**を示したテキストファイルのことです。それぞれの印は **ここから見出し** **ここまで見出し** というようなパターンであらかじめ決められていて、HTML5では約100種類あります。HTMLを学ぶということは、その約100種類の**印の意味を知り、印の付け方のルールを覚えること**だと言えます。

テキスト

こぶとりじいさん

昔々、あるところに、小太りのお爺さんがいました。
...



HTML

<ここから見出し>

こぶとりじいさん

<ここまで見出し>

<ここから本文>

昔々、あるところに、小太りのお爺さんがいました。
...

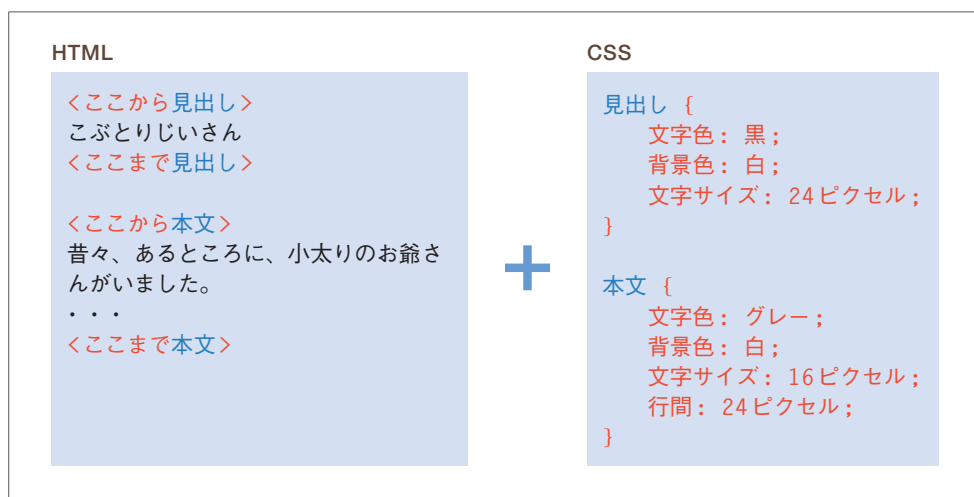
<ここまで本文>

HTMLのイメージ。テキストにこのような書式の印を付けて、テキストの各部分が何であるのかを示す

しかし、印の種類が約100種類もあるからといって気を重くする必要はありません。一般的なWebページでふだん使用されているのは、その中のごく一部だからです。すべてを暗記する必要はまったくありませんし、ふだん使うものに関しては使っているうちにすぐに覚えられますので安心してください。このあとに実際にその印を付けてみますが、あっけないほど簡単です。

▶▶ CSSってどんなモノ？

それに対して、HTMLの印によって示された**各範囲の表示方法**を指定するのがCSSです。たとえば、HTMLで見出しの印をつけた範囲の「文字色」「背景色」「文字サイズ」などを指定できます。その「文字色」や「背景色」のように表示方法として指定できる種類は、HTML5の印の種類よりかなり多く、ぜんぶだと軽く150以上はあるでしょう(まだすべての仕様が確定しているわけではありませんので最終的にどのくらいの数になるかは現時点では不明です)。ただし、CSSに関してはまだブラウザがサポートしていない機能も多くあり、現実的に使用できるのはだいたい100種類くらいであるとも言えます。



CSSのイメージ。HTMLで示された各部分の表示方法を指定する

HTMLは基本的には印をつけるだけですが、CSSは「**どの印をつけた範囲**」の「**何**」を「**どう表示させる**」といった指定をする必要がありますのでHTMLよりは少々複雑です。しかし、結局は「**どう指定すればどう表示させられるのか**」ということを知るだけです。プログラミング言語のように難しいわけではありません。CSSもこのあとに実際に使ってみますが、指定方法のパターンさえ覚えれば意外に簡単であることが分かるはずです。

HTMLのタグをつけてみよう!

