

HTML マニュアル

HTML 構文とスタイルシート記述まで

入門編

1.HTML の役割と特徴

2.HTML の標準化

3.HTML 文書の全体構造

4.タグの記述

5.タグの省略

6.要素・属性・属性値

- ・開始タグに含まれるもの

7.要素の種類

- ・ブロック要素とインライン要素

8.文字参照

- ・特殊な文字をブラウザ表示する「文字参照」

9.属性値に指定できるデータ形式の種類

- ・CDATA
- ・NAMEトークン・IDトークン
- ・NUMBERトークン
- ・URI
- ・カラーの値
- ・長さ(ピクセル数または%)
- ・相対サイズ
- ・日時
- ・MIME タイプ
- ・言語コード
- ・文字コード(符号化方法)
- ・(任意の)一文字
- ・スクリプトデータ
- ・(任意の)テキスト
- ・スタイルシートデータ

10.HTML 文書の構造と DTD(文書型宣言)

- ・DTD(Document Type Declaration : 文書型宣言)
- ・ヘッダ(head 要素)
- ・ボディ(body 要素)

11.スタイルシートの役割

- ・表示方法を指定するスタイルシート
- ・スタイルシートのメリット

12.スタイル指定のルール

13.セレクタ

- ・特別な条件を示す「疑似クラス・疑似要素」
- ・条件一致のパターン

14.スタイルの記述場所と適用

- ・スタイルを宣言する場所
- ・プロパティを適用可能な要素の種類

15.要素の生成するボックス

16 プロパティの値－値の継承と inherit

17.プロパティの値－データ形式

- ・長さ(単位付き数値)
- ・カラーの値
- ・%(パーセント値)
- ・URI

1.HTML の役割と特徴

HTML(Hyper Text Markup Language)は、Web ページを作成するための記述言語で、World Wide Web の出版言語といえます。HTML タグが行うことは文章の構造化です。通常のテキスト列に対して、見出し・段落・リスト・問い合わせ先……といったそれぞれの意味(役割)を指定して、文章を構造的に組み立てます。この意味の指定をマーク付けといい、マーク付けに使うのが「`<`」「`>`」で記述された、いわゆる HTML のタグです。

意味をマーク付けする際の「意味(役割)」にあたるものが、HTML の要素(element)です。要素には、見出しや段落といったテキスト列の構造的な意味を示すものだけでなく、HTML を特徴付けているハイパーリンクの機能をもたせるリンクの始点を示すものや、画像やムービー、スクリプトなどを埋め込むものもあります。

HTML の要素には、「意味」を示すものの他に、「表示方法」を示すものもあります。こうした見た目をコントロールするものは、スタイルシートでの置き換えが勧められています。

2.HTML の標準化

HTML は徐々に進化をしてきて、マーク付けできる要素の種類を増やしてきました。その要素の種類は、HTML の標準化を行った「W3C」が、最終バージョンとして HTML4.01 の仕様を勧告しています。W3C は会員制のコンソーシアム(協会)で、そこにブラウザの開発元を含む企業や大学など多くの人が参加して議論を重ねた後に仕様としてまとめられました。

一方、マーク付けされた文章の閲覧に使うのが、Explorer や Netscape といったブラウザです。段落や見出しとしてマーク付けされた内容がどのように表示されるかは、(表示方法について仕様で言及しているものもありますが)最終的にはそれぞれのブラウザに依存します。マーク付けする要素には、このブラウザが主導で生まれたものもあり、特定の環境でしか表示できないものもあります。このような HTML が独自に発展していく動きは、「情報の共有化」や「相互で活用できる」という HTML の本来の目的を妨げる動きになっていました。そこで、プラットフォームの違いやブラウザの違いを超えて、Web ページの情報を見られるようにするために勧められたのが HTML の標準化だったのです。

ただし、過去の文章も蓄積され続ける Web 上には、W3C の仕様に準拠したものだけでなく、独自の要素を用いたものや本来の意味とは異なる使い方をした文章などさまざまなものがあるのが現状です。

3.HTML 文書の全体構造

HTML 文書の全体は次の3部分で構成されます。

- HTML のバージョン情報を含む記述(文書型宣言)
- ヘッダ部分(head 要素)→文書に関する情報
- ボディ部分(body か frameset 要素)→文書のコンテンツ本体

ヘッダ部分とボディ部分の前後が HTML の要素として区切られます(省略可能)
また、文章内ではスペースやタグなどの空白類文字、コメントはどこに記述しても構いません。

➤ HTML 文章の例

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">
<html>
  <head>
    <title> ページの名前 </title>
  </head>

