HTML5程序设计基础

第十七章 CSS3动画

主要内容

01 过渡

02 动画

01 过渡

transition

- CSS3 transform 属性所实现的元素变形，呈现的仅仅是一个“结果”，而 transition 呈现的是一种过渡“过程”，即一种动画转换过程，如渐显、渐隐、动画快慢等。
- **transition** 通过将元素的某个属性从一个属性值在指定的时间内平滑过渡到另一个属性值来实现动画功能。

语法

值	描述
transition-property	规定设置过渡效果的 CSS 属性的名称。
transition-duration	规定完成过渡效果需要的时间。默认是 0。
transition-timing-function	规定过渡效果的时间曲线。默认是 "ease"。
transition-delay	定义过渡效果何时开始。默认是 0。

定义过渡属性

transition-property 指定参与过渡的属性，语法格式如下：

```
transition-property：all | none | <property>[ ,<property> ]*
```

- all：默认值，表示所有可以进行过渡的 CSS 属性；
- none：表示不指定过渡的 CSS 属性；
- <property>：表示指定要进行过渡的 CSS 属性。可以同时指定多个属性值，以逗号“,”进行分隔。

定义过渡属性

```
div{
        background-color: #ffff00;
 }

div:hover{
        background-color: #00ffff;
        transition-property: background-color;
 }
```

demo17-1.html

定义过渡时间

transition-duration 属性指定过渡持续的时间，即设置从旧属性换到新属性花费的时间（以秒或毫秒计）。语法格式如下：

```
transition-duration：<time>[ ,<time> ]*
```

- <time>默认值为0，适用于所有元素以及 :before 和 :after伪元素，如果存在多个属性值，以逗号“,”进行分隔。

定义过渡延迟时间

transition-delay 属性规定在过渡效果开始之前需要等待的时间，以秒或毫秒计。语法格式如下：

```
transition-delay: time
```

- 默认值为0，适用于所有元素以及:before 和 :after伪元素。
- 设置时间可以为正整数、负整数和0。非零时必须设置单位为秒s或者毫秒ms。为负数时过渡的动作会从该时间点开始显示，之前的动作被截断；为正数的时候，过渡的动作会延迟触发。

定义过渡效果

transition-timing-function 属性规定过渡效果的速度曲线。
语法格式如下：

```
transition-timing-function：linear | ease | ease-in | ease-out |
ease-in-out | cubic-bezier(x1,y1,x2,y2)；
```

transition-timing-function过渡效果的速度曲线

值	效果	描述
linear	线性效果	规定以相同速度开始至结束的过渡效果（等于 cubic-bezier(0,0,1,1)）
ease	缓解效果	慢速开始，然后变快，最后慢速结束的过渡效果（cubic-bezier(0.25,0.1,0.25,1)）
ease-in	渐显效果	以慢速开始的过渡效果（等于 cubic-bezier(0.42,0,1,1)）
ease-out	渐隐效果	以慢速结束的过渡效果（等于 cubic-bezier(0,0,0.58,1)）
ease-in-out	渐显渐隐	以慢速开始和结束的过渡效果（等于 cubic-bezier(0.42,0,0.58,1)）
cubic-bezier(n,n,n,n)	贝塞尔曲线效果	在 cubic-bezier 函数中定义自己的值。可能的值是 0 至 1 之间的数值

transition-timing-function过渡效果的速度曲线

通过带有五个不同过渡效果值的div 元素理解过渡效果速度曲线：

```
#div1 {transition-timing-function: linear;}
#div2 {transition-timing-function: ease;}
#div3 {transition-timing-function: ease-in;}
#div4 {transition-timing-function: ease-out;}
#div5 {transition-timing-function: ease-in-out;}
```

demo17-2.html

语法

合写方式：

transition: all .5s ease-in .1s,

分写方式：

transition-property: all;

transition-duration: .5s;

transition-timing-function: ease-in;

transition-delay: .1s;

transition

```
div{
        background-color: #ffff00;
        transition: background-color 1s linear;
}
div:hover{
         background-color: #00ffff;
}
```

