第3回　覚えておきたい、その他のセレクタ

前回は基本とも言える3種類のセレクタ、「タイプセレクタ」、「idセレクタ」、「classセレクタ」について説明しました。今回はそれに加えて覚えておきたい、その他のセレクタを紹介します。基本のセレクタだけを使ったスタイル指定では、同じような指定が重複することもあり、結果としてCSS全体の見通しが悪くなりがち。今回紹介するセレクタをうまく使いこなせるかどうかが、シンプル＆クリーンなCSSファイルを作れるかの大きなポイントです。ぜひ、頭の中に入れておいてください。（解説：こもりまさあき）

リンク疑似クラスとダイナミックリンク疑似クラスを覚えておこう

「セレクタ」という名称こそ付いていませんが、基本形のセレクタと合わせて覚えておきたいものとして「リンク疑似クラス」と「ダイナミックリンク疑似クラス」があります。従来のHTMLでは「＜body link=″色指定″ vlink=″色指定″＞」のようにリンクの状態を指定していたものは、CSSでa要素のリンク疑似クラスを利用し「a:link { … }」「a:visited { … }」と指定します。つまり、リンクテキストの前景色や下線などの装飾指定は、このa要素のセレクタとリンク疑似クラスを利用するというわけです。

【CSS】

```
a {
 text-decoration: underline;
}
a:link {
 color: #cc0000;
}
a:visited {
 color: #003366;
}
```

【1】HTMLでは、リンク指定にはa要素を使う。まず、タイプセレクタの「a { … }」に対してリンクの下線のみを指定している。ここにリンク疑似クラスである「a:link { … }」と「a:visited { … }」を指定することで、リンクカラーと訪問済みの前景色を、それぞれ適用することが可能だ

Webページのデザインによっては、マウスの状態に合わせてテキストのカラーを変更したいこともあります。その場合はリンク疑似クラスだけでなく、ダイナミックリンク疑似クラスと呼ばれる「:hover」「:active」を組み合わせましょう。ダイナミックリンク疑似クラスは、マウスの状態（:hoverはマウスオーバー時、:activeはクリック時）に応じてスタイルを変化させることが可能です。

「:active」や「:hover」などダイナミックリンク疑似クラスは、IE 6までa要素にしか適用されないというブラウザの仕様上の制限があります。そのため、今現在ではa要素のみで利用される機会が多いものですが、本来はa以外の要素でもマウスの状態にあわせてスタイルの切り替えが可能です（*1）。

*1: a要素に対してリンク疑似クラスとダイナミックリンク疑似クラスを指定する場合は記述順に注意しましょう。「:link」、「:visited」、「:hover」、「:active」の順番で記述しないと動作しないブラウザも存在します。

【CSS】

```
a:hover {
 color: #ffcc00;
}
a:active {
 color: #000000;
}
```

【2】a要素に対してダイナミックリンク疑似クラスを追加する。「a:hover { … }」はリンクテキストにマウスオーバーの状態、「a:active { … }」はマウスクリック時の状態をスタイル指定する。このサンプルではリンクのa要素に対して指定しているが、本来a要素以外にも指定可能だ。

子孫セレクタでスタイルをピンポイント指定

ここからは、基本セレクタの応用ともいえるセレクタを紹介します。前回、Webページ中の特定の要素に同じスタイルを指

定するならタイプセレクタ、特定の場所を指定するときはidセレクタ、共通化したいときはclassセレクタと覚えていました。しかし、HTML中のさまざまな要素にすべてid属性やclass属性を指定していくのは大変ですし、装飾の用途のためだけの属性指定が増えるのは、シンプルになったはずのHTMLがまた混乱してしまいます（*2）。HTML文書のことも考えて、もう少し効率的にスタイル指定をするなら「子孫（しそん）セレクタ」を覚えておきましょう。

子孫セレクタは、HTMLの文書構造に従って親要素から子要素へと段階的に要素を記述して要素を特定します。例えば、id属性「contents」のdiv要素に含まれるh2要素のみに同じスタイルを適用するなら、「div#contents h2 { … }」というセレクタを指定すれば、そのdiv要素内に含まれるh2要素のみにスタイルが適用されます。h2要素にclassセレクタを指定しておけば同様の装飾はできますが、h2要素の数が増えればHTMLソースのデータ量はどんどん大きくなっていきます。idセレクタとclassセレクタ、そして子孫セレクタ、いずれを使った方が効率的かHTMLの文書構造と相談して決めましょう。