  <body>
    <!-- コメント -->
    <h1> 見出し </h1>
    本文
  </body>
</html>
```

4. タグの記述

HTML のマーク付けは、要素の前後に「`<`」と「`>`」を付けたタグで行います。タグは開始タグ（`<element>`）と終了タグ（`</element>`）がセットで、このセットにはさまれたものが要素（`element`）としてマーク付けされた内容です。

たとえば、「段落」を示す要素は「`p`」ですが、これでマーク付けする場合は「`<p> 内容 </p>`」と記述します。

開始タグには、さらに属性、属性値を加えて指定するものもあります。要素、属性、属性値の関係については後述します。

5. タグの省略

マーク付けの基本は、「`<element> 内容 </element>`」ですが、開始タグ、終了タグを省略できる要素もあります。たとえば、「段落」を示す `p` や「リスト項目」を示す `li` は終了タグを省略できる要素です。

また内容をもたない要素もあり、この場合は終了タグがありません。たとえば、「改行」を示す `br` が終了タグをもたない要素です。

6.要素・属性・属性値

➤ 開始タグに含まれるもの

文書をマーク付けする開始タグには、「要素」と「属性」、そして属性に適用する値である「属性値」が含まれます。これらを使った構文は以下のようになっています。

	開始タグ		終了タグ			
	< p	align = "center"	class = "chap1"	>	内容	</p>
	要素	属性	属性値	属性	属性値	要素

➤ **要素 (element)** は、マーク付けする対象 (内容) がどのような意味をもつのかを示すもので、上の例にある「段落を意味する p や「見出し」の h1、「リンクのアンカー」の a などが 있습니다。要素は開始タグの最初にひとつだけ登場します。

➤ **属性 (attribute)** は、要素に追加して指定するもので、その要素に持たせる特性を示します。上の例で、「段落」を示す要素 p に対して追加した align 属性は「水平方向の配置」を示す属性です。

➤ **属性値 (attribute value)** は、属性のとる具体的な値で、上の例では center がそれにあたり「中央」を示します。この場合要素 p は、「align="center"」で「左右中央に配置」という特性をもったことになります。

「属性 = "属性値"」は要素名に続いて半角スペース区切りで記述します。また、属性は上の例のようにひとつの要素に対して複数指定することができます。

要素によってもてる属性は異なり、それぞれの属性がとれる属性値の種類も決まっています。また属性のなかには、その要素特有の属性と多くの要素で共通して指定できる汎用の属性とがあります。

➤ タグの記述

HTML のタグは、大文字・小文字どちらで記述しても構いません。また属性値の値は、+3、70%、など記号をとともなう値はすべて引用符 (") で括る必要があります。数字やアルファベットのみの場合は省略可能ですが、基本的には引用符で括ることで統一してください。

7.要素の種類

HTML の要素は、いくつかの種類に分類することができます。
まず大きく分けられるのは、記述の場所が限定されるもので、文書のヘッダ (head) 内、または文書のボディ (body) 内にしか記述できないものです (一部には両方に記述できるものもあります)

▶ ブロック要素とインライン要素

本体を構成するボディ内の要素を大きく2分するのが、ブロック要素とインライン要素です。

ブロック要素は、ブラウザ上の表示で前後に改行をとおもう要素で、ウインドウ内の行の左右いっぱいとその範囲となります。「段落」を示す p、「見出し」を示す h1 ~ h6、「リスト」のコンテナ(入れ物)となる ul などがその例です。

インライン要素は、新しい行を開始せずにその行内におさまる要素です。「リンクのアンカー」を示す a、「省略語」を示す abbr、「強調テキスト」を示す strong などがその例です。

ブロック要素は文書構造の枠組みを作り、インライン要素は行内テキストの修飾に使われます。

また、ブロック要素とインライン要素では、なかに内包できるものが異なります。ブロック要素は、なかにブロック要素、インライン要素の両方を含むことができます。一方、インライン要素は、インライン要素しか含むことができません。

たとえば、「段落」を示す p 要素はブロック要素で、インライン要素で「強調」を示す strong 要素を含むことができます。一方で、仮に「段落」全体を「強調」したくても、インライン要素である strong は p 要素をなかに含むことはできません。ブロック要素のなかにも、インライン要素しか含むことのできないものもあります。

種類	要素名
ブロック要素	address, blockquote, center, dir, div, dl, from, h1, h2, h3, h4, h5, h6, hr, menu, noframes, ol, p, pre, table, ul, xmp
インライン要素	a, abbr, acronym, b, basefont, big, br, button, cite, code, em, font, i, iframe, img, input, kbd, map, object, q, s, samp, script, select, small, span, strike, strong, sub, sup, textarea, tt, u, var

8.文字参照

➤ 特殊な文字をブラウザ表示する「文字参照」

HTML では、キーボードから直接入力できないような文字や、タグの記述に使うためブラウザ表示で無視されてしまう「<」「>」といった、特殊な文字を指定し、表示させる方法があります。これを「文字参照」といいます。この文字参照には、数値実体参照と文字実体参照の2種類があります。

➤ 数値実体参照

Unicode(欧文、日本語など各国の言語に対応するために制定された2バイトの文字コード体系)の文字コードを記述して参照する方法です。数値実体参照では、文字コードを「&# 」と「 ; 」ではさむ形で指定します。

数値実体参照

© → ©

➤ 文字実体参照

DTD(文書型定義)に宣言されている名前を記述する方式です。この名前は、参照する文字に名前を付けてわかりやすくしたものです。文字実体参照では、「copy」「amp」などの文字の名前を「 & 」と「 ; 」ではさむ形で指定します。

文字実体参照

© → ©

数値実体参照よりは、文字実体参照のほうが直感的に指定しやすいものになっています。以下は、文字参照の例です。ブラウザによっては表示できないものもあります。

表示	実体参照	数値参照	説明
<	<	<	小なり記号
>	>	>	大なり記号
&	&	&	アンパサンド
”	"	"	引用符
'	‘	‘	引用符(始まり)
`	’	’	引用符(終わり)
“	“	“	二重引用符(始まり)
”	”	”	二重引用符(終わり)
¢	¢	¢	セント記号
£	£	£	ポンド記号
¥	¥	¥	円記号
	¦	¦	縦線
§	§	§	セクション
©	©	©	著作権マーク
®	®	®	商標登録マーク
°	°	°	度
±	±	±	プラスマイナス
¶	¶	¶	改行記号
♠	♠	♠	スペード
♣	♣	♣	クローバー
♥	♥	♥	ハート
♦	♦	♦	ダイヤ

その他多数あり

9.属性値に指定できるデータ形式の種類

➤ CDATA

CDATA とはいわゆる文字実体を含む任意の文字列です。属性値としても要素の内容としても使われますが、属性値として指定する場合は以下のような特徴をもちます。

- ✓ 文字実体は解釈され、指定した文字に置き換えられる。
- ✓ 「<」「>」によるマーク付けは無視される
- ✓ 改行文字(改行コード)LFは無視される
- ✓ 改行文字(改行コード)CRや、タブはひとつの空白類文字に置き換える

改行文字 LF、CR は文書の改行に使われる改行コードで、UNIX では LF、Windows では LF+CR が使われています。

CDATA が要素の中身となった場合には、文字実体は解釈されずそのまま表示され、マーク付けも文字として扱われます。script や style 要素の中身 plaintext, xmp の中身が CDATA です。

