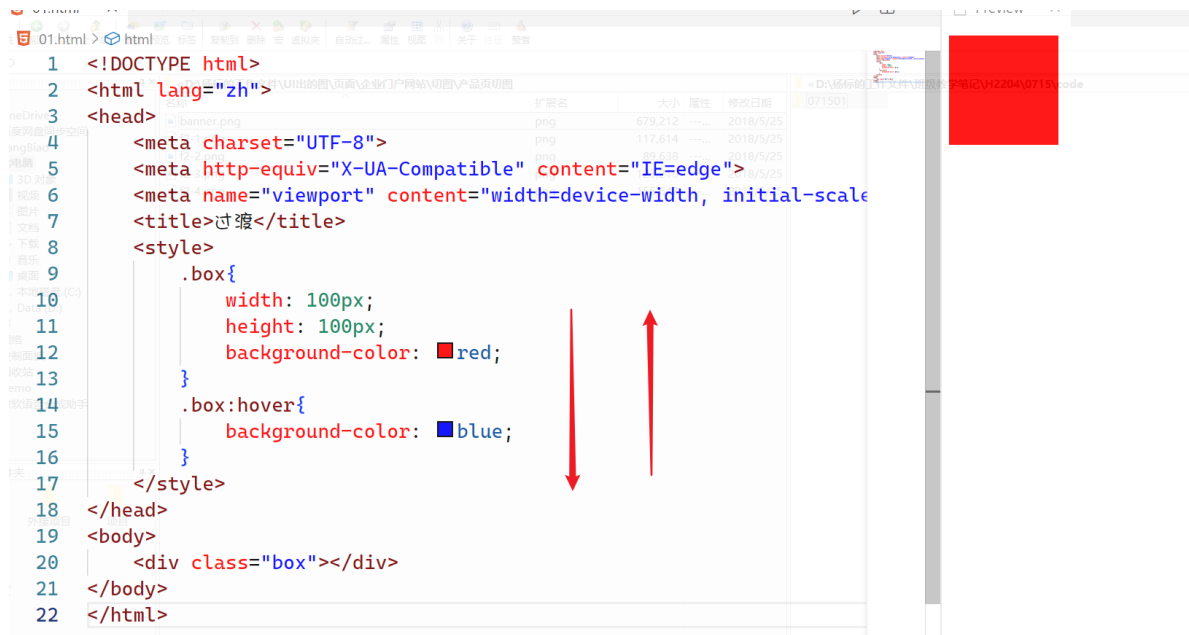
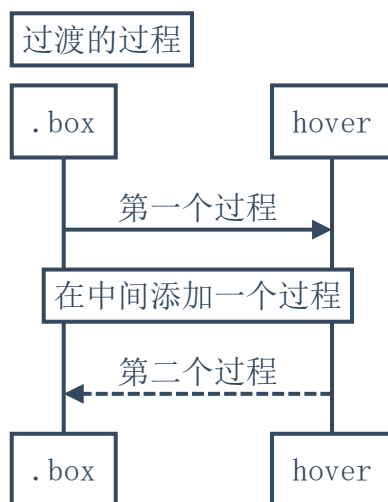


CSS3过渡

CSS过渡指的是元素从一个状态到另一个状态的转变过程，过渡使用的是CSS当中的 `transition` 这个属性，这个属性本身很简单，但是它需结合起来使用



在上面的代码里面，我们可以看到，当鼠标放在元素上面以后，元素会由红色变成蓝色，当鼠标松开以后，元素会由蓝色转变成红色，这个转变是一瞬间就完成了，那么，我们如果在这个转里面添加一个过程，这过程就叫过渡



过渡的属性

1. `transition-property` 要执行过渡的属性
2. `transition-duration` 要执行过渡的时间，它可以以 `s` 秒为单位，也可以以 `ms` 毫秒为单位
3. `transition-timing-function` 过渡的效果，也叫过渡的时间函数【默认值是 `ease`】
 - `linear` 匀速
 - `ease-in` 先慢后快，`ease` 单词是轻松的意思，`in` 是开始的意思

- `ease-out` 先快后慢, `out` 代表结束的意思
- `ease-in-out` 前后都很慢, 中间很快
- 我们还可以在浏览器里面手动的编辑我们需要的效果, 如 `cubic-bezier(0.74, -1.06, 0.35, 2.34)`

4. `transition-delay` 过渡等待【默认值0s】

上面的四个上属性是可以合起来一起写的

```
1 transition-property: width;
2 transition-duration: 1s;
3 transition-timing-function: linear;
4 transition-delay: 3s;
```

