

2.6.CSS - 變形、轉換與動畫

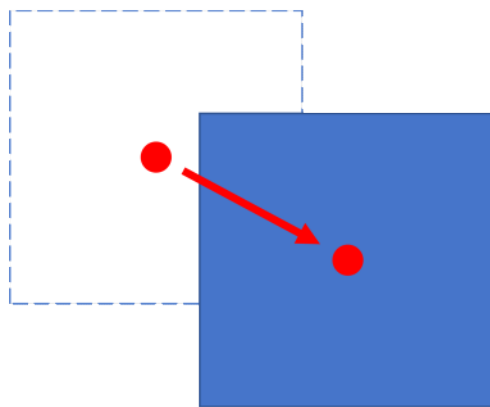
一、變形效果：transform

transform 是 CSS3 新增的元素屬性，可以設定元素的變形效果，分為移動、縮放、旋轉及傾斜等，格式如下：

transform：變形效果

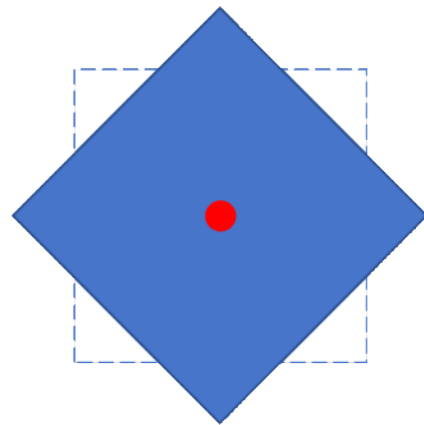
變形效果	說明
translate(x, y)	移動，預設以元素的中央為基準，x、y 為在二軸上的移動距離，單位預設為 px
translateX(x)	在 X 軸上移動，以元素的中央為基準，x 是移動距離
translateY(y)	在 Y 軸上移動，以元素的中央為基準，y 是移動距離
scale(x, y)	縮放，預設以元素的中央為基準，x、y 為在二軸上的縮放倍數
scaleX(x)	在 X 軸上縮放，以元素的中央為基準，x 是縮放倍數
scaleY(y)	在 Y 軸上縮放，以元素的中央為基準，y 是縮放倍數
rotate(deg)	旋轉，預設以元素的中央為基準，以右方為起始點進行順時針的旋轉，deg為旋轉的角度
skew(x, y)	傾斜，預設以元素的中央為基準進行順時針的傾斜，x、y 為在二軸上的傾斜角度
skewX(x)	以元素的中央為基準，在 X 軸上進行順時針的傾斜，x 是傾斜角度
skewY(y)	以元素的中央為基準，在 Y 軸上進行順時針的傾斜，y 是傾斜角度
none	取消變形

移動：translate()、translateX()、translateY()



translate(20px, 10px);

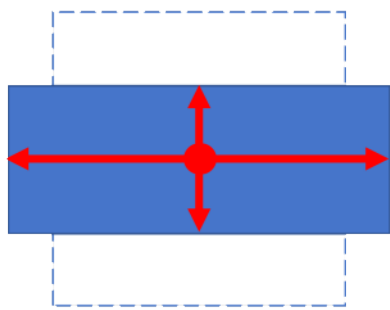
旋轉：rotate()



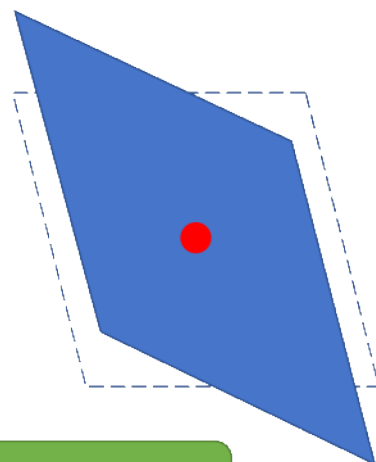
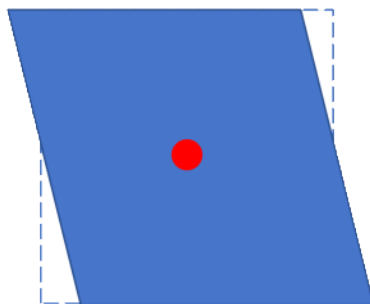
rotate(45deg);

縮放：scale()、scaleX()、scaleY()

傾斜：skew()、skewX()、skewY()



```
scale(1.5, 0.5);
```



```
skew(20deg, 30deg);
```

範例：

Ex:

```

<style>
div{
  float:left;
  width: 100px;
  height: 100px;
  margin: 45px;
  font: bold 50px/100px arial;
  text-align: center;
  background-color: ■lightblue;
  border: 1px solid □black;
}
.box1 {
  transform: translate(-20px, 10px);
}
.box2 {
  transform: rotate(45deg);
}
.box3{
  transform: scale(0.5, 1.5);
}
.box4{
  transform: skew(-20deg, -30deg);
}
</style>

<div class="box1">1</div>
<div class="box2">2</div>
<div class="box3">3</div>
<div class="box4">4</div>