いよいよここからは、手を動かしてタグ付けをやってみましょう。使用するサンプルファイルはダウンロードファイルに含まれています。

▶▶ テキストを「大見出し」と「段落」に分ける

それでは、実際にテキストに印をつけてHTMLにしてみましょう。HTMLでは印のことをタグと言いますので、以降本書ではタグという用語を使用します。タグをつけるのは、以下のテキストです。

sample/chapter-02/lecture-2-2/01.txt

かちかち山

昔々、ある会社に山田という名前の若い社長さんがおりました。社長仲間のあいだでは「やま」と呼ばれていました。山田社長には、ことあるごとに「Win-Winの考え方が大切なんです」と語りだすクセがありました。

このテキストにタグをつける

さて、このテキストを読んでみると、どうやら**タイトル**と**本文**で構成されているらしいことが分かります。1つのページ内の構成要素として考えると、**大見出し**と最初の**段落**であるとも言えそうです。ここではそれを後者であると想定して、まずは日本語で〈ここから大見出し〉〈ここまで大見出し〉〈ここから段落〉〈ここまで段落〉というタグをつけてみることにしましょう。

〈ここから大見出し〉

かちかち山

〈ここまで大見出し〉

〈ここから段落〉

昔々、ある会社に山田という名前の若い社長さんがおりました。社長仲間のあいだでは「やま」と呼ばれていました。山田社長には、ことあるごとに「Win-Winの考え方が大切なんです」と語りだすクセがありました。

〈ここまで段落〉

タグをつけてみた状態

タグについて、ページ内の2つの構成要素が何であるのかが明確に示されました。しかし、タグがつけられたテキストを見てみると、タグに含まれている「ここから」と「ここまで」は無い方がすっきりして見やすいような気がします。そこで、「ここから」はそのままカットしてしまい、「ここまで」は代わりに「/」であらわすことにして、次のように変更します。

```
<大見出し>  
かちかち山  
</大見出し>
```

```
<段落>  
昔々、ある会社に山田という名前の若い社長さんがおりました。社長仲間のあいだでは「やま」と呼ばれていました。山田社長には、ことあるごとに「Win-Winの考え方が大切なんです」と語りだすクセがありました。  
</段落>
```

「ここから」をカットし、「ここまで」を「/」に変更する

▶▶ タグを英語にする

これですっきりとして見やすくなりました。しかし、HTMLのタグは日本語ではなく英語で作られているはず。そこで、タグ内の日本語部分を英語にしてみましょう。

まず、「見出し」は「**heading**」にします。大見出しの「大」はそのまま「大」「中」「小」のようにはあらわしてもいいのですが、そうすると見出しの種類が「大見出し」「中見出し」「小見出し」の3段階に限定されてしまいます。ここではもっと多くの階層の見出しが表現できるように、大きい方の見出しから順に数字であらわすようにしてみましょう。ここでは、「大見出し」は「**heading1**」ということにします。

次は「段落」です。「段落」は英語では「**paragraph**」です。というわけで、タグの日本語部分を英語に変更すると次のようになります。

```
<heading1>  
かちかち山  
</heading1>
```

```
<paragraph>  
昔々、ある会社に山田という名前の若い社長さんがおりました。社長仲間のあいだでは「やま」と呼ばれていました。山田社長には、ことあるごとに「Win-Winの考え方が大切なんです」と語りだすクセがありました。  
</paragraph>
```

タグ内の日本語を英語に変更

CSSを使ってみよう!