示例文字

示例文字

示例文字

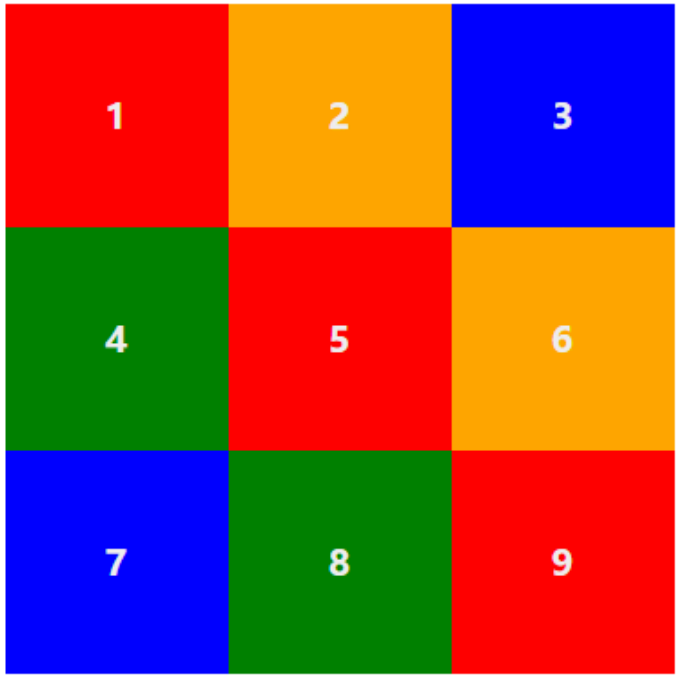
使用transition功能同时过渡多个属性值

如需向多个样式添加过渡效果，则添加多个属性，由逗号隔开：

```
div{
        background-color: #ffff00;
        color: #000000;
        width: 300px;
        transition:  background-color 1s linear, color 1s linear, width 1s linear;
}
div:hover{
        background-color: #003366;
        color: #ffffff;
        width: 400px;
}
```

demo17-3.html

练习



demo17-4.html

demo17-5.html

02 动画

动画

动态切换

动画

- 通过 CSS3 创建动画，可以在许多网页中取代动画图片、Flash 动画以及 JavaScript。动画效果使用 animation 属性来实现。
- animations 功能与 transitions 功能的相似点与不同点？
  - 相似点：都是通过改变元素的属性值来实现动画效果。
  - 不同点：transitions 过渡只是通过指定属性的开始值与结束值，然后以在这两个属性之间进行平滑过渡的方式来实现简单动画效果。animations 动画则是通过定义多个关键帧以及定义每个关键帧中元素的属性值来实现更为复杂的动画效果。

CSS3动画

- 在 CSS3 中，动画需要 2 步：

1. 制定动画关键帧

```
@keyframes 动画名{
    0%{
        background-color: red;
    }
    70%{
        background-color: yellow;
    }
    100%{
        background-color: red;
    }
}
```

2. 调用动画

animation-name ：调用动画名

animation-duration ：动画持续时间

……

@keyframes 规则

@keyframes 规则用于创建动画。在其中规定某项 CSS 样式，就能创建由当前样式逐渐改为新样式的动画效果。

在 @keyframes 中创建动画时，需要把它捆绑到某个选择器。

通过规定至少以下两项 CSS3 动画属性，即可将动画绑定到选择器：

规定动画的名称、动画的时长

```
@keyframes  关键帧集合名 { 创建关键帧的代码 }
```

创建关键帧集合

1.开始帧

2.背景色为深蓝色的关键帧

在整个动画过程中40%处有一帧为背景色是深蓝色的关键帧。

3.背景色为黄色的关键帧

在整个动画过程中70%处有一帧为背景色是黄色的关键帧。

4.结束帧

整个动画中最后一帧，在结束帧之后，元素的属性不再发生变化。

```
@keyframes mycolor{
        0%{
          background-color: red;
        }
        40%{
          background-color: darkblue;
        }
        70%{
          background-color: yellow;
        }
        100%{
          background-color: red;
        }
}
```

创建关键帧集合

创建好关键帧集合之后，在元素的样式中使用该关键帧的集合。

在animation-name属性中指定关键帧集合的名称。

```
div:hover{
        animation-name: mycolor;
        animation-duration: 5s;
        animation-timing-function: linear;
}
```

demo17-6.html

创建关键帧集合

```
position: relative;
top: 0;
left: 0;
```

```
div:hover {
    animation: mymove 2s 0.5s;
}
```

```
@keyframes mymove {
    0% {
        left: 0;
        top: 0;
    }
    50% {
        left: 300px;
        top: 0;
    }
    100% {
        left: 300px;
        top: 300px;
    }
}
```

demo17-7.html

animation

animation: name duration timing-function delay iteration-count direction;

值	描述
animation-name	规定需要绑定到选择器的 keyframe 名称。
animation-duration	规定完成动画所花费的时间，以秒或毫秒计。
animation-timing-function	规定动画的速度曲线。
animation-delay	规定在动画开始之前的延迟。
animation-iteration-count	规定动画应该播放的次数。
animation-direction	规定是否应该轮流反向播放动画。

animation-iteration-count

- animation-iteration-count 属性定义动画的播放次数。

animation-iteration-count : n | infinite;

值	描述
n	定义动画播放次数的数值，默认为1
infinite	规定动画应该无限次播放。

demo17-8.html

实现多个属性值同时改变的动画

在各关键帧中同时指定多个属性值可以实现多个属性值同时改变的动画。

demo17-9.html

练习

demo17-10.html

THANKYOU