



第5回 デザイン・レイアウトのポイント、ボックスモデルを理解しよう

前回まで、セレクタの種類やCSSの特性といった解説を中心に話を進めてきました。ここからは、いよいよCSSを実際に書きながら装飾やレイアウトのやり方を覚える番です。今回は、CSSファイルの作成とHTML側での参照(読み込み)の仕方、さらにいくつかのプロパティを使ってボックスモデルの説明をします。ボックスモデルは、デザインやレイアウト作業中に嫌でも利用することになるので、必ず理解しておきたいものです。
(解説:こりまさあき)

HTMLにCSSを適用するには？

装飾的なタグや属性を取り除いたHTMLに視覚的な装飾や表現をするためには、CSSを記述しなければなりません。CSSをHTMLに適用するためには、

1. HTML文書中の任意の要素にstyle属性を使ってスタイルを記述し適用する
2. HTMLのhead要素内にstyle要素を使ってスタイルを記述し適用する
3. HTMLから外部CSSファイルを参照して(もしくは読み込んで)適用する

これらのいずれかの方法が利用可能です。ここでひとつ思い出すべきことがあります。HTMLとCSSでWebページを作成する目的のひとつは「情報と装飾を分離すること」です。分離することで、装飾やレイアウトに関する指示をCSSに任せることができます。

1の場合、HTML文書内から分離・独立させた装飾指示を、再びHTML文書中にstyle属性で指定することを意味します。CSSが適用されない環境ではスタイル指定は無視されますが、作業そのものはHTML中に装飾をしているのとあまり差がありません。この状態で修正が発生した場合はHTMLファイルを1枚1枚開いて該当箇所を修正することになり、メンテナンスを考えると頭が痛くなります。

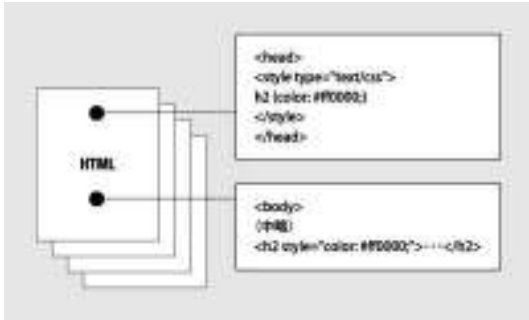
2の場合も同様、いくらHTMLのhead要素内にまとめてスタイルを記述したとしても、修正時には結局それぞれのHTMLファイルを開いて該当部分を修正しなければなりません。HTMLファイルが10枚程度であれば検索置換やコピー&ペーストでも対応できますが、100枚を超えるHTMLファイルがあった場合は、検索置換などを使った作業も現実的ではありません。

そこで3の方法の登場というわけです。

【図1】

```
【HTML】
<h2 style="color: #ff0000;">見出しのテキスト </h2>
```

```
【HTML】
<html>
<head>
(中略)
<style type="text/css">
h2 {color: #ff0000;}
/* ここにスタイルを記述していく */
</style>
</head>
<body>
(以下略)
```



任意の要素にstyle属性を使ってスタイルを記述・適用する場合と、head要素内にstyle要素を使ってスタイルを記述・適用する場合の記述例。HTML中に記述した場合、制作・管理するファイルが多ければ多いほど修正などのメンテナンス性は悪くなる

CSSは外部ファイル化して、HTML側で参照(読み込み)しよう

3の方法は、スタイルを記述するCSSファイルを外部のテキストファイルにします。このCSSファイルをHTMLに適用するには「head要素内でlink要素を使って参照する」、もしくは「style要素内で@importルールを使って読み込む」のいずれかの方法が利用できます。

CSSを外部ファイル化しておくことで、HTMLファイルが100枚だろうが1000枚だろうが、同じスタイルを適用することができるようになり、修正が発生した場合でもこのCSSファイルの修正だけで済みます。つまり、メンテナンス性が格段に向上します。また、外部ファイル化することは、実際には制作側だけの利点ではなく閲覧側にもありがたいことです。

【図2】link要素を使った参照

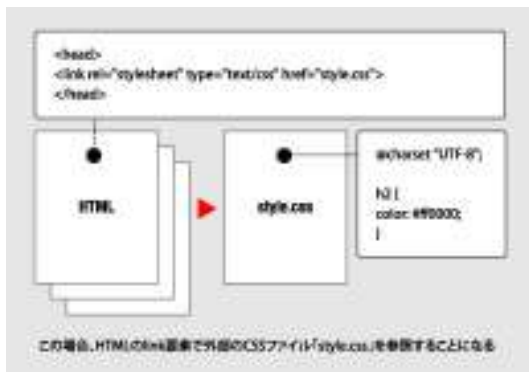
```
【HTML】
<html>
<head>
(中略)
<link rel="stylesheet" type="text/css" href="スタイルシートのファイル名">
</head>
<body>
(以下略)
```

@importルールを使った読み込み

```
【HTML】
<html>
<head>
(中略)
<style type="text/css" media="all">
@import "スタイルシートのファイル名";
もしくは
@import url("スタイルシートのファイル名");
</style>
</head>
<body>
(以下略)
```

