

**border: [罫線]**

borderは要素に罫線をつけるプロパティです。

デフォルトではnoneとなっているので何らかの形状を入れると表示されます。

一般的な実線の罫線はsolidで表示できます。

(1.html)

```
<style>
<!--
h4{
color:red;
text-align:center;
border:solid;
}
```

```
none
dotted
dashed
solid
double
groove
ridge
inset
outset
```

他にも線の種類には次のようなものがあります。

(単独でborder-style:で指定することも出来ます。)

**CSSの基本書式**

プロパティ:値

**一つのプロパティに複数の値をつける**

borderには形状以外にも色・太さなど複数の値がつけられます。

一つのプロパティに複数の値を付ける場合、それぞれの値の間を半角スペースで区切ります。

```
h4{
color:red;
text-align:center;
border:solid 1px black;
}
```

プロパティ:値

(それぞれ単独にborder-size:、border-color:で指定することも出来ます。)

**課題**

1.htmlに指定したH4の扱いでdiaryのすべてのページを統一してみましょう。

**1. クラス/IDセレクタ**

これまでスタイルシートの定義の対象=セレクタにはタグを対象にしてきましたが、class/IDセレクタを使うことで独自のタグを定義するように文章中のあらゆる部分に名前をつけてスタイル指定が行えるようになります。

**class/ID属性**

タグにはclass属性、ID属性を用いて名前をつけて同じタグの中で識別させることが出来ます。

**<TAG class="クラス名">**

**<TAG id="ID名">**

classは文章中に何度も使われる部分の名称をつけるのに使われ、

IDは同じものが他にない一度だけ現れるものを区別するために用いられます。

このclassとIDはセレクタに用いて、それぞれ毎にスタイル指定することができます。

**TAG.クラス名 {属性:値}**

**TAG#ID名 {属性:値}**

classはclass名の前に、(ドット)を、idはid名の前に#をつけます。

同じタグの中で見栄えを区別したい場合、インラインスタイルシートで直接記述することが出来ました。

しかしインラインスタイルシートではソースが複雑になってしまい、かつその見栄えが複数ある場合同じ指定が複数で必要になってしまいます。

そうした場合にもこのclass,idという識別子をつけて用いるとセレクタの同一タグを区別して別の指定をすることが出来ます。

```
<body>
<p style="color:red;font-size:20px">おはようございます</p>
こんにちは<br>
<p class="bye">さようなら</p>
<p>またあした</p>
</body>
</html>
```

```
<head>
<title>クラス/IDセレクタ</title>
```

```
<style>
<!--
p{
color:blue;
font-size:36px;
}
p.bye{
color:green;
font-size:24px;
}
```

```
-->
</style>
</head>
```

class.html  
(saiteigi.htmlから複製)

さようなら部分の<p>にclass属性でbyeと名前をつけて

他の<p>と識別しました。

p.byeは名前がついて区別されているので他の<p>とは異なる指定をすることが出来ます。

## 2. タグと無関係にスタイルを指定

このクラスとIDはタグ無しで指定することもできます。

**クラス名 {属性:値}**  
**#ID名 {属性:値}**

この方式で用いればタグの構造とは無関係に好きな部分を好きなようにスタイル指定することができます。

タグには元々何らかの構造定義の意味があり、たとえば「公式」のように元々タグに用意されていないものを無理やり<見出し4>のように定義させると元々の表示効果とのバッティングなど様々な不都合があります。そこでこうした場合セレクトにタグを使用せずclassとidだけで指定すれば、タグで構造定義を行うのとは無関係に特定のエリアに名前をつけて識別させ、そこにスタイル指定をすることが出来ます。またHTMLの構造大義は見出しなど主に短い段落ベースのものが主だったのですが、見出しと公式を含めた部分をサンプルという扱いにして、他と区切って指定するなどこれを活用することによってレイアウトの自由度が高まります。

ただしエリアを識別するにはHTMLである限り何らかのタグが必要です。そこで、使われるのが固有の表示効果のないタグ<div>、<span>タグです。

### <div>タグと<span>タグ

タグは元々文意を表すもので何らかの表示効果を持っています。しかし、全く特別な表示効果を持たないタグがこのふたつのタグです。

<div> 囲まれたエリアに改行だけが入る。 ブロック単位でエリア指定する場合  
 <span> 改行も入らず全く表示効果が無い 1行の中の一部を指定する場合

つまり、これらのふたつのタグは、class,ID属性と組み合わせることで文章上の任意のエリアやブロックを識別させることができ、それに対してスタイル指定を行う為に用いられるのです。

**<div (span) class="クラス名">~</div (span)>**  
**<div (span) id=" ID名">~</div (span)>**

divspan.html

そこで固有の表示効果のないspanタグ、divタグを使用する。  
 <span>spanタグ</span>で囲んでも、何ら表示効果は現れない。  
 <div>divタグで囲むと改行だけが行われる。</div>

spanタグをつけてもなにも変わらず  
 divタグを入れると改行します。

タグには何らかの表示効果があるが、スタイルを設定する場合、それが煩わしく感じられる。  
 そこで固有の表示効果のないspanタグ、divタグを使用する。 spanタグで囲んでも、何ら表示効果は現れない。  
 divタグで囲むと改行だけが行われる。

<body>  
 タグには何らかの表示効果があるが、スタイルを設定する場合、それが煩わしく感じられる。<br>  
 そこで固有の表示効果のないspanタグ、divタグを使用する。  
 <span class="dv">spanタグ</span>で囲んでも、何ら表示効果は現れない。  
 <div class="dv">divタグで囲むと改行だけが行われる。</div>  
 これら二つのタグにclass/ID属性でクラス名/ID名を指定する。  
 </body>

```
<head>
<title>表示効果のないタグ</title>
<style>
<!--
.dv{
color:red;
font-size:24px;
border:solid;
}
-->
</style>
</head>
```

両方のタグにクラス名.dvをつけてスタイル指定を行います。  
 タグに関係なく同じ指定が行えます。

タグには何らかの表示効果があるが、スタイルを設定する場合、それが煩わしく感じられる。  
 そこで固有の表示効果のないspanタグ、divタグを使用する。 spanタグで囲んでも、何ら表示効果は現れない。  
 divタグで囲むと改行だけが行われる。  
 これら二つのタグにclass/ID属性でクラス名/ID名を指定する。

class.html

```
<body>
<p style="color:red;font-size:20px">おはようございます</p>
<div class="eve">こんにちは</div>
<p class="bye">さようなら</p>
<p>またあした</p>
さようなら部分のpタグにbyeという名前をつけて識別し緑で24pxに指定し
<div class="eve">こんにちはの部分をdivタグで識別し指定した</div>
</body>
```

```
<style>
<!--
p{
color:blue;
font-size:36px;
}
p.bye{
color:green;
font-size:24px;
}
.eve{
color:gold;
font-size:36px;
}
-->
```