次はCSSを使ってみましょう。HTMLでタグ付けをした部分に、表示の指定をしてみます。

▶▶ CSSファイルを読み込ませる

さて、文法的に正しいHTMLファイルは完成しましたが、表示方法はまだ一切指定していません。この段階でHTMLを表示させるとどのようなのかを確認してみましょう。

Lecture 2-2で完成させたファイル(sample/chapter-02/lecture-2-2/03.html)をブラウザで開いてください。ほぼ右のように表示されるはずです(ブラウザの設定によって表示結果は異なります)。



表示指定をしていない状態でのHTMLの表示

では、さっそくこれにCSSを適用してみましょう。「lecture-2-3」フォルダの中には、すでにCSSを書き込んだファイル「style.css」を用意してありますので、HTMLの中に「style.css」を読み込む指定を追加します。**CSSファイルを読み込ませる**には、以下のような指定を追加してください。

sample/chapter-02/lecture-2-3/index.html

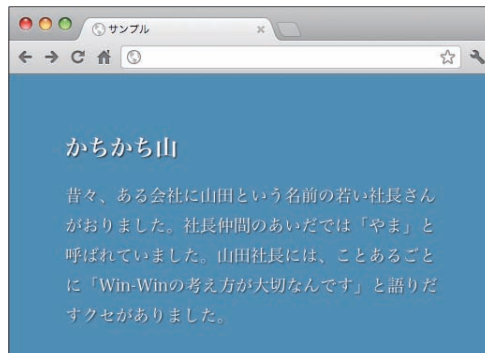
```
01 <!DOCTYPE html>
02 <html>
03 <head>
04 <meta charset="utf-8">
05 <title>サンプル</title>
06 <link rel="stylesheet" href="style.css"> ← この行を追加
07 </head>
08 <body>
09 <h1>
10 かちかち山
11 </h1>
12 <p>
13 昔々、ある会社に山田という名前の若い社長さんがおりました。社長仲間のあいだでは「やま」と呼ばれてい
   ました。山田社長には、ことあるごとに「Win-Winの考え方が大切なんです」と語りだすクセがありました。
14 </p>
```

```
15 </body>
16 </html>
```

CSSファイル「style.css」を読み込む指定を追加

「lecture-2-3」フォルダ内の「index.html」は、CSSを読み込む指定をすでに追加してありますので、それをブラウザで表示させてみましょう。

右のように表示されるはずですが(表示結果はOSの種類やそのバージョンによって多少の違いがあります)。



CSSファイルを読み込ませたあとの表示結果

▶▶ 読み込ませたCSSの内容を確認する

表示結果ががらりと変化しました。続けて、具体的にどのようなCSSを指定すればこのような表示になるのかを確認してみましょう。

テキストエディタで、「lecture-2-3」フォルダ内の「style.css」を開いてください。

では、ざっくりとですが、この指定内容を説明します。赤で示した部分は、それぞれ何に対する表示指定であるのかを示しています。

先頭にある `body` というのは、HTMLの枠組みの中にあつた `<body>` と `</body>` の範囲のことで、ひとことで言えば **ページ全体** を意味します。`h1` はもちろん `<h1>` と `</h1>` で囲った **大見出し** のことで、`p` は `<p>` と `</p>` で囲った **段落** のことです。

つまり、`body { }` の中に書かれているのは **ページ全体に対する表示指定**、`h1 { }` の中に書かれているのは **大見出しの表示指定**、

sample/chapter-02/lecture-2-3/style.css^{※4}

```
01 body {
02   margin: 60px;
03   font-family: serif;
04   background: steelblue;
05 }
06
07 h1 {
08   color: white;
09   font-size: 24px;
10   text-shadow: 1px 1px 2px black;
11 }
12
13 p {
14   color: white;
15   font-size: 18px;
16   text-shadow: 1px 1px 2px black;
17   line-height: 1.8;
18 }
```

「style.css」に書かれている全内容

※4：CSSファイルの拡張子は「.css」です(サーバーの設定で変更可能です)。

HTMLの全体構造

Chapter 2では、テキストにタグをつけたあとに、それらをページ全体の枠組みとなるタグの中に入れました。ここではまず、その枠組みの各部分について説明します。ここに書かれているタグは、すべてが文法的に必須というわけではありませんが、一般的なHTMLであれば通常はすべてが使用されます。

それでは、この枠組みの各部分を1つずつ確認していきましょう。

```
01 <!DOCTYPE html>
02 <html>
03 <head>
04 <meta charset="utf-8">
05 <title>サンプル</title>
06 </head>
07 <body>
08
09 </body>
10 </html>
```