```

二、改變變形的基準點：transform-origin

元素的變形效果預設以元素範圍中心為基準點，可以使用 transform-origin 改變基準點。表示方法可以使用數值、百分比或是「left」、「center」等名稱，格式如下：

transform-origin：水平位置 垂直位置

範例：

Ex:

```

<style>
div{
  width: 200px;
  height: 200px;
  background-color: ■lightblue;
  border: 1px dotted □black;
  position: absolute;

```

```

<style>
div{
  width: 200px;
  height: 200px;
  background-color: lightblue;
  border: 1px dotted black;
  position: absolute;
  top: 50px; left: 300px;
  transform-origin: 0% 0%;
}
.box1 {
  transform: rotate(0deg);
}
.box2 {
  transform: rotate(20deg);
}
</style>

<div class="box1"></div>
<div class="box2"></div>

```

三、轉換效果：transition

CSS3 也可以使用 transition 屬性設定元素的轉換效果，轉換是指由一種樣式轉為另一種樣式，格式有四種屬性：「transition-duration」是設定兩個效果之間的轉換時間，單位為秒 (s)，預設為 0 秒；「transition-property」是設定兩個效果之間改變的屬性；「transition-timing-function」是指轉換速度，預設為「ease」，即開始與結束皆平滑的轉換；「transition-delay」是指開始轉換前等待的時間，單位為秒 (s)，預設為 0 秒，格式如下：

transition-duration：轉換時間

transition-property：轉換屬性

transition-timing-function：轉換方法

transition-delay：延遲時間

轉換方法	說明
ease	預設值，平滑的開始，平滑的結束
linear	開始到結束保持平均的速度
ease-in	開始到結束由慢到快的轉換
ease-out	開始到結束由快到慢的轉換
ease-in-out	開始到結束由慢到快再到慢的轉換

範例：

Ex:

```

<style>
.box {
  width: 100px;
  height: 100px;
  background-color: ■lightblue;
  border: 1px dotted □black;
  text-align: center;
  line-height: 100px;
  transition-duration: 1s;
  transition-property: width;
  transition-timing-function: ease-in-out;
  transition-delay: 0s;
}
.box:hover {
  width: 400px;
}
</style>

<div class="box">BOX</div>

```

四、整合設定轉換效果：transition

可以使用 transition 屬性來簡化語法，格式如下：

transition：轉換時間 轉換屬性 轉換方法 延遲時間

範例：

Ex:

```

<style>
.box {
  width: 100px;
  height: 100px;
  background-color: ■lightblue;
  border: 1px dotted □black;
  text-align: center;
  line-height: 100px;
  transition: 1s width ease 0s;
}
.box:hover {
  width: 400px;
}
</style>

<div class="box">BOX</div>

```

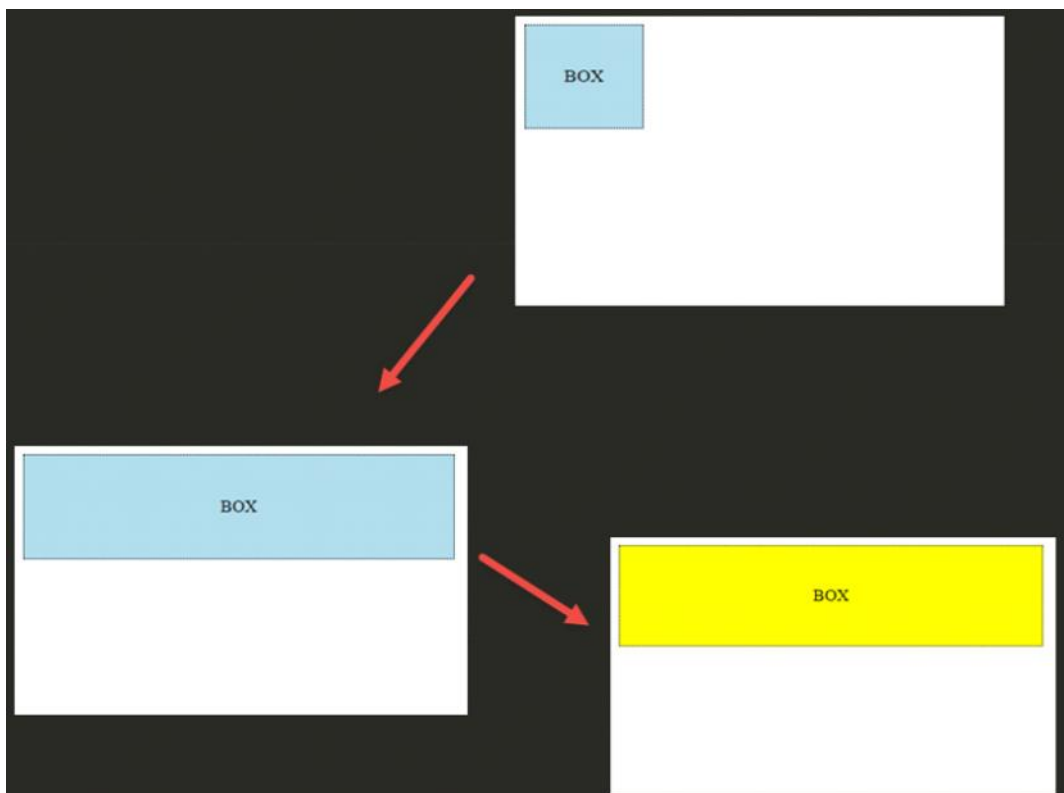
五、分段執行轉換效果

如果要分段執行多個不同的屬性轉換效果，可以利用「,」

範例：

Ex:

```
<style>
.box {
  width: 100px;
  height: 100px;
  background-color: lightblue;
  border: 1px dotted black;
  text-align: center;
  line-height: 100px;
  transition: 1s width ease 0s,
             1s background-color ease 1s;
}
.box:hover {
  width: 400px;
  background-color: yellow;
}
</style>
```