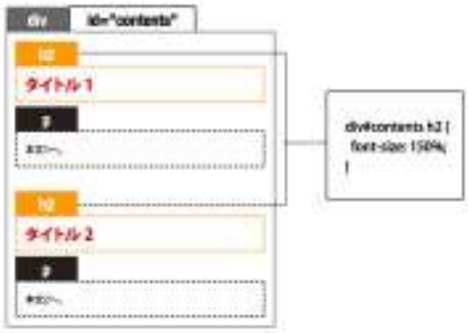
*2: 装飾的なタグや属性を取り除いてシンプルなHTML文書にしたとしても、id属性やclass属性が必要以上に増えればデータ量は変わらなくなってしまいます。

[HTML]

```
<div id="contents">
 <h2> タイトル 1</h2>
 <p> 本文 1</p>
 <h2> タイトル 2</h2>
 <p> 本文 2</p>
 <h2> タイトル 3</h2>
 <p> 本文 3</p>
</div>
```

[CSS]

```
div#contents h2 {
 font-size: 150%;
}
```



【3】HTML中の特定のブロック内で同じ要素が連続する場合、特別な意味を持たない限り子孫セレクタを使った方がHTMLソースは煩雑にならずに済む

グループセレクタでスタイルをまとめて指定

ここまで覚えたセレクタを使えば、もうWebページをひと通り作っていくことは可能です。しかし、Webページの中でいくつもの同じ装飾指定が頻発することは決して珍しくありませんし、HTMLの構造上は＜h2＞と＜h3＞の異なるレベルの見出しであっても、配置位置によっては同じスタイルを適用したいことも考えられます。そこで要素ごとに個別のセレクタを作って同じ装飾指定をしていては、お世辞にも効率が良いとはいえません。

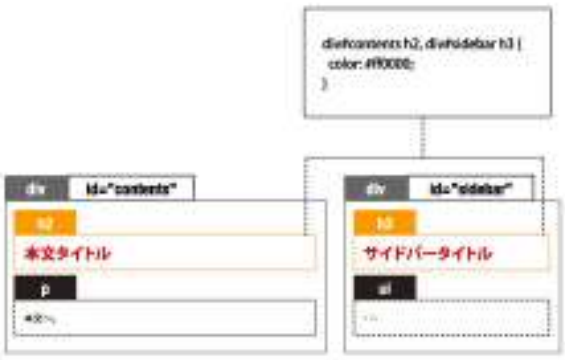
そこで、スタイルが共通化できるものをひとまとめにするために、「グループセレクタ」というセレクタを利用することが可能です。個々のセレクタを「,（カンマ）」区切りで列挙して、グループセレクタとして同じスタイルを記述し、簡略化することができます。例えHTML文書中で異なる要素が割り当てられたとしても、それぞれに同じ装飾を指定するならば、何度も同じスタイルを記述するよりも、グループ化してまとめてしまって記述した方が効率的です。

[HTML]

```
<div id="contents">
 <h2> 本文タイトル </h2>
 <p> 本文 </p>
</div>
<div id="sidebar">
 <h3> サイドバータイトル </h3>
 <ul>
 <li> … </li>
 </ul>
</div>
```

[CSS]

```
div#contents h2, div#sidebar h3 {
 color: #ff0000;
}
```



【4】HTML文書の構造上は異なる要素であっても、Webページの見た目上は同じ装飾を適用したい場合もある。個々の要素に対して同じスタイル指定を個別に記述するよりは、グループセレクタを使ってひとまとめにした方がスマートだ

適材適所でセレクタは使い分けよう

前回紹介した基本的なセレクタだけでは、無駄にidセレクタやclassセレクタが増えてしまいCSSファイルそのものが冗長になりがちです。慣れないうちはどのセレクタを使うべきか判断に迷いがちですが、今回後半紹介した子孫セレクタとグループセレクタはWebページをスマートに装飾するうえで欠かせないものですので、せひ覚えておきましょう。慣れてきたら、子孫セレクタやグループセレクタで代用できないか考えてみると良いでしょう。

Point

・リンクのa要素の装飾には「リンク疑似クラス」を利用する
・idセレクタやclassセレクタを多用せず、子孫セレクタもうまく利用する
・同じスタイルは何度も指定するより、グループセレクタでひとまとめに

次回は、さらにCSSを極めるために必要なCSSの特性について解説をします。