```
<p>  
  <a href="../../menu.html" title="メニューページへのリンク">  
    ■back to menu page  
  </a>  
</p>
```

➤ NAMEトークン・IDトークン

NAME トークンは、名称を指定する属性の値に使われる形式です。ID トークンは、id 属性の値として使われる形式です。ともに、最初の文字と、それ以外の文字で使える文字が限定されます。

- ✓ 最初の文字→アルファベット(A~Z、a~z)
- ✓ それ以外→アルファベット(A~Z、a~z)、数字(0~9)、アンダースコア(_)、コロン(:)、ピリオド(.)

この形式を値としてとるものに、meta 要素の name 属性、http-equiv 属性などがあります。

```
<div id="chapter_1">  
  <h1>HTML 概論</h1>  
  <p>HTML とは、(中略)</p>  
</div>
```

➤ NUMBER トークン

数の指定に使われる形式です。正の整数(0~9)を1桁以上で指定します。この形式を値としてとるものに、td や th 要素の rowspan, colspan 属性などがあります。

```
<table border="1">  
<tr> <td> 左上 </td>  
      <td rowspan="2"> 真ん中 </td>  
      <td> 右上 </td>  
<tr> <td> 左下 </td> <td> 右下 </td>  
</table>
```

➤ URI

属性値のほとんどはデータ形式としては CDATA ですが、内容によってさらにいくつかの種類に分類できます。URI は、リンクさせたり埋め込み表示させる外部ファイルの場所の指定に使われる形式です。URI (Uniform Resource Identifiers) は、Web 上で参照するソースを識別するための文字列で、直接的にソースを指定する URL (Uniform Resource Locators) と名前を指定する URN (Uniform Resource Name) が含まれます。

URI は、絶対 URI、相対 URI いずれの形でも指定可能で、基本的に大文字・小文字を区別します。絶対 URI は、「http」などのスキーム名 (URI が示す情報 (サービス) の種類を表すもの) から始まるもので、インターネット全体から見た視点でファイルの場所を指定します。「http://google.co.jp/index.html」といった表記になります。

相対 URI は、「現在記述中のファイルから見た」、ほかのファイルの場所を記載する方法です。基準となる HTML 文書から見て、ファイルが「同じ階層にある」・「下の階層にある」・「上の階層にある」という形で表現します。

また、文書内の特定の箇所を URI として指定することも可能です。特定の箇所とは id 属性、name 属性を追加した要素のことで、参照先として指定する場合には、ファイルを指定する URI の後ろに「#」を付けて id 属性、name 属性の値を指定します。たとえば、同じ階層にある「menu.html」文書内で、「name="topics"」を指定された箇所の URI は、「menu.html#topics」となります。

URI を値としてとるものに、a 要素の href 属性や、img 要素の src 属性があります。