Chapter 2で使用したHTMLのページ全体の枠組みとなるタグ

▶▶<!DOCTYPE html> | HTML5新 |

まずは先頭にある、普通のタグとはちょっと違った雰囲気タグからです。実はこれはHTMLの中では特別な役割を持ったもので、要素のタグではなく **DOCTYPE宣言** (文書型宣言) と呼ばれているものです。

HTML5より前の各バージョンのHTMLには、それぞれに **DTD** (Document Type Definition: 文書型定義) と呼ばれるファイルが用意されていました (Webの仕様を作成している **W3C** という組織のサイト内にあります)。DTDには、そのバージョンのHTMLではどの **要素・属性** が使用できるのか、各要素はどこに何回どの順序で配置できるのか、といった情報が独自の文法で記述されています。そして、そのDTDをHTMLの先頭にある **DOCTYPE宣言** で指定することで、そのHTMLファイルがどのバージョンの文法にしたがって記述されているのかを示すルールになっていたのです (そのような意味で文書型宣言という名前になっています)。

しかし、2000年頃から、各社ブラウザはこのDOCTYPE宣言を別の用途にも利用するようになっていきました。当時、HTMLで表示指定をしていたページのほとんどは文法を無視しており、DOCTYPE宣言はつけていなかったため、DOCTYPE宣言がなければ「HTMLで表示指定をしている旧式のページ」、DOCTYPE宣言があれば「標準仕様に準拠した新しいページ」という風に判定し、**ブラウザの表示モード** を自動的に切り替えるようになったのです。はじめにMac版の

Internet Explorerがその方式を採用すると他社のブラウザもそれにならない、現在ではほとんどのブラウザが同様の仕組みになっています。

しかし、HTML5からはDTDはなくなりました。もともとDTDに書かれていた情報は、HTML5の仕様書の中に専用の言語を使用せずに書き込まれることになったからです。そのため、本来の意味からすればHTML5にはDOCTYPE宣言は不要となったのです。ところが現在のほとんどのブラウザは、DOCTYPE宣言をつけないければ「HTMLで表示指定をしている旧式のページ」と認識してそのページを扱ってしまうこととなります(その場合、古いブラウザの独自仕様などに合わせた表示となるので正しい表示結果は得られなくなります)。そうなることを回避する目的で(つまりブラウザの表示モードを制御するだけの目的で)、HTML5では必要最低限のシンプルなDOCTYPE宣言をつけることとなっています。

なお、本書では `<!DOCTYPE html>` のように一部を大文字で記述していますが、DOCTYPE宣言は大文字で書いても小文字で書いてもOKです。

▶▶ html要素

DOCTYPE宣言のあとには `<html>` ~ `</html>` を配置します。それ以外の要素はすべてその中に書き込みます。

ある要素の中に直接入っている要素をその要素の**子要素**であると表現することがあります。逆に、自分を直接含んでいる要素は**親要素**ということになります。HTMLの各種要素をこのような親子関係で考えたとき、html要素はすべての要素の先祖ということになりますので**ルート要素**(the root element)とも呼ばれています。

```
01 <!DOCTYPE html>
02 <html>
03 <head>
04 <meta charset="utf-8">
05 <title>サンプル</title>
06 </head>
07 <body>
08
09 </body>
10 </html>
```

DOCTYPE宣言を除くすべての要素はhtml要素の中に入れる

テキスト関連のプロパティ

ここからテキストに対して指定できるCSSプロパティの説明をしていきます。プロパティで使用する値の説明と、実際に際して気をつけなければいけないことをまとめています。

▶▶ フォント関連プロパティ

続けて、テキストに関連するCSSのプロパティを紹介します。まずはテキスト部分のフォントを制御するプロパティ(フォントサイズ・行間・フォントの種類・太字・斜体)からです。