下面的4个属性值可以简写成下面的方式

```
1 transition: width 1s linear 3s;
```

`transition` 后面的四个属性值是可以更改位置的, 如 `transition: 1s 3s linear width ;` ;
但是一定要注意, 在属性值里面有2个时间, 第一个时间代表过渡持续的时间, 第二个时间代表等待时间

多个属性的过渡

在上面学习属性的时候, 我们已可以设置一个属性的过渡, 如果有多个属性值发生变化以后也需要过渡, 那怎么办呢?

```
1 <style>
2   .box {
3       width: 100px;
4       height: 100px;
5       background-color: deeppink;
6       /*执行过渡*/
7       transition-property: width,height;
8       /*transition-duration: 2s,0.5s;*/
9       transition-duration: 2s;
10      transition-timing-function: linear,ease-in;
11      transition-delay: 4s,0s;
12  }
13
14  .box:hover {
15      width: 400px;
16      height: 200px;
17  }
18 </style>
19 <div class="box">
20     盒子
21 </div>
```

在上面的属性里面, 我们可以看到, 如果多个属性需要过渡, 我们也是可以的

同样多个属性的过渡也是可以简写成一个 `transition`

```
1 transition: width 2s linear 4s,height 0.5s;
```

全属性的过渡

当所有变化的属性都需要过渡的时候，我们可以使用 `all` 来代替

```
1 transition-property: all;    /*这里就使用了all来代替所有变化的属性*/
2 transition-duration: 4s;
3 transition-timing-function: linear;
4 transition-delay: 2s;
```

综合成一个属性

```
1 transition:all 4s linear 2s;
```

不对称的过渡效果

请看下面代码

```
1 <style>
2   .box{
3       width: 100px;
4       height: 100px;
5       background-color: deeppink;
6       transition-property: width;
7       transition-duration: 2s;
8       transition-delay: 0s;
9       transition-timing-function: linear;
10  }
11  .box:hover{
12      width: 300px;
13  }
14 </style>
15 <body>
16   <div class="box">
17   </div>
18 </body>
```

我们可以看到，当鼠标进入触发 `hover` 效果或鼠标离开失去 `hover` 效果，它的效果是一样的，但是我们现在希望改变一下

如：鼠标进去 `hover` 的时候过渡时长为 `2s`，鼠标离开失去 `hover` 效果的时候，过渡时长为 `5s` 怎么处理呢

```

<html>
<head>
  <meta charset="utf-8">
  <title>标准的过渡效果</title>
  <style>
    .box{
      width: 100px;
      height: 100px;
      background-color: deeppink;
      transition-property: width;
      transition-delay: 0s;
      transition-timing-function: linear;
      transition-duration: 5s; 失去hover的时候，它是5s
    }
    /* 进来的时候是2s，离开的时候是5s */
    .box:hover{
      width: 300px;
      transition-duration: 2s; 进入hover的时候它是2s
    }
  </style>
</head>
<body>
  <div class="box">
  </div>
</body>
</html>

```

像上面这种情况，就称之为不对称的过渡，它的进入效果和离开效果是不一样的

不能过渡的属性

在CSS3的过渡里面，并不是所有的属性都可以过过渡

1. float
2. display
3. visibility
4. overflow
5. position
6. z-index
7. font-family
8. text-align/text-align-last
9. cursor

同时 auto 的属性值也是不可以实现过渡效果的

兼容性处理

过渡是一个CSS3的属性，它在旧版本的浏览器里面是需要注意它兼容性

浏览器支持

表格中的数字表示支持该属性的第一个浏览器版本号。

紧跟在 -webkit-, -ms- 或 -moz- 前的数字为支持该前缀属性的第一个浏览器版本号。



属性					
transition	26.0 4.0 -webkit-	10.0	16.0 4.0 -moz-	6.1 3.1 -webkit-	12.1 10.5 -o-
transition-delay	26.0 4.0 -webkit-	10.0	16.0 4.0 -moz-	6.1 3.1 -webkit-	12.1 10.5 -o-
transition-duration	26.0 4.0 -webkit-	10.0	16.0 4.0 -moz-	6.1 3.1 -webkit-	12.1 10.5 -o-
transition-property	26.0 4.0 -webkit-	10.0	16.0 4.0 -moz-	6.1 3.1 -webkit-	12.1 10.5 -o-
transition-timing-function	26.0 4.0 -webkit-	10.0	16.0 4.0 -moz-	6.1 3.1 -webkit-	12.1 10.5 -o-

对于低版本的浏览器，我们需要添加特定的前缀

1. 以谷歌为核心的浏览器，我们要添加 `-webkit-`
2. 以IE为核心的浏览器则要添加 `-ms-`
3. 火狐浏览器则是使用 `-moz-`
4. 欧朋浏览器使用 `-o-`