```
<p align="center">  
    
  <a href=" ../gardening.html" title="ガーデニングのページ"> Gardening </a>  
  <br>  
  <a href="http://www.google.co.jp/">検索してみる</a>  
</p>
```

➤ カラーの値

文字や背景など要素の色指定に使う形式です。指定方法は2通りあります。

✓ RGB の 16 進数値

16 進数値は、R(赤)・G(緑)・B(青)の値を、通常使用している 10 進数から 16 進数に変換したものです。

色は、RGB3色の値の組み合わせで表現されます。通常は、それぞれ 0~255 の 256 段階(または 0~100 %)で表現します。HTML で指定する 16 進数値は、この 0~255 の値を 10 進数から 16 進数(0~9、A~F の 16 段階で桁が上がる)の値に変換して、RGB の順に並べたものです。指定時には、頭に「#」をつけます。

✓ カラーネーム

カラーネームは色を名前で指定するもので、英語で表記します。W3C の仕様にあるカラーネームは、16 種類です。ブラウザの対応によっては、他に豊富なカラーネームも選択できます。

カラーの値を属性値としてとるものに、body 要素の bgcolor 属性や、font 要素の color 属性があります。

色見本 カラーネーム 16 進数値

■	Black	#000000
■	Silver	#C0C0C0
■	Gray	#808080
■	White	#FFFFFF
■	Maroon	#800000
■	Red	#FF0000
■	Purple	#800080
■	Fuchsia	#FF00FF
■	Green	#008000
■	Lime	#00FF00
■	Olive	#808000
■	Yellow	#FFFF00
■	Navy	#000080
■	Blue	#0000FF
■	Teal	#008080
■	Aqua	#00FFFF

➤ 長さ(ピクセル数または %)

サイズの指定に使う形式です。

ピクセル数(整数値)は、単位なしで指定します。%(パーセント値)は、水平方向・垂直方向の利用(表示)可能な領域に対するパーセンテージで、数値に「%」をつけて指定します。たとえば、ページ全体を領域としたときの「50%」とは、ページの半分のサイズを指し、これはウインドウサイズを変えることで可変することになります。

長さ(ピクセル数または%)を属性値としてとるものに、img 要素、object 要素などのサイズを指定する width, height 属性や、table 要素の cellpadding, cellspacing 属性などがあります。

```

```


➤ 相対サイズ

サイズ指定に使う形式です。

相対的なサイズは、複数の要素で領域を分け合うような場合に使います。「2*, 3*」という形で「*」をつけた整数をコンマ区切りで指定、領域をその*のついた整数の比率で分割します。たとえば、属性値が「2*, 3*」と指定されたとすると、全体を2:3の比率で分割します。「*」は「1*」と同じ意味です。相対サイズに合わせてピクセル数や%の指定がある場合には、先にピクセル数、%のサイズを先に割り振り、残りを相対サイズで分割します。

相対サイズを属性値としてとるものに、frameset 要素の row, cols 属性や、colgroup 要素の width 属性などがあります。

```
<frameset cols="1*, 2*" frameborder="yes">  
  <frame src="left.html">  
  <frame src="right.html">  
</frameset>
```

➤ 日時

日付と時間の指定に使われます。書式としては以下のようになります。
YYYY-MM-DDThh:mm:ssTZD

- ✓ YYYY(年)→西暦4桁で指定
- ✓ MM(月)/DD(日)→2桁で指定
- ✓ T→日付と時間の区切り文字
- ✓ hh(時)→24時間制(0~23)で指定
- ✓ mm(分)/ss(秒)→2桁で指定
- ✓ TZD→タイムゾーンは、現在時間の標準時との差を「+hh:mm」「-hh:mm」の形で指定。日本の場合は、標準時より9時間進んでいるため、「+9:00」となります。

日時を属性値としてとるものに、ins や del 要素の datetime 属性などがあります。

```
<del cite=""/del/sample.html" datetime="1984-10-19T13:30:00+9:00">  
The materials </del> has been released.
```

➤ MIME タイプ

対象とするデータの種類の指定に使います。

MIME タイプはデータ形式を表したもので、Web サーバとブラウザがやりとりするときにデータの種類を示すのに使われます。

text/html, image/jpg のように、「タイプ名/サブタイプ名」の形で表記します。text(テキスト)や image(画像)がタイプ名で、「/」(スラッシュ)の後ろの html や jpg がサブタイプ名です。Web で使用される MIME タイプで標準化されたものには以下のようなものがあります。

MIME タイプ	ファイルの種類	拡張子
テキスト		
text/plain	プレーンテキストファイル	.txt
text/html	HTML ファイル	.html .htm
text/css	CSS ファイル	.css
text/javascript	JavaScript ファイル	.js
画像		
image/gif	GIF ファイル	.gif
image/png	PNG ファイル	.png
image/jpeg	JPEG ファイル	.jpg .jpeg
動画		
video/mpeg	MPEG ファイル	.mpeg .mpg
video/quicktime	Quicktime ファイル	.qt .mov
アプリケーション		
application/pdf	Adobe PDF ファイル	.pdf
application/msword	Microsoft Word ファイル	.doc
application/x-shockwave-flash	Flash ファイル	.swf
application/zip	Zip ファイル	.zip