🔗 font-size プロパティ

Chapter 2では、**font-size プロパティ**でh1要素とp要素の**フォントサイズ**を指定しました。ここでは数値に「px」という単位をつけてピクセル数でフォントサイズを指定しましたが、実際には次のようなさまざまな値が指定できます(初期値は「medium」で、ブラウザで設定している標準フォントのサイズとなります)。

font-sizeに指定できる値

- **単位つきの実数**
フォントサイズを単位つきの実数(16pxなど)で指定します。
- **パーセンテージ**
親要素のフォントサイズに対するパーセンテージで指定します(150%のように、実数の直後に半角の「%」をつけて指定します)。
- **xx-small, x-small, small, medium, large, x-large, xx-large**
7種類のキーワードで指定できます。
xx-smallがもっとも小さく、mediumは普通のサイズで初期値、xx-largeがもっとも大きなサイズとなります(実際に表示されるサイズはブラウザによって異なります)。
- **smaller, larger**
親要素のフォントサイズに対して、一段階小さく(smaller)、または大きく(larger)します。

🍷 CSSで使用する単位について

さて、font-sizeには「**単位つきの実数**」が指定できると書かれていますが、CSSでよく使われる主な単位をここで紹介しておきましょう。

単位	説明
px	ピクセル
pt	ポイント
cm	センチメートル
mm	ミリメートル
em	その要素のfont-sizeプロパティの値を1とする単位

CSSで使用できる主な単位

Chapter 2ですでに使用しましたが、**px**は画面の1ピクセルを1とする単位です。**pt**は、ワープロなどでフォントサイズを指定するときに使われる単位のポイント(1/72インチ)と同じです。**cm**と**mm**は主に印刷用に用意されています(画面表示用のCSSでは基本的に使用されません)。

さて、最後の**em**(エムと読みます)ですが、この単位はfont-sizeプロパティ以外に使用した場合と、font-sizeプロパティに使用した場合とで意味が変わってきます。font-sizeプロパティ以外に使用した場合、この単位は**その要素に指定されているfont-sizeプロパティの値(指定されていなければ初期値mediumの大きさ)を1とする単位**となります。font-sizeプロパティに使用した場合は、**親要素のfont-sizeプロパティの値を1とする単位**となります。たとえば、親要素のfont-sizeプロパティの値が12pxの場合、「font-size: 1.5em;」と指定するとフォントサイズは18ピクセル(親要素の1.5倍)になるということです。emは比較的よく使用される単位ですので、しっかりと覚えておきましょう。

🍷 line-height プロパティ

次に紹介するのは、行間(行の高さ)を指定するプロパティの**line-height**(ラインハイトと読みます)です。これもChapter 2で使用したプロパティの1つです。次の値が指定できます(初期値は「normal」です)。

line-heightに指定できる値

- **実数**
行間を単位をつけない実数(1.5など)で指定します。
行間は、ここで指定した実数とフォントサイズを掛けた高さとなります。
- **単位つきの実数**
行間を単位つきの実数(24pxなど)で指定します。
- **パーセンテージ**
行間をフォントサイズに対するパーセンテージ(150%など)で指定します。
- **normal**
ブラウザ側で妥当だと判断する行間に設定します(ブラウザによって表示結果は異なります)。

角を丸くする | CSS3新 |

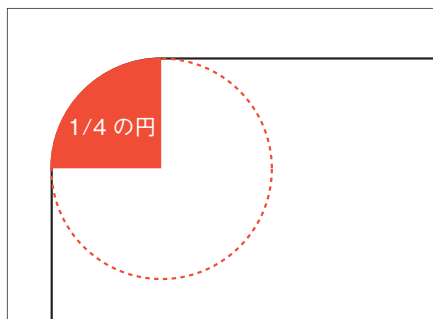
ボックスの角は、丸くすることもできます。ただし、ここまでで紹介してきたマージンやパディング、ボーダー、幅と高さなどの指定はCSS2.1でも定義されていたため基本的にどのブラウザでも使用できますが、**角を丸くするプロパティ**に関しては勧告候補(2012年7月現在)の段階のCSS3プロパティであるため、Internet Explorer 8以前のバージョンでは未対応となっている点に注意してください(それ以外のブラウザは比較的古いバージョンから対応しています)。角を丸くするプロパティには次のものが用意されており、以下に示した値が指定できます。指定する値は、角を1/4の円にみたてたときの半径です。