六、動畫效果：animation

CSS3 可以使用 animation 屬性設定元素的動畫效果，動畫效果就是由播放的起點到終點設定多個關鍵影格，接著指定元素在每個關鍵影格中屬性不同變化所呈現的動畫。設定動畫前必須先設定好關鍵影格，格式如下：

```
@keyframes 動畫自訂名稱 {
  from { CSS 設定; }
  to   { CSS 設定; }
}
```

```
@keyframes myanimation{
  from { top: 0px;}
  to   { top: 300px;}
}
```

動畫關鍵影格的進階設定

```
@keyframes 動畫自訂名稱 {
  百分比1 { CSS 設定; }
  百分比2 { CSS 設定; }
  ...
}
```

```
@keyframes myanimation{
  0%   { top: 0px;}
  25%  { top: 150px;}
  50%  { top: 300px;}
  75%  { top: 150px;}
  100% { top: 0px;}
}
```

動畫效果格式有七種屬性，格式如下：

「animation-name」是設定要播放動畫的名稱，預設值是「none」；

「animation-duration」是指動畫單次播放的時間，單位為秒(s)，預設為 0秒；

「animation-delay」是指動畫開始轉換前等待的時間，單位為秒(s)，預設為 0 秒；

「animation-iteration-count」是指動畫播放的次數，可以設定大於 0 的數字，預設為 1，但若設定

「infinite」則表示無限重複播放；

「animation-play-state」是設定動畫播放的狀態，「runing」是播放中，「paused」是暫停；

「animation-timing-function」是指動畫轉換速度，預設為「ease」，即開始與結束皆平滑的轉換；

轉換方法	說明
ease	預設值，平滑的開始，平滑的結束
linear	開始到結束保持平均的速度
ease-in	開始到結束由慢到快的轉換
ease-out	開始到結束由快到慢的轉換
ease-in-out	開始到結束由慢到快再到慢的轉換

「animation-direction」是指動畫播放的方向；

播放方向	說明
normal	預設值，從頭開始播放到最後
reverse	從最後反向播放到開始
alternate	從頭開始播放到最後，再由最後反向播放到開始
alternate-reverse	從最後反向播放到開始，再由開始播放到最後

animation-name：動畫名稱

animation-duration：動畫時間

animation-timing-function：動畫轉換方法

animation-delay：動畫延遲時間

animation-iteration-count：動畫重複次數

animation-direction：動畫播放方向

animation-play-state：動畫播放狀態

範例：

Ex:

```
<style>
@keyframes myanimation{
  from { background-color: yellow;}
  to { background-color: red;}
}

.box {
  width: 150px;
  height: 150px;
  background-color: lightgreen;
  border-radius: 50%;
  border: 1px dotted black;
  text-align: center;
  line-height: 150px;
  position: absolute;
  animation-name: myanimation;
  animation-duration: 3s;
  animation-timing-function: ease;
  animation-delay: 1s;
}
</style>

<div class="box">Light</div>
```

範例：

Ex:

```
@keyframes myanimation{
  from { left: 10px; }
  to { left: 300px;}
}

.box {
  width: 100px;
  height: 100px;
  background-color: lightblue;
  border: 1px dotted black;
  text-align: center;
  line-height: 100px;
  position: absolute;
  animation-name: myanimation;
  animation-duration: 1s;
  animation-timing-function: ease-in-out;
  animation-direction: alternate;
  animation-iteration-count: infinite;
}

.box:hover{
  animation-play-state: paused;
}

</style>

<div class="box">BOX</div>
```








七、整合設定動畫效果：animation

可以使用 animation 屬性來簡化語法，格式如下：

animation： 動畫名稱 動畫時間 動畫轉換方式 動畫延遲時間 動畫重複次數 動畫播放方向 播放狀況

範例：

Ex:

```
<style>
@keyframes myanimation{
  0%   {background:  red; left:10px; top:10px;}
  25%  {background:  yellow; left:310px; top:10px;}
  50%  {background:  blue; left:310px; top:210px;}
  75%  {background:  green; left:10px; top:210px;}
  100% {background:  red; left:10px; top:10px;}
}
.box {
  width: 100px;
  height: 100px;
  background-color:  lightblue;
  border: 1px dotted  black;
  text-align: center;
  line-height: 100px;
  position: absolute;
  animation: myanimation 4s ease-in-out alternate infinite;
}
</style>

<div class="box">BOX</div>
```