```
<object data="img/sample.png" type="image/png" width="176" height="220"></object>
```

➤ 言語コード

対象とする内容の言語の指定に使います。言語コードとは、fr（フランス語）、en（英語）など言語名の略称で、これに副コードをつけて「言語コード-副コード」の形で表す場合もあります。この場合の副コードは国コードとなります。たとえば「en-us」の場合、「アメリカ英語」を示します。

言語コードを属性値としてとるものに、共通属性の lang 属性があります。

言語コード	言語名
ar	アラビア語 (Arabic)
de	ドイツ語 (German)
el	ギリシャ語 (Greek)
en	英語 (English)
eo	エスペラント語 (Esperanto)
es	スペイン語 (Spanish)
fr	フランス語 (French)
it	イタリア語 (Italian)
ja	日本語 (Japanese)
ko	韓国語/朝鮮語 (Korean)
la	ラテン語 (Latin)
ms	マレー語 (Malay)
pt	ポルトガル語 (Portuguese)
ru	ロシア語 (Russian)
sw	スワヒリ語 (Swahili)
th	タイ語 (Thai)
vi	ベトナム語 (Vietnamese)
zh	中国語 (Chinese)

[back to INDEX](index.html "English menu")

➤ 文字コード(符号化方法)

文字コードの指定に使います。文字コードは、どんな文字の集合体(文字セット)で、それをどう数値コードへ対応させるかのルールを示すものです(符号化方法)。Web サーバがユーザーの要素に応じてバイト列で送り出した HTML 文書を、ブラウザがユーザーに読めるように文字列に変換するときに使われます。

日本語で使われている3種を含めて、文字コードには以下のようなものがあります。

文字コードを属性値としてとるものに、a 要素の charset 属性などがあります。

言語コード	言語名
iso-2022-jp	日本語/JIS コード
shift_jis	日本語/主に Windows や Macintosh で使用されている方式
euc-jp	日本語/主に UNIX 系 OS で使用されている方式
iso-8859-1	西欧言語の大半で使用可能な方式
iso-8859-5	キリル諸語で使用可能な方式
utf-8	Unicode の文字を使用するための方式

```
<a href="index.html" charset="iso-2022-jp"> About us </a>
```

➤ (任意の)一文字

「 a 」や「 Z 」といった一文字の指定です。世界中で使われている文字のレパートリーを定義した規格である「ISO10646」にある文字集合のなかから指定します。「©」や「&」といった文字参照の指定も可能です。

任意の一文字を属性値としてとるものに、a 要素の accesskey 属性などがあります。

```
<a href="index.html" accesskey="a" title="profile"> About us </a>
```

➤ スクリプトデータ

スクリプトを記述したデータです。属性値として指定された場合には、要素の中身になる場合と違って、文字参照が解釈されます。

スクリプトデータを属性値とするものに、onClick などのイベント属性があります。

```
<form>  
  <input type="button" onClick="location.href='http://www.google.co.jp/'" value="jump">  
</form>
```

➤ (任意の)テキスト

説明をしたり、メッセージとして表示するものなど、いわゆる「普通のテキスト列」の指定に使われます。文字参照などは CDATA としての扱いの通りです。

➤ スタイルシートデータ

スタイルを記述したデータです。属性値として指定された場合には、要素の中身になる場合と違って文字参照が解釈されます。

スタイルシートデータを属性値とするものに、共通属性の style 属性があります。