プロパティ名	設定対象	指定できる値の数
border-top-left-radius	左上の角丸	1
border-top-right-radius	右上の角丸	1
border-bottom-right-radius	右下の角丸	1
border-bottom-left-radius	左下の角丸	1
border-radius	上下左右の角丸	1~4

ボックスの角を丸くするプロパティ

角丸を設定するプロパティに指定できる値

- **単位つきの実数**
角の半径を単位つきの実数で指定します。
- **パーセンテージ**
角の半径をボーダー領域までの大きさに対するパーセンテージで指定します。



値には、角を1/4の円とみたてたときの半径を指定する

これらはボックスの角に関する指定ですので、これまでのように「上下左右」ではありませんが、同様の4つの角に関する個別のプロパティが用意されており、また一括指定のプロパティでは次のように値を最大4つまで指定することができます。

値の数	各値の適用場所	指定例
1	4つの角すべて	<code>border-radius: 10px;</code>
2	左上と右下 右上と左下	<code>border-radius: 10px 20px;</code>
3	左上 右上と左下 右下	<code>border-radius: 10px 20px 30px;</code>
4	左上 右上 右下 左下	<code>border-radius: 10px 20px 30px 40px;</code>

border-radius プロパティの値の指定パターン

▶▶ ブラウザの対応状況について

角を丸くするプロパティはすでに草案ではなくなっていますので、新しいブラウザだけを対象とするのであればベンダープレフィックスをつけることなく使用できます。Firefox 3.6以前やSafari 4以前にも対応させたい場合に限り、ベンダープレフィックスをつけた指定も追加してください。Internet ExplorerとOperaに関しては、草案段階を過ぎてから対応したため、ベンダープレフィックスをつけた指定には対応していません(つまり、Internet ExplorerとOpera向けのベンダープレフィックスは不要です)。

HTML sample/chapter-07/lecture-7-3/03.html

```
01 <p>角を丸くしています</p>
```

CSS sample/chapter-07/lecture-7-3/03-border-radius.css

```
01 p {
02   padding: 3em;
03   color: white;
04   background-color: silver;
05   -webkit-border-radius: 10px;
06   -moz-border-radius: 10px;
07   border-radius: 10px;
08 }
```

border-radius プロパティの指定例

複数の背景画像を指定する | CSS3改 |

CSS2.1の仕様では、1つのボックスに対して1つの背景画像しか指定できないことになっています。しかし、CSS3からは1つのボックスにいくつでも背景画像を指定できます。ただし、Google ChromeやSafari、Firefoxは少々古いバージョンでも複数の背景画像に対応していますが、Internet Explorerについてはバージョン8以前は未対応である点に注意してください。

複数の背景画像を指定する方法は簡単で、単純に値をカンマで区切って複数指定するだけです。より前に(左側)に指定している背景画像ほど上に重なって表示されます。各背景画像の表示位置や繰り返しなども指定できるように、背景関連のその他のプロパティも同様にカンマで区切って複数の値を指定できるようになっています。

次のサンプルでは、body要素に3つの背景画像を指定しています。

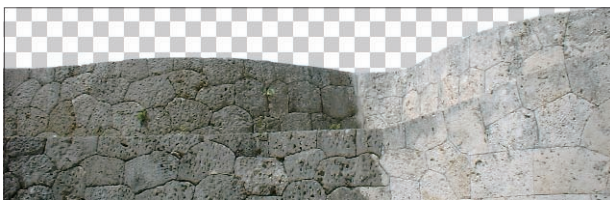
HTML sample/chapter-07/lecture-7-4/05.html

```
01 . . .
02 <body>
03
04 </body>
05 </html>
```