外部にCSSファイルを用意して参照(もしくは読み込み)する場合はこのように記述するのが基本になる。@importルールに関しては、どちらでも読み込み可能だ。ただし、@importルールの記述の仕方には引用符の有無の差など、実際にはいくつかの記述方法がある。これは@importルールを特定の書式にすることで、CSSが適用されない古いブラウザもあるためにCSSの適用の切り替えなどに用いられることが多い。市販の解説書やセミナーなどでは、link要素で参照したCSSファイル内で@importルールを使ってCSSファイルを読み込むようなやり方を説明されていることもある

【図3】



外部CSSファイルを用意し、それをHTMLから参照（もしくは読み込み）すれば、HTMLの枚数が多くても修正はCSSファイルだけで済むという利点が生まれる

【図4】



CSSファイルは、単純なテキストファイルなので「ファイル名.css」のように、cssの拡張子をつけて用意すればよい。CSSファイルの冒頭には、「@charset "UTF-8";」のように使用している文字コードを指定するのが一般的な作法だ

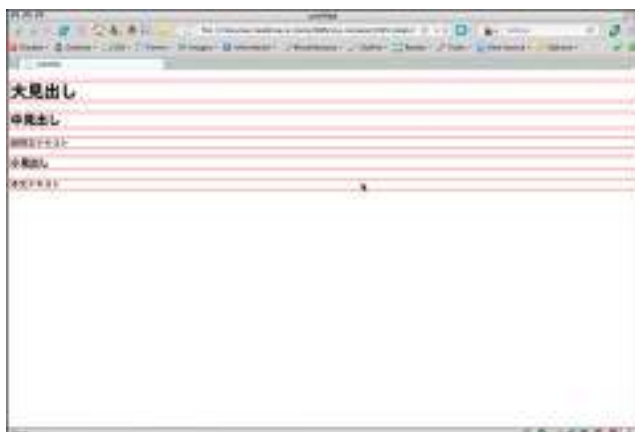
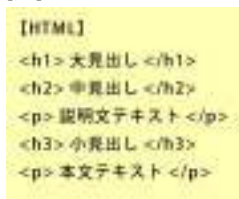
装飾やレイアウトの前にボックスモデルを理解しよう

CSSファイルの準備ができたら、HTMLファイルを用意してhead要素内で参照、または読み込んでおきましょう。早速HTMLを書きながらセレクタを作って装飾やレイアウトを……、といきたいところですが、その前にHTMLとCSSでWebページをデザインする前に覚えておきたいことがあります。それは「ボックスモデル」というものです。

HTML中に記述する<h1>や<h2>、<p>といった要素は、それだけで「箱」を形作ります。HTMLで各要素を使って情報構造をマークアップしたものをブラウザで見ると、それらの要素がHTMLの上から下へ順番に表示されます。これはそれぞれの要素が「ブロックレベル要素」であるため、ブラウザウィンドウ内の横幅一杯に表示され、連続して積み重なっている状態なのです。

CSSを使ってデザインやレイアウトを進めていくためには、このような縦に繋がった状態ではうまくいきません。そこで各ブロックの幅や高さを調整したり、隣り合う要素同士の間隔や箱と内容の間に余白を入れて調整したりする作業が必要になります。

【図5】



上のHTMLをブラウザで開いた場合、それぞれの要素内に含まれるテキストが縦に連続して表示されるはずだ。実際には、図のようにブラウザウィンドウ一杯に要素のブロックが広がっている状態なのだ

ボックスモデルを理解するには、まずこの「箱」の構造を確認しておきましょう。＜h1＞のような要素にはテキストなどの内容物が含まれます。内容物の大きさは幅(widthプロパティ)と高さ(heightプロパティ)で表されます。箱にはそれに加えて、箱全体を取り囲む目に見えない外枠とも言える境界線(borderプロパティ)が存在しています。内容物とこのボーダーの間には余白(paddingプロパティ)、そして隣り合う箱同士にはマージン(marginプロパティ)という名の間隔があるのです。

これを人に例えるなら、身体の基本的な大きさは骨格みたいなもので「width」、「height」で表せるでしょう。もちろん骨だけでは成立しませんので、皮膚が「border」にあたります。その骨と皮膚の間には肉があり、それが「padding」といえます。最後に他の人と接する側にはまるでオーラのような間隔が存在し、これが「margin」と言い表せるでしょう。

【図6】



ボックスモデルを図で示すと、このようになる。要素の大きさは、単純な内容物の大きさ(widthとheight)だけでなく、それを取り囲むマージンとボーダー、余白も含めた大きさになるので注意しよう。ブラウザには、あらかじめ要素によってこれらのプロパティが設定されていることも併せて覚えておこう

HTML中の要素をCSSでデザイン・レイアウトするためには、それぞれの要素がボックスモデルに基づいて形作られていることを理解しておきましょう。幅と高さ、ボーダーと余白、マージンの関係は絶えずついて回るので、どこがどの部分かはしっかりと覚えてください。さあ、ここからがよいよ本番です！ 次回は、色に関するプロパティを説明します。

Point

- ・CSSはHTML中に記述しないで外部ファイル化しておこう
- ・デザインやレイアウトに欠かせない「ボックスモデル」は必ず覚えよう