```
<p style="font-size: 12pt;" > About us <br> We are designers and ...</p>
```

10.HTML 文書の構造と DTD(文書型宣言)

HTML 文書の全体構造は、DTD(文書型宣言)と、その後の html 要素となります。html 要素のなかでは大きくヘッダ部分とボディ部分に分けられます。

➤ DTD(Document Type Declaration : 文書型宣言)

こういったルールに基づいてこの文書が書かれているかを宣言するものです。「<!DOCTYPE」に続く html はこの文書のルート要素、つまり最上位要素を示しています。PUBLIC はこれが公開された文書を参照していることを示し、その後の引用符に囲まれたものが、参照する文書の公開識別子(名称)です。HTML4.01 の仕様では、3種類の DTD が規定されています。

HTML 4.01 厳密型 DTD	「非推奨」の要素・属性とフレーム設定用の要素・属性を含まない
<code><!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN" "http://www.w3.org/TR/html4/strict.dtd"></code>	
HTML 4.01 移行型 DTD	厳密型 DTD の要素・属性に、「非推奨」の要素・属性を加えたもの
<code><!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/TR/html4/loose.dtd"></code>	
HTML 4.01 フレーム設定型 DTD	移行型 DTD の要素・属性に、フレーム設定用の要素・属性を加えたもの
<code><!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Frameset//EN" "http://www.w3.org/TR/html4/frameset.dtd"></code>	

➤ ヘッダ(head 要素)

文書に関連する情報。文書のタイトルを示す title 要素、付加情報を示す meta 情報、関連性を示す link 要素などがあります。

➤ ボディ(body 要素)

文書の内容を含む本体。body 要素または frameset 要素になります。body 内には、本体を構成するさまざまな要素が指定されます。

11.スタイルシートの役割

▶ 表示方法を指定するスタイルシート

スタイルシートは、HTML と組み合わせて使用し、文書の見たい目、表示方法について指定するしくみです。HTML が意味・役割をマーク付けをして文書の構造を定義するとブラウザはその意味に応じた表示をします。ただし、それだけでは「表示方法」そのものを細かくコントロールすることができません。その「見たい目」「表示方法」を指定するのがスタイルシート言語です。

HTML の要素のマーク付けを、特定の「表示」をさせる目的で使用すると、文書の組み立てとしては混乱します。たとえば、表組みの table 要素を配置目的で使ったり、引用を示す blockquote 要素をインデント表示のために引用文以外のものに使用するなど、マーク付けの意味とは異なる使い方は現状でもよく見られる手法です。こうした使用方法による文書の混乱と HTML の表現不足を補うのがスタイルシートの役割です。

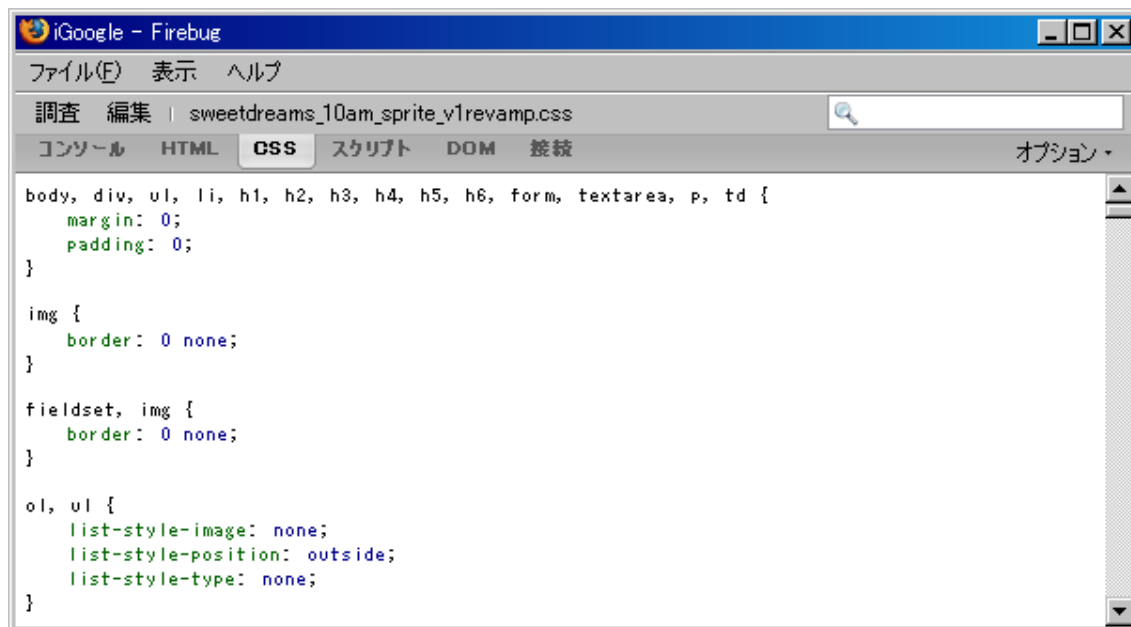
▶ スタイルシートのメリット

スタイルシートは、ワープロや DTP ソフトの機能にある「スタイル設定」と同じようなメリットがあります。スタイルの適用条件と内容を定義すると、条件に合う複数箇所に適用されます。修正する場合はもとの定義を変更すると、適用箇所すべてに反映されます。同じ文書内だけでなく、複数の文書で同じスタイルを使用することもできるため、デザイン作業を効率的に行えます。

スタイルシート言語にはいくつか種類がありますが、主要ブラウザの対応などから最も使用されているものは、CSS スタイルシート言語です。今回は W3C で勧告されている CSS2 を基に解説していきます。

12.スタイル指定のルール

スタイルシートの基本は、「body 要素の背景色を赤に」といったスタイル指定となるルールセットを記述することです。このルールセットは、適用の対象・条件とスタイルの内容の組み合わせになります。



書式としては、セクタ(スタイル適用の対象・条件)を記述し「 { }」「 []」の間に宣言ブロックと呼ばれるスタイル宣言の内容を記述します。セクタには、シンプルな要素名から、さらに複雑な条件を付け加えたものまであります。

```
selector [ ,selector] { property : value ; [ property : value ; ] }
```

「 { }」「 []」で囲まれた宣言ブロックでは「property : value ;」という形が宣言の基本になります。property(プロパティ)は、背景色、幅といったスタイル属性です。value(値)は、そのプロパティにあてはめる値です。プロパティとそれに指定できる値の種類は、仕様のなかで定められています。その他、宣言ブロックでは、以下のようなルールがあります。

- ✓ プロパティ名「:」(コロン)、値の前後には空白類文字(スペース, タブ)を自由に挿入可
- ✓ 「property : value ;」は、セミコロンまでをひと区切りとし、複数記述可能
- ✓ **値として指定するキーワードには引用符(")をつけない**
- ✓ (引用符をつけると文字列となり、キーワードとして認識されない場合があるため)
- ✓ **宣言の後ろに「!important」と記述することで最重要指定が可能**
- ✓ (文書内で同じ要素に複数の異なる指定がある場合に使用する)

13.セレクトク

スタイル適用の対象を示すセレクトクは、ある条件を備えた「要素」を指します。これはひとつ以上のシンプルセレクトクからなります。シンプルセレクトクは、要素、またはすべての要素を示す「*」の後に、属性セレクトク、ID セレクトクなどが任意に指定されたものです。

シンプルセレクトクが複数組み合わせられた場合は、最後のものがスタイルを適用される対象(主体)となります。

例)