CSS sample/chapter-07/lecture-7-4/05-multiple-images.css

```
01 html, body { height: 100%; }
02 body {
03   margin: 0;
04   background-image: url(images/wall.png), url(images/plane.png),
    url(images/sky.jpg);
05   background-position: bottom, 50% 10%, center;
06   background-repeat: no-repeat;
07   -webkit-background-size: 100% auto, auto, cover;
08   -moz-background-size: 100% auto, auto, cover;
09   background-size: 100% auto, auto, cover;
10 }
```

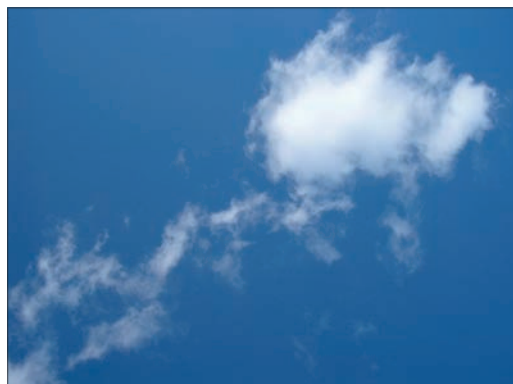
複数の背景画像を指定する例。背景関連のプロパティには、カンマで区切れば複数の値が指定できる



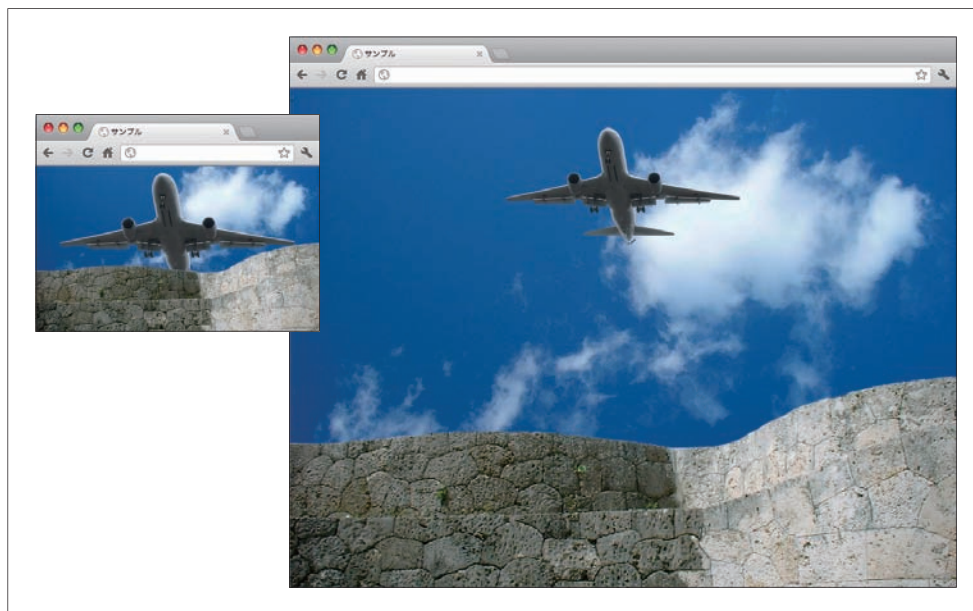
表示させている背景画像「wall.png」



表示させている背景画像「plane.png」



表示させている背景画像「sky.jpg」



各背景画像の表示位置や表示サイズを変えてあるので、ウィンドウの大きさを変えると背景もこのように変化する

背景関連プロパティの一括指定

フォント関連のプロパティの値をまとめて指定できる font プロパティのように、背景関連のプロパティの値をまとめて一度に指定できるプロパティがあります。それが **background プロパティ** です(実は Chapter 2 で背景色を指定したのはこのプロパティでした)。本書で紹介した次のプロパティの値はすべて指定できます。