案例

1. 根据效果图完成如下效果

变化前	变化后
	

```
1  <!DOCTYPE html>
2  <html>
3    <head>
4      <meta charset="utf-8">
5      <title>过渡的案例</title>
6      <style>
7        .switch-button {
8          --h: 40px;
9          width: 80px;
10         height: var(--h);
11         display: block;
12         border-radius: calc(var(--h) / 2);
13       }
14
15       .switch-button>input{
16         display: none;
17       }
18       .switch-inner-box {
19         width: 100%;
20         height: 100%;
21         background-color: #f5f5f5;
22         border-radius: inherit;
23         /* 过渡 */
24         transition: all 0.3s linear;
25       }
```

```

26         .switch-button>input:checked+.switch-inner-box {
27             background-color: #7dc9fd;
28         }
29         /* 小圆 */
30         .switch-button .circle {
31             width: var(--h);
32             height: var(--h);
33             background-color: #ffffff;
34             border-radius: 50%;
35             transition: inherit;
36             /* 把边框设置在小球上面 */
37             border: 5px solid #f5f5f5;
38             box-sizing: border-box;
39         }
40         .switch-button>input:checked+.switch-inner-box>.circle{
41             margin-left: 40px;
42             background-color: #bee3fd;
43             /* 在切换过来以后，把边框的颜色变掉 */
44             border: 5px solid #7ec9fd;
45         }
46     </style>
47 </head>
48 <body>
49     <!-- checkbox -->
50     <label class="switch-button">
51         <input type="checkbox">
52         <div class="switch-inner-box">
53             <div class="circle"></div>
54         </div>
55     </label>
56 </body>
57 </html>
58 <!--
59     总结：
60     1.checked的伪类实现状态的切换
61     2.使用了CSS的变量及CSS的计算
62     3.使用了inherit来主动继承外部的样式
63     4.使用了display来切换元素类型
64     5.使用了过渡，让效果变得更加的简单
65     -->

```

2. 要据效果图完成案例



```
1  <!DOCTYPE html>
2  <html lang="zh">
3    <head>
4      <meta charset="UTF-8">
5      <title>选项卡案例</title>
6      <style>
7        .clearfix::after {
8          content: "";
9          display: block;
10         clear: both;
11       }
12       .box{
13         width: 600px;
14         /* 溢出隐藏 */
15         overflow: hidden;
16       }
17       .label-box {
18       }
19       .label-box>label {
20         display: block;
21         width: 130px;
22         height: 50px;
23         border: 2px solid black;
24         text-align: center;
25         line-height: 50px;
26         float: left;
27         margin-right: 20px;
28       }
29       .tab-list{
30         height: 400px;
31         width: calc(600px * 3);
32         /* 过渡 */
33         transition: all 0.5s linear;
34       }
35       .tab-item{
36         width: 600px;
37         height: 100%;
38         float: left;
39       }
40       .tab-item:nth-child(1){
41         background-color: red;
42       }
43       .tab-item:nth-child(2){
44         background-color: blue;
45       }
46       .tab-item:nth-child(3){
47         background-color: lightblue;
48       }
49       #a1:checked~.tab-list{
50         margin-left: 0;
51       }
52       #a2:checked~.tab-list{
53         margin-left: -600px;
54       }
55       #a3:checked~.tab-list{
56         margin-left: -1200px;
57       }
```

```

58         .box>input{
59             display: none;
60         }
61     </style>
62 </head>
63 <body>
64     <div class="box">
65         <!-- 三个label标签 -->
66         <div class="label-box clearfix">
67             <label for="a1">选项一</label>
68             <label for="a2">选项二</label>
69             <label for="a3">选项三</label>
70         </div>
71         <input type="radio" name="aaa" id="a1">
72         <input type="radio" name="aaa" id="a2">
73         <input type="radio" name="aaa" id="a3">
74         <div class="tab-list clearfix">
75             <div class="tab-item">第一个选项卡</div>
76             <div class="tab-item">第二个选项卡</div>
77             <div class="tab-item">第三个选项卡</div>
78         </div>
79     </div>
80 </body>
81 </html>
82 <!--
83     总结:
84     1. 目前的状态切换可以有checked,target,hover,active,
85        这里我们使用是checked, 并且是单选
86     2. 合理的布局, 选择器只是能哥哥找弟弟 ,
87        所以input标签是要在切换元素的前面
88     3. 添加过渡效果
89     4. 要适合的使用新知识, 如calc
90 -->

```