```
p [ align ] : first-child
```

タイプセレクトク [属性セレクトク] : 擬似クラス

```
div > p [ align ] { color : red }
```

シンプルセレクトク1 > シンプルセレクトク2 { 宣言ブロック }

この場合、後のシンプルセレクトクがスタイルが適用される主体となる。

この例では、セレクトクが示すものは、「div 要素の子供要素である p 要素で align 属性をもつもの」になる

➤ 特別な条件を示す「擬似クラス・擬似要素」

擬似クラスとは、通常の要素・属性に対する指定では実現できない、特定の状況(範囲やタイミング)を分類し、スタイル適用できるようにしたものです。擬似要素は、要素・属性といった文書構造で示される以外の箇所、たとえば要素の一行目・一文字目などを適用対象にします。擬似クラスはセレクトク内のどこにでも記述できますが擬似要素はセレクトクの主体の後にしか記述できません。

条件一致のパターン

パターン	種類	説明
*	全称セクタ	すべての要素
e	タイプセクタ	e 要素
e f	子孫セクタ	e 要素の子孫要素である f 要素
e > f	子供セクタ	e 要素の子供要素である f 要素
e + f	隣接セクタ	e 要素の直接に続いている f 要素
e [att]	属性セクタ	att 属性(値は問わない)をもつ e 要素
e [att = "val"]	属性セクタ	att 属性をもち、その値が val である e 要素
e [att ~ = "val"]	属性セクタ	att 属性の属性値がスペース区切りのリストで、そのひとつに val という値をとる e 要素
e [att = "val"]	属性セクタ	att 属性の属性値がハイフン区切りの値のリストで、そのリストが val ではじまる値をとる e 要素
e.cla	クラスセクタ	cla という名前のクラス名 (class="cla") をもつ、e 要素
e#myid	ID セクタ	myid という名前の ID (id="myid") をもつ、e 要素
e:first-child	擬似クラス	親要素内で、最初の子供要素である e 要素
e:link e:visited	擬似クラス	リンク先が未訪の e 要素 リンク先が訪問済みの e 要素
e:hover e:active	擬似クラス	マウスオーバーの状態にある e 要素 マウスが押されている e 要素
e:focus	擬似クラス	ユーザーの働きかけによってフォーカスされている e 要素
e:lang(co)	擬似クラス	特定の言語(co という言語)で記述された e 要素
e:first-line	擬似要素	e 要素(ブロック要素)の最初の1行目
e:first-letter	擬似要素	e 要素(ブロック要素)の最初の1文字
e:before	擬似要素	e 要素の前
e:after	擬似要素	e 要素の後

14.スタイルの記述場所と適用

➤ スタイルを宣言する場所

HTML 文書に対してスタイルを適用するために、スタイル宣言のルールセット(セレクタ { property: value; })を記述(宣言)する方法には、次のようなパターンがあります。

- ✓ 適用する HTML 文書のヘッダ(head)内で、style 要素の中身としてルールセットを記述

```
.....  
<head>  
  <title> style sheet </title>  
  <style type="text/css">  
    <!--  
    h1 { color: red; }  
    body { background-image: url (img/01.gif) }  
    -->  
  </style>  
</head>  
.....
```

- ✓ 同じく style 要素の中身として記述し、@media を使ってメディアルール別にルールセットを作成して記述

メディアタイプ	説明
all	すべてのデバイス
screen	一般コンピュータ画面
tty	テレタイプ、ポータブル機器など固定文字幅での出力
tv	テレビ
projection	プロジェクターなど、投影による出力
handheld	モバイル機器
print	プリンタ
braille	点字出力機器
aural	音声出力機器
embossed	点字ページを出力するプリンタ

```
.....
<head>
  <title> style sheet </title>
  <style type="text/css">
    <!--
    @media screen {
      h1 { color: red; }
      body { background-image: url (img/01.gif) }
    }
    @media print {
      h1 { color: black; }
      body { background-image: url (img/02.gif) }
    }
    -->
  </style>
</head>
.....
```

- ✓ 外部のスタイルシートファイルを style 要素内で、@import ルールを使って参照

```
.....
<head>
  <title> style sheet </title>
  <style type="text/css">
    <!--
    @import url (base.css);
    h1 { color: red; }
    body { background-image: url (img/01.gif) }
    -->
  </style>
</head>
.....
```


- ✓ 適用させる要素に対して、共通属性の style 属性を追加してその属性値としてスタイル宣言をする

```

.....
<head>
</head>
<body style="background-color: red;">
  <h2 style="color: blue;">
    Chap.1 The global structure</h2>
  <p style="font-size: 10pt; color: green;">
    The body element encloses a document's content.This element can take many attribute.
  </p>
</body>
.....

```

適用の対象が style 属性を追加された要素となるため、セクタは必要なく、style 属性の属性値に宣言ブロックの中身「property: value;」だけを追加します。

➤ プロパティを適用可能な要素の種類

要素のスタイル属性を示すプロパティは、多くの場合すべての要素に対して適用可能です。ただし、そのプロパティの特性によって、適用できる要素が限られるものもあります。

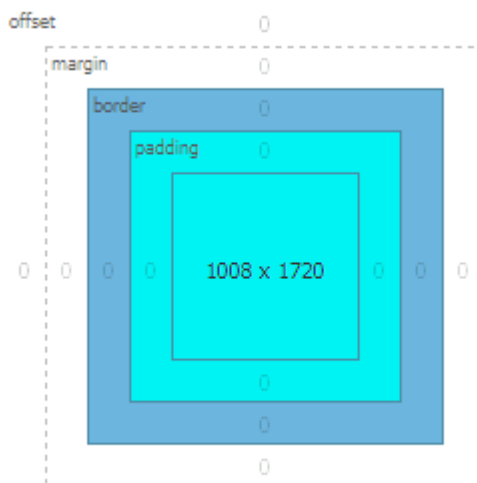
たとえば、ボディ(body)の内容となる要素にはブロック要素とインライン要素とがありますが、ブロック要素にしか適用できないプロパティがあります。また、表組に関連するプロパティなど、特定の要素にしか適用できないものもあります。

こうした、プロパティの適用可能対象を示す場合、要素の種類分類で分けられます。

要素の種類	内容
ブロック要素	ウインドウ内の行の左右いっぱいを範囲とし、文書構造の枠組みを作る要素
インライン要素	行内におさまり、行内テキストのマーク付けに使用する要素
置き換え要素	Img, object など、入力されるソースによって内容が置き換えられるもので、その寸法で認識される要素のこと。Input, textarea, select
生成された内容	content プロパティによって要素の前後に追加された内容
配置された要素	position プロパティによって配置指定された要素
リスト関連要素	ul, ol, li 要素、及び、display: list-item 指定された要素
テーブル関連要素	table, td, th, tr などのテーブル生成に関連する要素

15.要素の生成するボックス

CSS では、ブロック要素、インライン要素ともに、その要素が生成する範囲をボックスとして見立てています。この内容が入ったボックスをコンテンツエリア(内容領域)といい、スタイルを指定するベースになっています。たとえば、このボックスはその周囲に枠内余白、枠(ボーダー)、周辺余白をもつことができます。



