

學習要點:

1. 動畫簡介
2. 屬性詳解
3. 簡寫和版本

本章主要探討 HTML5 中 CSS3 的動畫效果，可以通過類似 Flash 那樣的關鍵幀模式控制運行。

一、動畫簡介

CSS3 提供了類似 Flash 關鍵幀控制的動畫效果，通過 `animation` 屬性實現。那麼之前的 `transition` 屬性只能通過指定屬性的初始狀態和結束狀態來實現動畫效果，有一定的局限性。

`animation` 實現動畫效果主要由兩個部分組成:

1. 通過類似 Flash 動畫中的關鍵幀聲明一個動畫:
2. 在 `animation` 屬性中調用關鍵幀聲明的動畫。

CSS3 提供的 `animation` 是一個復合屬性，它包含了很多子屬性。如下表所示:

屬性	說明
<code>animation-name</code>	用來指定一個關鍵幀動畫的名稱，這個動畫名必須對應一個 <code>@keyframes</code> 規則。CSS 加載時會應用 <code>animation-name</code> 指定的動畫，從而執行動畫。
<code>animation-duration</code>	用來設置動畫播放所需的時間
<code>animation-timing-function</code>	用來設置動畫的播放方式
<code>animation-delay</code>	用來指定動畫的延遲時間
<code>animation-iteration-count</code>	用來指定動畫播放的循環次數
<code>animation-direction</code>	用來指定動畫的播放方向
<code>animation-play-state</code>	用來指定動畫的播放狀態
<code>animation-fill-mode</code>	用來設置動畫的時間外屬性
<code>animation</code>	以上的簡寫形式

除了這 9 個屬性之外，動畫效果還有一個重要的屬性，就是關鍵幀屬性: `@keyframes`。它的作用是聲明一個動畫，然後在 `animation` 調用關鍵幀聲明的動畫。

//基本格式，“name”可自定義

```
@keyframes name {  
    /* ... */  
}
```

```
}
```

二、屬性詳解

在講解動畫屬性之前，先創建一個基本的樣式。

//一個 div 元素

```
<div>我是 HTML5</div>
```

//設置 CSS

```
div {  
    width: 200px;  
    height: 200px;  
    background-color: white;  
    border: 1px solid green;  
}
```

1. @keyframes

//創建動畫的第一步，先聲明一個動畫關鍵幀。

```
@keyframes myani {  
    0% {  
        background-color: white;  
        margin-left: 0px;  
    }  
    50% {  
        background-color: black;  
        margin-left: 100px;  
    }  
    100% {  
        background-color: white;  
        margin-left: 0px;  
    }  
}
```

//或者重複的，可以並列寫在一起

```
@keyframes myani {  
    0%, 100% {  
        Background-color: white;  
        Margin-left: 0px;  
    }  
}
```

```

    50% {
        Background-color:black;
        Margin-left:0px;
    }
}

```

2. animation-name

//調用@keyframes 動畫

animation-name: myani;

屬性值	說明
none	默認值，沒有指定任何動畫
INDEX	是由@keyframes 指定創建的動畫名稱

3. animation-duration

//設置動畫播放的時間

animation-duration: 1s;

當然，以上通過關鍵偵的方式，這裡插入了三個關鍵偵。0%設置了白色和左偏移為 0，和初始狀態一致，表明從這個地方開始動畫。50%設置了黑色，左偏移 100px。而 100%則是最後一個設置，又回到了白色和左偏移為 0。整個動畫就一目了然了。

而對於關鍵偵的用法，大部分用百分比比較容易控制，當然，還有其他一些控制方法。

//從甚麼狀態過渡到甚麼狀態

```

@keyframes myani {
    from {
        background-color: white;
        margin-left:0px;
    }
    to {
        background-color:black;
        margin-left:100px;
    }
}

```

4. animation-timing-function

//設置緩動

animation-timing-function: ease-in;

屬性值	說明
ease	默認值，元素樣式從初始狀態過渡到終止狀態時速度由快到慢，逐漸變慢。等同於貝塞爾曲線(0.25,0.1,0.25,1.0)
linear	元素樣式從初始狀態過渡到終止狀態速度是恆速。等同於貝塞爾曲線(0.0,0.0,1.0,1.0)
ease-in	元素樣式從初始狀態過渡到終止狀態時，速度越來越快，呈一種加速狀態。等同於貝塞爾曲線(0.42,0,1.0,1.0)
ease-out	元素樣式從初始狀態過渡到終止狀態時，速度越來越慢，呈一種減速狀態。等同於貝塞爾曲線(0.42,0,0.58,1.0)
ease-in-out	元素樣式從初始狀態過渡到終止狀態時，先加速，再減速。等同於貝塞爾曲線(0.42,0,0.58,1.0)
cubic-bezier	自定義三次貝塞爾曲線

5. animation-delay

//設置延遲時間

animation-delay: 1s;

6. animation-iteration-count

//設置循環次數

animation-iteration-count: infinite;

屬性值	說明
次數	默認值為 1
infinite	表示無限次循環

7. animation-direction

//設置緩動方向交替

animation-direction: alternate;

屬性值	說明
normal	默認值，每次播放向前
alternate	一次向前，一次向後，一次向前，一次向後這樣交替

8. animation-play-state

//設置停止播放動畫

animation-play-state: paused;

9. animation-fill-mode

//設置結束後不再返回

animation-fill-mode: forwards;

屬性值	說明
none	默認值，表示按預期進行與結束
forwards	動畫結束後繼續應用最後關鍵幀位置，即不返回
backforwards	動畫結束後迅速應用起始關鍵幀位置，即返回
both	根據情況產生 forwards 或 backforwards 的效果

//both 需要結合，次數和播放方向

animation-iteration-count: 4;

animation-direction: alternate;

六、簡寫和版本

//簡寫形式完整版

animation: myani 1s ease 2 alternate 0s both;

為了兼容舊版本，需要加上相應的瀏覽器前綴，版本信息如下表:

	Opera	Firefox	Chrome	Safari	IE
支持需帶前綴	15 ~ 29	5 ~ 15	4 ~ 42	4 ~ 8	無
支持不帶前綴	無	16+	43+	無	10.0+

//兼容完整版，Opera 在這個屬性上加入 webkit，所以沒有-o-

-webkit-animation: myani 1s ease 2 alternate 0s both;

-moz-animation: myani 1s ease 2 alternate 0s both;

-ms-animation: myani 1s ease 2 alternate 0s both;

animation: myani 1s ease 2 alternate 0s both;

//@keyframes 也需要加上前綴

@-webkit-keyframes myani {...}

@-moz-keyframes myani {...}

@-o-keyframes myani {...}

@-ms-keyframes myani {...}

Keyframes myani {...}