```
1 <html>↵
2 ^   <head>↵
3 ^     <title>sample</title>↵
4 ^     <style type="text/css">↵
5 ^     <!--↵
6 ^     em {↵
7 ^         background-color: pink;↵
8 ^         padding: 15px;↵
9 ^         border: 5px black solid;↵
10 ^        margin: 20px;↵
11 ^     }↵
12 ^     -->↵
13 ^ </style>↵
14 </head>↵
15 <body>↵
16 ^   <p>これがem要素の<em> contents area </em>になります。</p>↵
17 </body>↵
18 </html>↵
```

これがem要素の

contents area

になります。

16. プロパティの値 — 値の継承と inherit

値は通常、HTML のツリー構造のなかで親要素から子要素に継承されます。たとえば子要素の `em` にスタイル指定がなければ、親要素の `p` のもつスタイル属性が `em` に継承されます。

ただし、プロパティによってはその値が継承されないものもあります。また、すべてのプロパティは「`inherit`」という値を指定できます。これは親要素と同じ値(算出値)をとることを意味します。通常は指定がなかった場合の代替値として使われる継承値を、強制的に指定するものです。

```
1 <html>↵
2   ^
3   ^   <head>↵
4   ^   ^   <title>sample</title>↵
5   ^   ^   <style type="text/css">↵
6   ^   ^   <!--↵
7   ^   ^   p {↵
8   ^   ^     font-size:10px; color:3366FF;↵
9   ^   ^   }↵
10  ^   ^   -->↵
11  ^   ^   </style>↵
12  ^   </head>↵
13  ^   <body>↵
14  ^   ^   <p>すべてのプロパティは「<em> inherit </em>」という値を指定できます。</p>↵
15  ^   </body>↵
16 </html>↵
```

すべてのプロパティは「`inherit`」という値を指定できます。

17.プロパティの値ーデータ形式

➤ 長さ(単位付き数値)

縦横のサイズの指定に使います。整数値に px(ピクセル) や、pt(ポイント)といった単位をつけた数値です。長さを指定する単位には相対・絶対の2種類があります。

単位	説明
相対指定	em 適用要素の font-size を参照 (font-size プロパティ自身にこの単位を使うときは、親要素の値を参照)
	ex 関連要素のフォントの x-height (小文字グリフの高さ=小文字「x」の高さに相当)を参照
	px ピクセル(ブラウズしているモニタなどの解像度を参照)
絶対指定	in インチ (1 in = 2.54cm)
	cm センチメートル
	mm ミリメートル
	pt ポイント (1pt = 1/72in)
	pc パイカ (1pc = 12pt)

```
p {  
  font-size: 12pt;  
  line-height: 2em;  
}
```

➤ カラーの値

文字や背景の色指定に使います。

値の指定方法は2通りで、RGB の数値、または色の名前を示したキーワードのカラーネームで指定します。RGB の数値は、4つの方法があります。カラーネームは HTML で使えるカラーネームが使えます。

値の表記	説明
#rrggbb	HTML の要素の属性値に指定する方法と同じ。RGB 各色の 16 進数値「#」に続けて指定
#rgb	この指定は、#rrggbb と同じ意味。ざら目の場合に略記できる
rgb(①)	①に、RGB 各色の 0~255 の値を、「 , 」(コンマ)区切りで指定
	①に、RGB 各色の 0%~100%の値を、「 , 」(コンマ)区切りで指定

```
body {  
  background-color: #FFCC33;  
  color: rgb ( 20%, 40%, 80% );  
}
```

➤ %(パーセント値)

縦横のサイズの指定に使います。整数値に%をつけた数値です。パーセント値は他の値を基準とした相対的で、親要素の内容領域や同じ要素の別プロパティが参照先になります。

➤ URI

画像ファイルなど、参照する Web 上のソースの指定に使います。プロパティの値として指定する場合には「url(①)」という形で、①にファイルの URI を記述します。

```
p {  
  background-image: url(sample.gif);  
}
```